190186546-MAS6002-A2

Student Registration Number: 190186546

2019/11/8

Contents

Part 1

The following function `getDefectiveProb` returns the proportion of defective leaflets in the random sample by setting p_1, p_2, N_1, N_2 and the number of samples. Specifically, since the paper printing conforms to the binomial distribution, the function simulates the case where the printing device 1 and 2 print out defective leaflets, that is, `N1` and `N2`. They are then integrated into `N` and randomly sampled therefrom to obtain the proportion of defective leaflets.

```
getDefectiveProb <- function(p_1,p_2,num.N1,num.N2,num.sample){
    N1 = rbinom(n = num.N1, size = 1, prob = p_1)
    N2 = rbinom(n = num.N2, size = 1, prob = p_2)
    N = c(N1,N2)
    prob = sum(sample(N,num.sample))/num.sample
    return (prob)
}
```

Part 2

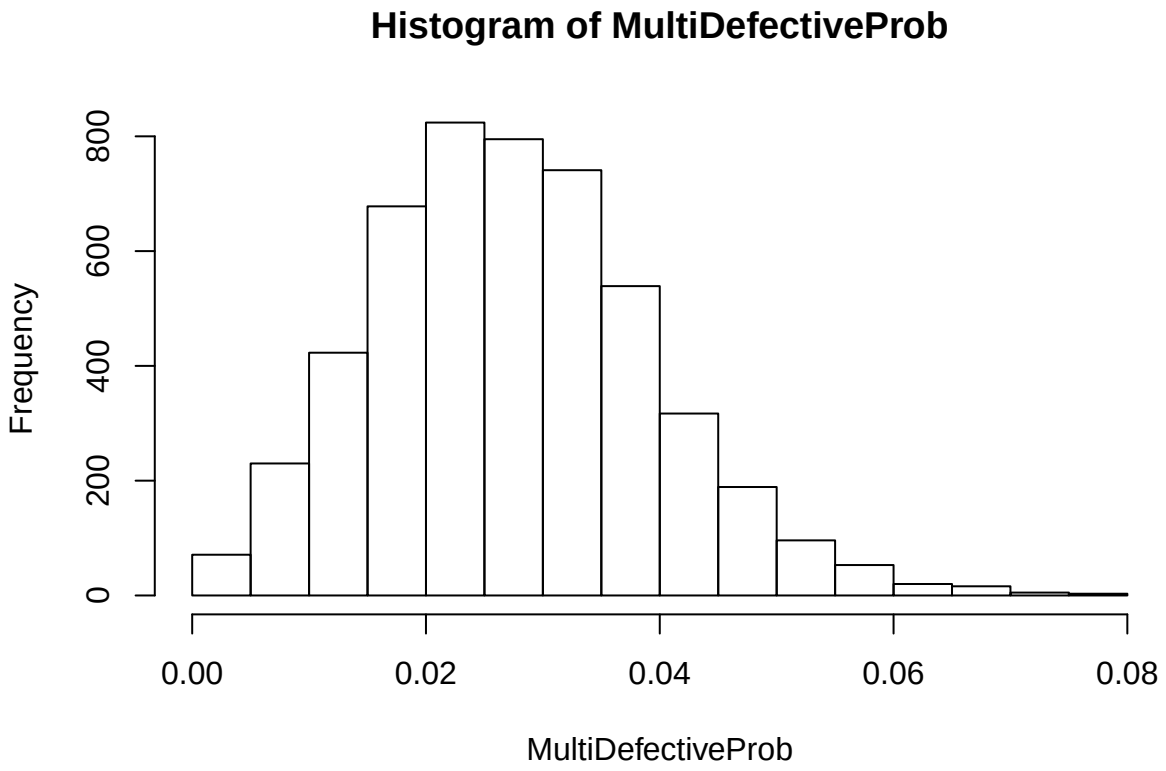
Set $p_1 = 0.05$, $p_2 = 0.02$, $N_1 = 10000$, $N_2 = 20000$ and $n_{\text{sample}} = 200$, and use the function in **Part 1** to calculate the proportion of defective leaflets. Since it is a sample, the entire calculation is iterated 5000 times to observe the probability distribution.

```
p_1 = 0.05
p_2 = 0.02
num.N1 = 10000
num.N2 = 20000
num.sample = 200
simulation = 5000
```

```
MultiDefectiveProb = replicate(simulation,
                               getDefectiveProb(p_1,p_2,num.N1,num.N2,num.sample))
```

