

**POINT ESTIMATION AND CONFIDENCE INTERVAL
FOR POPULATION PROPORTION
UNDER TRIPLE SAMPLING SCHEME**

Jun Dan

Department of Mathematical Sciences
Indiana University South Bend
E-mail Address: *jundan@uemail.iu.edu*

Submitted to the faculty of
Indiana University South Bend
In partial fulfillment of requirements for the degree of

**MASTER OF SCIENCE
in
APPLIED MATHEMATICS & COMPUTER SCIENCE**

Advisor

Dr. Yi Cheng
Department of Mathematical Science

Committee

Dr. Dean Alvis
Dr. Liguu Yu

Accepted by the Graduate Faculty, Indiana University South Bend, in partial fulfillment of the requirements for the degree of Master of Science.

Master's Thesis Committee:

Yi Cheng, Ph.D.

Dean Alvis, Ph.D.

Liguo Yu, Ph.D.

Dana Vrajitoru, Ph.D.
Graduate Program Director
Applied Mathematics and Computer Science

Copyright©2011
Jun Dan
All Rights Reserved

Dedication

*This thesis is dedicated to my husband Chunlei Li,
my mom Shuru Ren, and my grandma Qi Sun.*

Abstract

Population proportion is the percentage of the population that has a particular characteristic. The estimation for population proportion has broad applications in academic and industry fields such as insurance, banking, medical studies, bio-complexity and so on. Regarding an estimation procedure for population proportion, sampling scheme plays an important role. It directly decides the sampling space and the distribution of the sample statistics in interest, consequently, affects the results of estimations, from which people perceive and explore the characteristics of the population.

Classical statistics in this area mostly focuses on a single random sample. Double sampling scheme has been increasingly gaining attention in the last three decades. Compared with a single sampling, a double sampling scheme can save resources by culling a population early in the sampling process while keeping the error rates under the nominal level in a hypothesis testing framework. The estimation process follows testing procedure as data cumulates. One possible further extension is to work with a triple sampling designed data.

This thesis will focus on both point and confidence interval estimations for population proportions under a triple sampling scheme when the population following a binomial distribution. On the basis of introducing and reviewing methodologies for both single and double sampling schemes, this paper will explore how a triple sampling machinery works for the estimating process under a binomial distribution. The later part of this paper renders algorithms and simulation results for both double and triple sampling estimations, to evaluate the performance of this newly developed methodology.

Acknowledgments

In completing this graduate thesis, I am grateful for the supports and encouragements that many people have provided for me.

First of all, I appreciate Dr. Yi Cheng, for her great amount of time and energy putting in helping make this thesis better. I appreciate for her great teaching too, which built for me an insight and necessary backgrounds for a long path.

I thank the rest of my thesis committee members: Dr. Dean Alvis and Dr. Ligu Yu, their feedbacks helped to make this thesis better in many ways.

I also thank Dr. Zhong Guan and Dr. Morteza Shafii-Mousavi for many of the academic backgrounds they helped me to build.

All the professors mentioned above with whom I have taken two or more classes. Their meticulous teaching, patient tutoring and valuable advising provided for me a view, not only of my study at Indiana University South Bend, but also a preview of future. I dedicate my sincere appreciation and best regards to them.

As well, I thank Dr. Dana Vrajitoru, the current graduate program director, and Dr. Yu Song, the previous program director and chair of the Department of Mathematical Sciences, for their helps in my moving forward to further studies.

At last, my deep gratitude goes to my beloved family members: my husband Chunlei Li, my mom Shuru Ren and my grandma Qi Sun. The tireless supports they provided for me throughout my life are always remembered in my heart. I love them!

TABLE OF CONTENTS

1. Introduction	1
2. Literature Review	3
2.1 Single Sampling Scheme	3
2.1.1 Point Estimator for p in Single Sampling Scheme	4
2.1.2 Two Types of Confidence Intervals for p in Single Sampling Scheme	4
2.2 Group Sequential Sampling Scheme	5
2.3 Double Sampling Scheme	6
2.3.1 Point Estimator in Double Sampling Scheme	7
2.3.2 Confidence Interval for p in Double Sampling Scheme	9
3. Thesis studies: Estimation and Possible Advantages Under Triple Sampling Scheme	10
3.1 Descriptions of Triple Sampling Scheme	11
3.2 Analysis for Possible Advantages of Triple Sampling Scheme	12
4. My Research Objectives	13
4.1 Point Estimator for p Under Triple Sampling Scheme	14
4.2 Confidence Interval for p in Triple Sampling Scheme	14
5. Simulations and Numerical Studies	17
5.1 Sample Size Determination	17
5.2 Population Proportion	21

