Tram Le

Homework 3:

1. n = 1000, p = .1

a) Expected value: *E[X] = n* p = 1000 * .1 = 100*
b) Variance value: *Var[X] = n * p * (1 – p) = 1000 * .1 * (1 - .1) = 90*
Standard deviation: *Sd(X) = sqrt(Var(X) = sqrt(90) = 9.49*
c) Probability at most 100 patients: P(X <= 100)
pbinom(100, n, p) = pbinom(100, 1000, .1) = .53
d) Normal approximate:
P(X <= 100): *pnorm(100, mean, sd, lower.tail = TRUE) = 0.5* (Because Z value is less than or equal to 0 in Z table)

2. N = 12 + 8 = 20; n(size numbers of trials) = 5 ; x(number of success) = 5,

a) Probability success on single trial of red balls: p = 12/20 = 0.6.
P(X=5): *dbinom(x, n, p) = 0.07776*
b) Expected value: *E[X] = n*p = 5 * .6 = 3*
c) Standard deviation: *sd(X) = sqrt(np(1-p)) = 1.1*

3. mean = 100; sd = 15. Standard normal distribution: P(Z <= (x-mean)/sd) or P(Z >= (x-mean)/sd)

a) IQ < 120 : *pnorm(120, mean, sd, lower.tail = TRUE) = 0.908*
b) 110 < IQ < 130: *pnorm(130, mean, sd, lower.tail = TRUE) - pnorm(110, mean, sd, lower.tail = FALSE) = 0.724*
c) IQ > 140: *pnorm(140, mean = 100, sd = 15, lower.tail = FALSE) = 0.0038*

4. k = 5; p = .6

a) n = 100,000
Cal the sum of random geometric values: *sum(rgeom(5, .6))*

```
for(i in 1:n){
  if (sum(rgeom(5, .6)) < 8)
    countSum = countSum + 1
}
print(countSum)
>> 94423
```

b) Ratio of the number of sum < 8 / total number of trial n
ratio = countSum / n = 94423/ 100000 = .94423
c) Negative binomial distribution with r = 5; p = .6
P(X <= 7): *pnbinom(7 , r ,p) = .94269*
d) Negative binomial distribution is approximately equal to the sum of geometric distribution

5.

a) Population size 1000000
   Using rep() to generate a population of "Graduate" and "Not Graduate" of 32% and 68% of total population 1000000 than put in a vector.
   g = rep(c("Graduate", "Not Graduate"), times = c(320000, 680000))

b) n= 1000
   Using sample() to generate random sample of 1000 out of total population then count the
   sample(g, n)
c) Count sample success of "Graduate" then find the estimate for proportion of "Graduate"
   count = sum(s == "Graduate")
   p = count/n
d) 99% ⇔ 2.58; z value = .99506
   Apply formula:
   *ma = qnorm(.99506)*sqrt(p*(1-p)/n)*

   Cal lower and upper of confidence interval:
   low = p – ma
   high = p + ma
e) install.packages("epitools")
   library("epitools")
f) Cal 99% confidence interval using binom.approx()
   binom.approx(p, n, conf.level = 0.99)