After iterating 5000 times, draw the histogram of the proportion.

```
hist(MultiDefectiveProb)
```



Then calculate the mean of the probability distribution.

```
mean(MultiDefectiveProb)
```

```
## [1] 0.02993
```

Finally calculate its quantile.

```
quantile(MultiDefectiveProb,probs = c(0.025, 0.975))
```

```
##  2.5% 97.5%
## 0.010 0.055
```

Part 3

The `getSampleNum` function uses a binary search method (link: wikipedia) to continuously try different n_{sample} to calculate whether it satisfies $Pr(|p-\hat{p}|<\epsilon)>0.95$, where $1 \leqslant n_{\text{sample}} \leqslant N_1+N_2$. In particular,

1. Define a function `getlessThanEProb` to calculate the value of $Pr(|p-\hat{p}|<\epsilon)$ for given n_{sample}.
2. Build function `getSampleNum` using the binary search method to find the minimum sample value that meets $Pr(|p-\hat{p}|<\epsilon)>0.95$.

In addition, in order to speed up the operation, the parallel package `parallel` is used here. The new function `RepParallel` mimics the `replicate` function to achieve parallel processing which is included in a package published by Calhoun (2016). This function `RepParallel` only uses multi-core to speed up, independent of

the algorithm in the code. `RepParallel` can be replaced with R's built-in function `replicate`, but the speed may drop.

Parallel processing

```
library(parallel)
RepParallel <- function(n, expr, simplify = "array",...){
  mc <- getOption("mc.cores", detectCores())
  answer <-
    mclapply(integer(n), eval.parent(substitute(function(...) expr)),mc.cores = mc,...)
  if (!identical(simplify, FALSE) && length(answer))
    return(simplify2array(answer, higher = (simplify == "array")))
  else return(answer)
}
```

Calculate $Pr(|p - \hat{p}| < \epsilon)$

```
getlessThanEProb <- function(N1,N2,p,e,simulation,sample){
  prob = mean(replicate(simulation, abs(getDefectiveProb(p,p,N1,N2,sample)-p)<e))
  return (prob)
}
```

Find the minimum number of samples

```
getSampleNum <- function(p,e,N1,N2){
  top = N1+N2
  button = 1
  while(button<top){
    pin_sample = ceiling((top-button)/2+button)
    pin_Pr = mean(RepParallel(20,getlessThanEProb(N1,N2,p,e,30,pin_sample)))
    if (pin_Pr > 0.95){
      top = pin_sample - 1
    }
    else{
      button = pin_sample + 1
    }
  }
  return (top)
}
```

Part 4

Set $p_1 = p_2 = 0.1$, $N_1 = 10000$, $N_2 = 20000$ and $\epsilon = 0.05$, and use the function in **Part 3** to calculate the minimum number of samples which meet the condition. Due to the randomness of the sample, the entire calculation will iterate 100 times to observe the probability distribution of the minimum number of samples n_{sample} taken.

```
N1 = 10000
N2 = 20000
p = 0.1
e = 0.05
simulation = 100
```

```
getSampleNum(p,e,N1,N2)
```