5.3 Design Parameters	22
5.4 Programming	27
6. Comparisons & Analysis of Simulation Results	27
6.1 Results Analysis for $p=p_0$	28
6.1.1 Results Analysis for Single Sampling Scheme when $p=p_0$	28
6.1.2 Results Analysis for Double Sampling Scheme when $p=p_0$	29
6.1.3 Results Analysis for Triple Sampling Scheme when $p=p_0$	31
6.2 Results Analysis for $p\neq p_0$	33
6.2.1 Results Analysis for Single Sampling Scheme when $p\neq p_0$	34
6.2.2 Results Analysis for Double Sampling Scheme when $p\neq p_0$	35
6.2.3 Results Analysis for Triple Sampling Scheme when $p\neq p_0$	36
6.3 Discussions	38
7. Conclusions	39
8. References	41
9. Appendixes	
Appendix I R Program Codes for Single Sampling Scheme	43
Appendix II R Program Codes for Computations of Table 3 & Table 4	45
Appendix III R Program Codes for Double Sampling Scheme	47
Appendix IV R Program Codes for Triple Sampling Scheme	52

1. Introduction

The generic problem of estimating a population proportion occurs often in business and industry. Data are often collected for the initial purpose of testing a hypothesis about the population proportion. Compared with a single sampling scheme, a double sampling plan can save resources by culling a population early in the sampling process, but how is a triple sampling scheme? Does a triple sampling scheme significantly improve the efficiency and accuracy of the estimation over a double sampling scheme? Or it does not worth the extra labor and time payment in case of a close yield. A question of how to create a confidence interval for a population proportion arises when we use a triple sampling plan. This thesis is motivated by these questions.

There was a classic statistical literature in multistage sampling study. O'Brien and Fleming (1979) and Pocock (1977) had proposed to implement multiple stages testing procedure for clinic trials, as early as nineteen seventies. They showed in their papers that it was remarkably better to divide patient entry into a few equal-sized groups and then carried out the test procedures based on accumulated data within each group, namely interval analysis, until sufficient statistical evidence was achieved. All these experiments were designed for two treatments, normal response and known variance with possible extension to binary data.

In specifying multiple stage ideas into the statistical inference of a binomial distributed population, a double sampling is the most popular inheritance from multiple stage sampling scheme for its relative simplicity. A good review of double

sampling scheme was given by Hewett and Spurrier (1983), which focused on determining optimal double sampling plans based on a specific hypothesis testing. More, Yeh and Van (1997) applied double sampling plan to the normally distributed population. Another application of double sampling scheme using for exponentially distributed population was introduced by Bulgren and Hewett (1973).

It is not a surprise that a double sampling is an efficient approach for a binomial distributed population. Cohen and Sackrowitz published their results of using Bayesian double sampling estimates for binomial distributed population, in 1984. An interesting paper was recently published by Ryan (2009) that addressed how to approximate confidence intervals for p in a double sampling scheme, where p is the notation for the population proportion of success. The fundamental difference between these two papers was that in the first paper the sample size of second stage depends on the result of the first stage experiment; while the two sample sizes were prefixed in the latter publication. Since I follow the second publication in my thesis closer than what I do with the first one, the detailed Bayes procedure of double sampling scheme is not discussed in this work.

The sections of this thesis are organized in a following sequence: Section 1 is an introduction and a brief review of sampling scheme literatures; Section 2 is a detailed review of Ryan(2009), the literature I followed as a basis of this thesis; Section 3 through Section 6 are the methodology study, numerical study, comparisons and discussions of simulation results respectively, where a full picture of the point estimation and confidence interval of the population proportion of a binomial distributed population under triple sampling scheme is rendered.

2. Literature Review

Sampling is a part of statistical practice that concerns with the selection of a representative or random subset of individual observations within a population of individuals intended to yield some knowledge about the population in interest, especially for the purposes of making predictions based on statistical inference. Sampling is an important aspect of data collection and there are various ways to obtain a random sample.

A sampling plan for the population proportion of a specific characteristic, denoted as p , from a binomial distributed population, is a sampling scheme and a set of rules for making decisions. The decision, based on counting the number of successes, named S_n , in a sample with size n , can be to accept the null hypothesis $H_0: p=p_0$, reject the null or accept the alternative hypothesis $H_a: p \neq p_0$ equivalently, or to take another sample.

2.1 Single Sampling Scheme

For a single sampling plan, one sample of observations is selected at random, from the population which follows a binomial distribution with success proportion p . The conclusion about the value of p , equaling to p_0 or not, is determined from a statistical procedure based on a specific testing statistic that is a function of the point estimator \hat{p} . Single sampling plan may be the most common and easiest plans to use. However, it is not the most efficient one in terms of the average number of sample sizes needed to satisfy the regulations.

2.1.1 Point Estimator for p in Single Sampling Scheme

To estimate the probability of success p , the simplest way is using a single sampling. In other words, a total of n independent and identical Bernoulli experiments are performed before any data analysis and the number of success S_n are observed in order to derive an estimation of p , specifically,

$$\hat{p} = \frac{S_n}{n}. \quad (1)$$

The probability mass function (PMF) of S_n is

$$P(S_n) = C_n^{S_n} p^{S_n} (1 - p)^{n - S_n}, \quad (2)$$

Where S_n takes values in $0, 1, \dots, n$. The above formula can be easily derived and it can be found in any standard statistics textbook.

