STAT 498 B Marbles Simulation

Nan Tang 1662478

March 30, 2018

Q1

```
#Define a marble bag of 40 red and 40 clear.
#Simulate 1000 draws.
marbleBag <- rep(c("red", "clear"), 40)
draws <- 1000

#Sample size of 14 from a bag of 50% read and 50% clear marbles.
sampleSize <- 14
extremeEvents <- 0
for (i in 1 : draws) {
  marbleResult <- sample(marbleBag, sampleSize, replace = FALSE)
  if (all(marbleResult == "red") | all(marbleResult == "clear")) {
    extremeEvents <- extremeEvents + 1
  }
}
extremeEvents / draws
## Result is 0 ##
```

The result of this simulation conveys that the probability of having exactly half red and half clear marbles in 1000 draws of 14 marbles each is 0.

Q2

```
#Simulate 1000 draws with sample size of 28 from same marble bag.
sampleSize <- 28
extremeEvents <- 0
for (i in 1 : draws) {
  marbleResult <- sample(marbleBag, sampleSize, replace = FALSE)
  if (all(marbleResult == "red") | all(marbleResult == "clear")) {
    extremeEvents <- extremeEvents + 1
  }
}
```

```
extremeEvents / draws
## Result is 0 ##
```

This simulation demonstrates that the probability of having exactly half red half clear marbles in 1000 draws of 28 marbles each is also 0.

Q3

```
#Simulate 1000 draws with sample size of 4 from same marble bag, while
#success is defined to be exactly half red and half clear.
sampleSize <- 4
successEvents <- 0
for (i in 1 : draws) {
  marbleResult <- sample(marbleBag, sampleSize, replace = FALSE)
  if (sum(marbleResult == "red") == 2) {
    successEvents <- successEvents + 1
  }
}
successEvents / draws
## Result is 0.38 ##
```

This simulation show that within 1000 draws of 4 marbles each, the probability of getting half red and half clear is 0.38.

Q4

```
#This function calculates the proportion of schools which have over 3% of
#students scores top 1%.
distributionTestFun <- function(simulations, groupSize) {
  successProb = 0.01
  extremeProb = 0.03
  groupRandom <- rbinom(simulations, p = successProb, size = groupSize)
  groupProb <- groupRandom / groupSize
  extremeCount <- sum(groupProb > extremeProb)
  return(extremeCount / simulations)
}

#Calculate proportion of schools which have over 3% students scores top 1%
#within 1000 Group A schools.
group_a <- distributionTestFun(1000, 1000)
group_a
## Result is 0 ##

#Calculate proportion of schools which have over 3% students scores top 1%
```

```
#within 1000 Group B schools.
group_b <- distributionTestFun(1000, 100)
group_b
## Result is 0.016 ##
```

The result of the simulation demonstrates that in groups of different school size, the proportions of schools that have over 3% top students vary largely. In group A, which contains 1000 school of 1000 students each, 0% of these schools have over 3% top students. In group B, which is consisted of schools of size 100, shows 1.6% of these schools have over 3% top students.

Q5

```
#This function calculates the proportion of schools which have an average score below 15
distributionTestFun2 <- function(simulations, groupSize) {
  scoreMean <- 160
  stanError <- 40 / sqrt(groupSize)
  groupRandom <- rnorm(simulations, mean = scoreMean, sd = stanError)
  extremeCount <- sum(groupRandom < 155)
  return(extremeCount / simulations)
}

group_a_extreProb <- distributionTestFun2(1000, 1000)
group_a_extreProb
## Result is 0 ##

group_b_extreProb <- distributionTestFun2(1000, 100)
group_b_extreProb
## Result is 0.104 ##
```

Within these two groups of schools with normal distributed score, 0% schools of group A which is consisted of schools of 1000 students have an average score below 155, while 10.4% schools of group B which is consisted of schools of 100 students have an average score below 155.