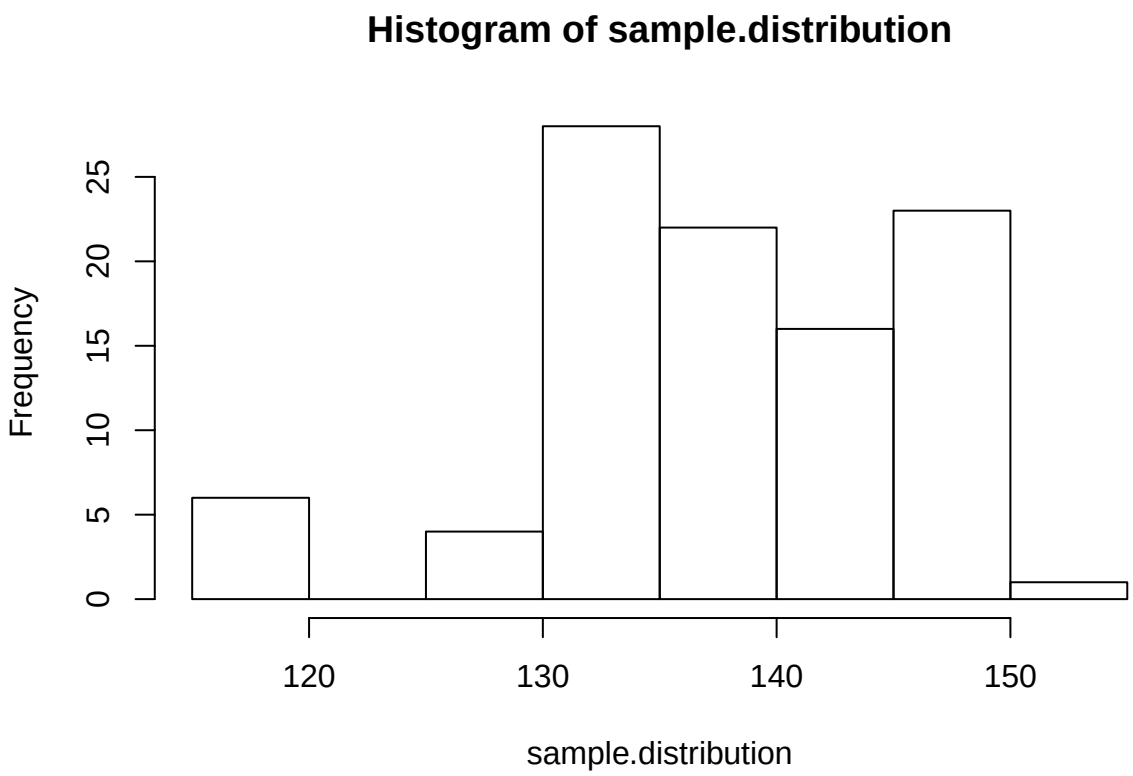
```
## [1] 134
```

Iterate the process to see its distribution.

```
sample.distribution = replicate(simulation, getSampleNum(p,e,N1,N2))
```

Histogram of the minimum number of samples satisfying the condition.

```
hist(sample.distribution)
```



Mean of the minimum number of samples that satisfy the condition

```
ceiling(mean(sample.distribution))
```

```
## [1] 138
```

Part 5

Question 1

Given condition: $1 \leqslant n \leqslant N_1 + N_2$, find p and ϵ that satisfy $Pr(|P - \hat{p}| < \epsilon) = 1$.

We know that

$$Pr(|P - \hat{p}| \geqslant \epsilon) + Pr(|P - \hat{p}| < \epsilon) = 1$$
$$Pr(|P - \hat{p}| \geqslant \epsilon) = 1 - Pr(|P - \hat{p}| < \epsilon)$$

Suppose Y is the population and y is the sample.

$$\bar{Y} = \frac{\sum_1^N y_i}{N} = \frac{A}{N} = P$$

Also, for the sample, we have,

$$\bar{y} = \frac{\sum_1^n y_i}{n} = \frac{a}{n} = P$$

And we get to know $E(\hat{p}) = P$, thus by using Chebyshev's Inequality $P(|X - E(x)| \geqslant b) \leqslant \frac{Var(X)}{b^2}$, we can deduce that

$$\begin{aligned} Pr(|P - \hat{p}| \geqslant \epsilon) &\leqslant \frac{Var(\hat{p})}{\epsilon^2} \\ 1 - Pr(|P - \hat{p}| < \epsilon) &\leqslant \frac{Var(\hat{p})}{\epsilon^2} \\ Pr(|P - \hat{p}| < \epsilon) &\geqslant 1 - \frac{Var(\hat{p})}{\epsilon^2} \end{aligned}$$

If we want the formula $Pr(|P - \hat{p}| < \epsilon) = 1$ to be true, there is a unique solution,

$$\begin{aligned} 1 &= 1 - \frac{Var(\hat{p})}{\epsilon^2} \\ Var(\hat{p}) &= 0 \end{aligned}$$

Because $\sum_1^N y_i^2 = A = NP$ and $\sum_1^n y_i^2 = a = nP$.