2.1.2 Two Types of Confidence Intervals for p in Single Sampling Scheme

Although the point estimator of p , the population proportion, is straight forward for a simple random sampling, the interval estimation of the probability of success can be quite complicated depending on value of p and different sample size n if we wish to have guaranteed confidence level.

Henceforth, several notations would be introduced. Let $\hat{p} = S_n/n$ be the point estimator of p and κ be the $1 - \alpha/2$ quantile of a $N(0, 1)$ distribution.

For Wald interval of p (quoted by Ryan, 2009) is

$$\hat{p} \pm \kappa \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}. \quad (3)$$

Wald interval has been generally recommended given that $np(1-p)$ is greater than 5 or 10. However, after serious scrutiny, Brown, Cai and DasGupta (2001) found that the a standard Wald interval may perform poorly and can be undesirable in many scenarios, even if n is large.

To address such a basic yet important problem in the statistical practice, several alternative intervals have been introduced and studied. Among them, an effective method is Wilson interval approach, which is among the major topics of this thesis.

To illustrate Wilson interval approach, more quantities are introduced and denoted as follows:

$$\widetilde{S}_n = S_n + \kappa^2/2, \quad \widetilde{n} = n + \kappa^2 \quad \text{and} \quad \widetilde{p} = \widetilde{S}_n/\widetilde{n}.$$

The Wilson interval of p (quoted by Brown, Cai and DasGupta, 2001) is

$$\widetilde{p} \pm \frac{\kappa n^{\frac{1}{2}}}{\widetilde{n}} \sqrt{\hat{p}(1-\hat{p}) + \frac{\kappa^2}{4n}}. \quad (4)$$

2.2 Group Sequential Sampling Scheme

Except for the simplicity of a single sampling, there are many hindrances to apply the single sampling scheme in a real practice. The foremost problem is nased on sample size. According to the large number theorem, more experiments are

preferred in order to achieve asymptotic property. However, in reality of binomial experiments, even if substantial amount of trials have been practiced, the outcomes could be hard to interpret or even misleading, especially in clinical trials. On the other hand, unnecessary large sample may result in waste of resources.

In order to deal with such an issue, group sequential method has been developed in the area of clinical trials. Armitage published his pioneer work in 1975. The idea of the group sequential method is to sample data sequentially and assess the accumulated data repeatedly as interim analyses. More specifically, given two treatments under comparison, the assessment of accumulating data after every stage of patient entry would indicate whether or not the statistical evidence for a treatment difference is sufficient to stop the trial and conclude efficiency, or evidence is sufficient to claim futility.

The group sequential method has an advantage over fully sequential method. The fully sequential procedure is time consuming due to the continuous analysis of observations after every single entry. It may not even be feasible. An adaptive way to reduce the amount of work is to perform the test at longer equally-spaced intervals, termed group sequential sampling scheme. This modification was introduced by Pocock in 1977.

Furthermore, Demets and Ware (1982) presented their studies and conclusions of asymmetric group sequential boundaries for testing procedures in clinical trials.

2.3 Double Sampling Scheme

A special case of group sequential method is double sampling scheme. A

triple sampling scheme is a further extension of a double sampling scheme. In a double sampling scheme, the whole experiment would be divided into two stages. In the first stage, n_1 subjects are observed. Depending on the results of the first stage, $n_2 - n_1$ more subjects might be observed. A criteria to determine whether the second stage is necessary is to compare the number of success (S) in the first stage with two integer numbers, denoted by a and b ($0 \leq a < b \leq n_1$). If $a \leq S \leq b$, the second stage examination need to be carried out. Otherwise, the experiment terminates at the end of the first stage.

It can be derived that the PMF of S_n under a double sampling plan is,

$$P(n, s) = C(n, s)p^s(1 - p)^{n-s}, \quad (5)$$

where $C(n, s)$ is the path count from $(0,0)$ to (n, s) with formula

$$C(n, s) = \begin{cases} C_{n_1}^s, & \text{if experiment terminates at stage 1;} \\ \sum_{i=a}^b C_{n_1}^i C_{n_2-n_1}^{s-i}, & \text{if experiment terminates at stage 2,} \end{cases} \quad (6)$$

in which,

$$n = \begin{cases} n_1, & \text{if experiment terminates at stage 1;} \\ n_2, & \text{if experiment terminates at stage 2.} \end{cases}$$

2.3.1 Point Estimator in Double Sampling Scheme

A natural way to estimate p is to calculate the ratio of the number of success S and the total times of experiments n as

$$\hat{p} = \frac{S}{n}. \quad (7)$$

Ryan (2009) presented a more sophisticated point estimator of p by applying uniform minimum variance unbiased estimator (UMVUE) to double sampling, given by

$$\tilde{p} = C'(n, s)/C(n, s), \quad (8)$$

where $C'(n, s)$ is the path count from $(1,1)$ to (n, s) with formula

$$C'(n, s) = \begin{cases} C_{n_1-1}^{s-1}, & \text{if experiment terminates at stage 1 and } S_1 \geq 1; \\ \sum_{i=a}^b C_{n_1-1}^{i-1} C_{n_2-n_1}^{s-i}, & \text{if experiment terminates at stage 2 and } S_1 \geq 1, \end{cases}$$

which implies $n - 1$ observations with $S - 1$ successes.