Hence,

$$\begin{aligned} S^2 &= \frac{1}{N-1}\sum_1^N (y_i - \bar{Y})^2 = \frac{1}{N-1}\sum_1^N y_i^2 - N\bar{Y}^2 \\ &= \frac{1}{N-1}(NP - NP^2) = \frac{N}{N-1}P(1-P) \end{aligned}$$

Similarly,

$$s^2 = \frac{1}{n-1}\sum_1^n (y_i - \bar{y})^2 = \frac{n}{n-1}\hat{p}(1-\hat{p})$$

Because the sample proportion $\hat{p}$ is an unbiased estimate. The sample variance can be calculated from the following (Cochran 1977)

$$\begin{aligned} Var(\hat{p}) &= E(\hat{p} - \mu)^2 = E(\hat{p} - P) \\ &= S^2 \frac{1}{n}\frac{N-n}{N} = \frac{N}{N-1}P(1-P)\frac{1}{n}\frac{N-n}{N} \\ &= \frac{1}{n}\frac{N-n}{N-1}P(1-P) \end{aligned}$$

Thus,

$$\begin{aligned} Var(\hat{p}) &= 0 \\ \frac{1}{n}\frac{N-n}{N-1}P(1-P) &= 0 \end{aligned}$$

And we can easily get $P = 0$ or $P = 1$.

Another case is a constant conditional probability, ie $|P - \hat{p}| < \epsilon$ alway true to achieve $Pr(|P - \hat{p}| < \epsilon) = 1$. Because $0 \leqslant |P - \hat{p}| \leqslant 1$, $0 \leqslant P \leqslant 1$ and $0 \leqslant \hat{p} \leqslant 1$, we can get

$$\epsilon > max\{max\{P - \hat{p}\}, max\{\hat{p} - P\}\} = max\{p, 1 - P\}$$

In general, $P = 0$, $P = 1$ or $\epsilon > max\{p, 1 - P\}$ can achieve $Pr(|P - \hat{p}| < \epsilon) = 1$.

Question 2

Given condition: $1 \leqslant n \leqslant N_1 + N_2$ and $\epsilon > 0$, find the range of n that satisfy $Pr(|P - \hat{p}| < \epsilon) = 0$.

According to the Chebyshev's inequality in the first question,

$$\begin{cases} Pr(|P - \hat{p}| < \epsilon) \geqslant 1 - \frac{Var(\hat{p})}{\epsilon^2} \\ Pr(|P - \hat{p}| < \epsilon) = 0 \end{cases}$$

It is easy to deduce that,

$$\begin{aligned} 0 &\geqslant 1 - \frac{Var(\hat{p})}{\epsilon^2} \\ Var(\hat{p}) &\geqslant \epsilon^2 \end{aligned}$$

As we get $Var(\hat{p})$ in the above question, Through the equation $Var(\hat{p}) = \frac{1}{n}\frac{N-n}{N-1}P(1-P)$, we can deduce

$$\begin{aligned} \frac{1}{n}\frac{N-n}{N-1}P(1-P) &\geqslant \epsilon^2 \\ (N-n)P(1-P) &\geqslant n(N-1)\epsilon^2 \\ NP(1-P) &\geqslant n((N-1)\epsilon^2 + P(1-P)) \\ 1 \leqslant n &\leqslant \frac{NP(1-P)}{(N-1)\epsilon^2 + P(1-P)} \end{aligned}$$

In general, n that satisfies the condition is in the following range,

$$1 \leqslant n \leqslant \frac{NP(1-P)}{(N-1)\epsilon^2 + P(1-P)}$$

Where $N = N_1 + N_2$.

References

Calhoun, Gray. 2016. "RepParallel: Parallel Version of 'Replicate'." https://rdrr.io/github/grayclhn/dbframe-R-library/man/RepParallel.html.

Cochran, William G. (William Gemmes). 1977. *Sampling Techniques*. 3rd ed. Wiley Series in Probability and Mathematical Statistics. New York ; London: Wiley.