Based on the work of Jung and Kim (2004), Equation (8) can be further reduced as

$$\tilde{p} = \begin{cases} \hat{p} = \frac{S}{n_1}, & \text{if experiment terminates at stage 1;} \\ \frac{\sum_{j=a}^b \frac{j}{n_1} C_{n_1}^j C_{n_2-n_1}^{s-j}}{\sum_{j=a}^b C_{n_1}^j C_{n_2-n_1}^{s-j}}, & \text{if experiment terminates at stage 2.} \end{cases}$$

(9)

In terms of variance, \hat{p} and \tilde{p} are comparable. Because of the simplicity of \hat{p} , I will use \hat{p} as the point estimator of p .

2.3.2 Confidence Interval for p in Double Sampling Scheme

Assume that $\hat{p} = x$ is observed. Corresponding to the point estimator \hat{p} , the endpoints of the confidence interval are the solutions, p_l and p_u , to the equations

$$\frac{\alpha}{2} = \sum_{\{(n,s) \in \nabla; \frac{S}{N} \geq x\}} C(n,s) p_l^s (1 - p_l)^{n-s}, \quad (10)$$

$$\frac{\alpha}{2} = \sum_{\{(n,s) \in \nabla; \frac{S}{N} \leq x\}} C(n,s) p_u^s (1 - p_u)^{n-s}, \quad (11)$$

where ∇ is the sample space, the set of all possible outcomes of the experiment.

The right hand side of Equation (10) is an increasing function of p_l for p_l in $(0, \hat{p})$, which can be verified by the first order derivative test. Similar argument can be used to prove that the right-hand side of Equation (11) is a decreasing function of p_u for p_u in $(\hat{p}, 1)$. Thus, these conditions ensure a unique solution of the confidence interval.

A more effective algorithm for double sampling scheme, Wilson confidence interval, is developed by Ryan (2009). There are two possible scenarios as follows:

- (i) Wilson interval with $n = n_1$, if experiment terminates at stage 1;
- (ii) Solve equations (10) and (11), if experiment terminates at stage 2, which leads to the following two equations:

$$\frac{\alpha}{2} = P_{p_l}(X_1 > b) + P_{p_l}(\{a \leq X_1 \leq b\} \cap \{X_1 + X_2 \geq s\})$$

(12)

and

$$\frac{\alpha}{2} = P_{p_u}(X_1 < a) + P_{p_u}(\{a \leq X_1 \leq b\} \cap \{X_1 + X_2 \leq s\}),$$

(13)

where $X_1 \sim N(n_1 p, n_1 p(1-p))$ and $X_2 \sim N((n_2 - n_1)p, (n_2 - n_1)p(1-p))$ approximately. Moreover, X_1 and X_2 are independent random variables given n_1 and n_2 . The Wilson interval has been given in Equation (4). The Equations (12) and (13) can be solved numerically and the similar arguments as the last paragraph guarantee a unique interval solution.

In my numerical study, the Wilson interval, as a feasible approximation, is applied for the calculation of confidence interval instead of solving Equation (12) and (13). The simulation results in section 6 validate that such a substitution does not bring unexpected violations, say of probability coverage ratio, length of confidence intervals and so on.

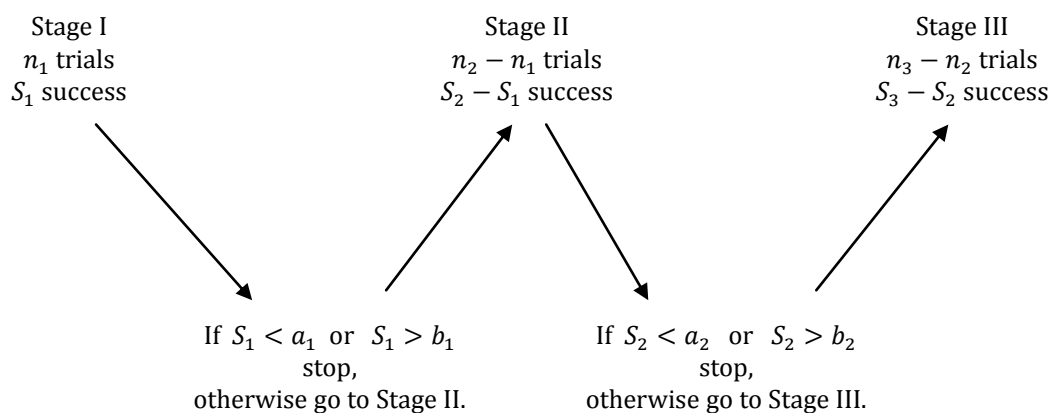
3. Thesis studies: Estimation and Possible Advantages Under Triple Sampling Scheme

In this section, I will discuss about the main focus of the thesis, the point estimation and confidence interval, under a special type of group sequential sampling plan– triple sampling scheme.

3.1 Descriptions of Triple Sampling Scheme

Under a triple sampling scheme, the whole experiment would be divided into three stages. It is an analogue to a group sequential sampling method with fixed maximum number of observations that have been divided into three blocks.

The data are collected in a following way. For the first stage, n_1 subjects would be observed. Depending on the results of the first stage, $n_2 - n_1$ more subjects might be obtained. The criteria to determine whether the second stage is necessary is to compare the number of successes (S_1) in the first stage with two integer numbers, denoted by a_1 and b_1 ($0 \leq a_1 < b_1 \leq n_1$). If $a_1 \leq S_1 \leq b_1$, the second stage sampling needs to be carried out. Otherwise, the experiment terminates after the first stage. The requirement to proceed to the third stage is similar to the rules above with another set of a_2 and b_2 . If S_2 is not between a_2 and b_2 , the experiment stops. Otherwise, the experiment stops after completion of $n_3 - n_2$ more observations. The following diagram is a demonstration of the procedure of triple sampling.



3.2 Analysis for Possible Advantages of Triple Sampling Scheme

First of all, the evaluation of a triple sampling plan is to assess the performance characteristics in interest. Confidence interval is one of the important aspects when different sampling schemes are considered. There are several criteria for comparing confidence intervals. Two primary determinants are the actual coverage probability and the expected length of confidence interval. Based on existing literatures, Wilson confidence interval under a double sampling plan was shown to be conservative, which means that the actual coverage probability is never less than, and often higher than, $1 - \alpha$ for any p in $[0,1]$. Also, for fixed α and n , the confidence interval under a double sampling scheme maintains expected length comparing to that from a single sampling design. Since Wilson confidence interval approach under a triple sampling plan is similar to that under a double sampling plan in a analytical derivation of the formula but with possible one more stage, it is reasonable to anticipate that the outcomes of a triple sampling plan possess the same properties such as the conservativeness of confidence intervals.

Secondly, a triple sampling design includes two sets of designing parameters, (a_1, b_1) and (a_2, b_2) , which renders more flexibility than a double sampling. Supposing we have adequate computational resource, all points in the parameter space would be evaluated and the best set of parameters, which maintains the smallest mean squared error of p to \hat{p} and the shortest length of confidence interval, would be obtained. In turn, a triple sampling design is expected to generate higher attained coverage ratio and comparable length of confidence interval with such set of parameters.

Moreover, the intervals are expected to be invariant to the choice of success

and failure in a symmetric triple sampling plan (*i. e.* $a_1 = n_1 - b_1, a_2 = n_2 - b_2$).

An intuitive explanation is that by exchanging the values of a_i 's and b_i 's, the success probability p exchanges to $1 - p$. Thus, the interval for p remains the same.

The second screen in a triple sampling design would focus on the study of the filtration of points which are rejected by the stage thresholds.

4. My Research Objectives

Giving the description of a triple sampling scheme, my objective here is to find a point estimator for p and then extend the results to the confidence interval formula for p under this scheme, where population follows a binomial distribution.

The PMF of S under a triple sampling plan is given by

$$P(n, s) = C(n, s)p^s(1 - p)^{n-s}, \quad (14)$$

where $C(n, s)$ is the path count from $(0,0)$ to (n, s) with the formula given below:

$$C(n, s) = \begin{cases} C_{n_1}^s, & \text{if experiment terminates at stage 1;} \\ \sum_{i=a_1}^{b_1} C_{n_1}^i C_{n_2-n_1}^{s-i}, & \text{if experiment terminates at stage 2;} \\ \sum_{i=a_1}^{b_1} C_{n_1}^i \sum_{j=a_2}^{b_2} C_{n_2-n_1}^{j-i} C_{n_3-n_2}^{s-j}, & \text{if experiment terminates at stage 3.} \end{cases} \quad (15)$$

4.1 Point Estimator for p Under Triple Sampling Scheme

One natural way to estimate p is to calculate the ratio of the number of success S and the total number of experiments n , that is,

$$\hat{p} = \frac{S}{n}. \quad (16)$$

Extend the uniform minimum variance unbiased estimator (UMVUE) given by Ryan (2009) to a triple sampling scheme, we have

$$\tilde{p} = C'(n, s)/C(n, s), \quad (17)$$

where $C'(n, s)$ is the path count from $(1,1)$ to (n, s) that can be computed by the formula below:

$$C'(n, s) = \begin{cases} C_{n_1-1}^{s-1}, & \text{if experiment terminates at stage 1;} \\ \sum_{i=a_1}^{b_1} C_{n_1-1}^{i-1} C_{n_2-n_1}^{s-i}, & \text{if experiment terminates at stage 2;} \\ \sum_{i=a_1}^{b_1} C_{n_1-1}^{i-1} \sum_{j=a_2}^{b_2} C_{n_2-n_1}^{j-i} C_{n_3-n_2}^{s-j}, & \text{if experiment terminates at stage 3.} \end{cases}$$

4.2 Confidence Interval for p in Triple Sampling Scheme

Assume that $\hat{p} = x$ is observed. Given a point estimator \hat{p} , the endpoints of the confidence interval, if the original idea of Ryan(2009) is extended, with

confidence level $1 - \alpha$, are the solutions p_l and p_u of such equations:

$$\frac{\alpha}{2} = \sum_{\{(n,s) \in \mathcal{V} : \frac{S}{N} \geq x\}} C(n,s) p_l^s (1 - p_l)^{n-s}, \quad (18)$$

$$\frac{\alpha}{2} = \sum_{\{(n,s) \in \mathcal{V} : \frac{S}{N} \leq x\}} C(n,s) p_u^s (1 - p_u)^{n-s}, \quad (19)$$

where \mathcal{V} is again the sampling space.

The right-hand side of Equation (18) is an increasing function of p_l for p_l in $(0, \hat{p})$, which can be verified by taking the first order derivative. Similar argument can be used to prove that the right-hand side of Equation (19) is a decreasing function of p_u for p_u in $(\hat{p}, 1)$. Thus, these properties ensure a unique solution of the confidence interval.

In the following, I extend the result of Wilson confidence intervals further to the triple sampling scheme. In this case, there are three possible situations would be involved:

- (i) Wilson interval with $n = n_1$, if experiment terminates at stage 1;
- (ii) the interval (p_l, p_u) , where p_l and p_u satisfy the Equations (20) and (21), if experiment terminates at stage 2

$$\frac{\alpha}{2} = P_{p_l}(X_1 > b_1) + P_{p_l}(\{a_1 \leq X_1 \leq b_1\} \cap \{X_1 + X_2 \geq s\}), \quad (20)$$

$$\frac{\alpha}{2} = P_{p_u}(X_1 < a_1) + P_{p_u}(\{a_1 \leq X_1 \leq b_1\} \cap \{X_1 + X_2 \leq s\});$$

(21)

(iii) the interval (p_l, p_u) , where p_l and p_u satisfy the Equations (22) and (23), if experiment terminates at stage 3.

$$\begin{aligned} \frac{\alpha}{2} = & P_{p_l}(X_1 > b_1) \\ & + P_{p_l}(\{a_1 \leq X_1 \leq b_1\} \cap \{X_1 + X_2 \geq b_2\}) + P_{p_l}(\{a_1 \leq X_1 \leq b_1\} \\ & \cap \{a_2 \leq X_1 + X_2 \leq b_2\} \cap \{X_1 + X_2 + X_3 \geq s\}) \end{aligned} \quad (22)$$

and

$$\begin{aligned} \frac{\alpha}{2} = & P_{p_u}(X_1 < a_1) \\ & + P_{p_u}(\{a_1 \leq X_1 \leq b_1\} \cap \{X_1 + X_2 \\ & \leq a_2\}) + P_{p_u}(\{a_1 \leq X_1 \leq b_1\} \cap \{a_2 \leq X_1 + X_2 \leq b_2\} \cap \{X_1 + X_2 + X_3 \\ & \leq s\}), \end{aligned} \quad (23)$$

where $X_1 \sim N(n_1 p, n_1 p(1-p))$, $X_2 \sim N((n_2 - n_1)p, (n_2 - n_1)p(1-p))$ and $X_3 \sim N((n_3 - n_2)p, (n_3 - n_2)p(1-p))$ approximately, according to the Central Limit Theorem. Furthermore, X_1 , X_2 and X_3 are independent. The algebra expression of Wilson interval has been given by Equation (4). The second set of equations, Equations (20) through (23), can be solved numerically and a similar argument to the last paragraph on Page 14 guarantees a unique solution for the confidence interval.

Given the complexity of the formulas of these equations, Equations (20) through (23), I will compute Wilson confidence interval in my numerical study for its computational feasibility. As it will be seen in section 6, such a simplification does not compromise the accuracy of coverage ratio and the length of confidence interval.

5. Simulations and Numerical Studies

I use R to implement triple sampling plan, where R is a programming language and software environment for statistical computing and graphing, for the major simulation procedure. The simulation results in this thesis are based on 10000 repetitions.

5.1 Sample size determination

The first step of the simulation is to determine the sample size. In this case, the Wald statistics is utilized to calculate the sample size. Equation (3) provides the confidence interval when point estimator \hat{p} is given. Thus, sample size n can be expressed as

$$n = \frac{\hat{p}(1 - \hat{p})\kappa^2}{\varepsilon^2}, \quad (24)$$

where ε is the desired half length of the confidence interval. If \hat{p} is replaced by p and ε is set to a specified value, the sample size can be approximated from Equation (24) (the maximum of $\hat{p}(1 - \hat{p})$ is 1/4). Table 1 provides the approximation of

sample size under various confidence levels and various confidence interval lengths.

Table 1 Approximation of Sample Size

$1 - \alpha$	ϵ	n^*
0.9	0.05	271
	0.04	423
	0.03	752
	0.02	1691
	0.01	6764
0.95	0.05	385
	0.04	601
	0.03	1068
	0.02	2401
	0.01	9604
0.99	0.05	664
	0.04	1037
	0.03	1844
	0.02	4147
	0.01	16588

* $\hat{p} \times (1 - \hat{p})$ was substituted by $\frac{1}{4}$ in the above calculations, which makes n conservative.

According to the results in Table 1, as the length of confidence interval decreases, the sample size increases sharply. For example, given $\alpha = 0.01$,

$\varepsilon = 0.04$, $n = 1037$, contrast to $\varepsilon = 0.02$, $n = 4147$, the sample size increases by more than three times as the length cuts by one half. As confidence level decreases, the sample size increases, which can be seen for given $\varepsilon = 0.02$, $\alpha = 0.05$, $n = 2401$, and contrast to $\alpha = 0.01$, $n = 4147$. Codes for a single sampling scheme by using R program are given in Appendix I .

It is worth noting that moderate sample sizes, for those significantly less than 17000, should be taken into account. Large sample sizes, for those larger than or equal to 17000, might be costly and often infeasible. Hence, sample sizes of preliminary simulations in my numerical studies are chosen at 17000, 10000, 5000, 2000, 1000, 500, 250 and 100, respectively. Table 2 presents the results of preliminary simulations, where p_0 is the value we test whether p equals to in the null hypothesis.

Table 2 Results of Preliminary Simulations

$1 - \alpha$	p_0	Sample Size	Point Estimate	Coverage Ratio	Length of CI
0.95	0.99	17000	0.9901	0.953	0.003
		10000	0.9902	0.949	0.004
		5000	0.9900	0.948	0.006
		2000	0.9901	0.944	0.009
		1000	0.9903	0.965	0.013
		500	0.9909	0.965	0.018
		250	0.9900	0.962	0.027
		100	0.9901	0.920	0.049

0.95	0.7	17000	0.7002	0.955	0.014
		10000	0.7006	0.948	0.018
		5000	0.7002	0.948	0.025
		2000	0.7000	0.955	0.040
		1000	0.7001	0.953	0.057
		500	0.7002	0.953	0.080
		250	0.7002	0.955	0.113
		100	0.7005	0.939	0.176
0.95	0.5	17000	0.5000	0.949	0.015
		10000	0.4990	0.946	0.020
		5000	0.5001	0.954	0.028
		2000	0.4996	0.950	0.044
		1000	0.5002	0.947	0.062
		500	0.5008	0.947	0.087
		250	0.5000	0.947	0.123
		100	0.5001	0.936	0.192
0.95	0.2	17000	0.2000	0.953	0.012
		10000	0.2000	0.949	0.016
		1000	0.2001	0.948	0.049
		500	0.2002	0.952	0.07
		250	0.2001	0.951	0.099
		100	0.2004	0.937	0.154
0.95	0.02	17000	0.0200	0.946	0.004
		10000	0.0200	0.949	0.005

5000	0.0201	0.948	0.008
2000	0.0200	0.953	0.012
1000	0.0200	0.947	0.018
500	0.0201	0.966	0.025
250	0.0201	0.961	0.036
100	0.0200	0.95	0.061

* The simulations with $\alpha = 0.01$ and 0.1 are also performed. The results preserve the same characteristic as $\alpha = 0.05$ as shown in Table 2.

Based on the results in Table 2, a few conclusions can be drawn: 1) for the sample sizes considered here, the point estimator of p is pretty close to the actual value of p , with absolute deviations smaller than .001. 2) The ratio that the actual p included in the computed confidence interval, labeled as *Coverage Ratio* in Table 2, is near the nominal confidence level $1 - \alpha = 0.95$ except scenario for $n=100$, which seems have $1-\alpha$ less than nominal confidence levels. In most cases, these ratios are larger than the nominated confidence level 0.95, which is consistent with the conservativeness of n and Wilson confidence intervals. 3) As the sample size decreases, the average length of confidence intervals gets larger. 4) There are no noticeable differences between the results at extreme proportions ($p = 0.99$ and 0.02) and the results at other moderate proportions ($p = 0.7, 0.5$ and 0.2).

According to the results of preliminary single stage simulations, the following sample sizes under 300 will be used: 120, 180, 240 and 300.

5.2 Population proportion

The next step in the simulation is to determine the values of population proportion to be considered in the study. Ryan (2009) showed that the actual coverage probability and the expected length are similar for a single sampling scheme and a double sampling scheme except for p in $(0.3, 0.7)$. Therefore, 0.2, 0.5 and 0.7 would be considered as representatives. On the other hand, 0.02 and 0.99 are also under my study because they are the extreme values, which might lead to different results in terms of expected length of confidence intervals and coverage ratios.

In the following numerical studies, the average of all point estimators over repetitions is the simulated expected value of point estimator. The percentage that the actual population proportion falls within the calculated confidence interval is the approximation of the true confidence level. Moreover, the average length of confidence interval is also simulated in the numerical study.

5.3 Design Parameters

Besides choosing sample size and proportion p in the numerical study, another issue is how to divide the whole sample process into sub-sample stages. In addition, a set of design parameters, a_1 , b_1 , a_2 and b_2 , has to be determined. These two issues have been discussed in many literatures, such as Colton and McPherson (1976), Words (1974) and Pocock (1977) among others.

Colton and McPherson's proposal (1976) is more relevant because the total sample size is fixed in this study. The objective in their paper is to restrain both Type I or α -error associated with a null hypothesis and Type II or β -error associated with an alternative hypothesis. Although hypothesis testing is not a focus of my

thesis, Colton and McPherson's idea is applicable in terms of the selection of design parameters a_1 , b_1 , a_2 and b_2 .

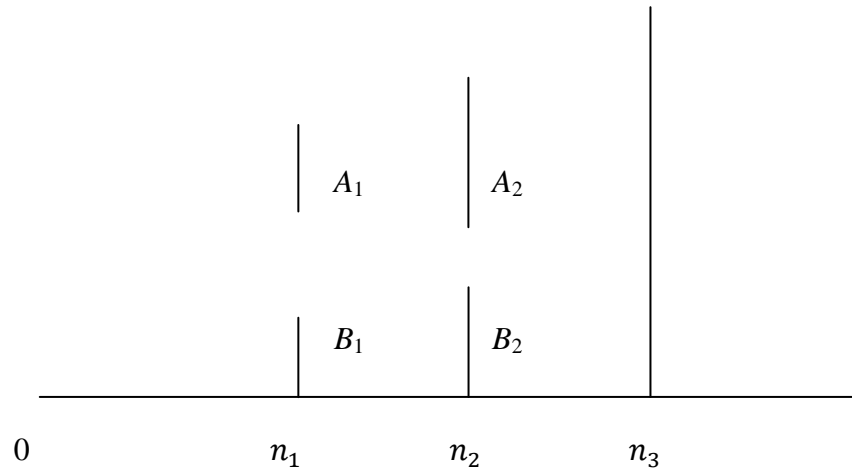


Figure 1 Illustration of triple sampling. The abscissa axis is the number of observations in triple sampling scheme. The ordinate axis represents the number of the success trials. A_1 , B_1 , A_2 , and B_2 define the stopping regions in the first stage and second stage respectively, where $A_1=[b_1, n_1]$, $B_1=[0, a_1]$, $A_2=[b_2, b_1 + n_2]$ and $B_2=[a_1, a_2]$.

In Figure 1, the experiments might stop, for instance, at region A_1 if the number of observed successes is greater than b_1 . Similarly, B_1 , A_2 and B_2 are stopping domains. Assume that the underlying null hypothesis is $H_0: p=p_0$. In the design of triple sampling scheme, if H_0 is true, the population proportion p is p_0 then it is more likely that the \hat{p} would fall in the intervals $[a_1/n_1, b_1/n_1]$ and $[a_2/n_2, b_2/n_2]$. In turn, the outcomes falling in region A_1 , A_2 , B_1 or B_2 are the events with small probability. Moreover, under H_0 the number of successes (if the experiment stops at the third stage) might deviate from p_0 with small probability. We would like to keep these facts in mind in the process of choosing design parameters. More specifically, the rule of determining the design parameters is to control the sum of the probabilities falling in the region of A_1 , A_2 , B_1 or B_2 at the nominal level. It is easy to

understand that, the minimization of the possibility of above events can be achieved by widening the gates of a_1 , b_1 , a_2 and b_2 . In turn, the advantage of the early termination of a triple sampling plan would be compromised. Thus, a balance between the reduction of Type I error rate and early termination should be achieved to generate a well-performing design parameter set.

The design parameters can be solved from following equation,

$$\begin{aligned}
\alpha = & P(\{\text{stop at } A_1\}) + P(\{\text{stop at } B_1\}) \\
& + P(\{\text{stop at } A_2\} \cap \{\text{do not stop at } A_1 \text{ and } B_1\}) \\
& + P(\{\text{stop at } B_2\} \cap \{\text{do not stop at } A_1 \text{ and } B_1\}) \\
& + P(\{\text{stop at the third stage with } S > (p * n_3 + 0.5)\} \\
& \cup \{\text{stop at the third stage with } S < (p * n_3 - 0.5)\}) \\
& \cap \{\text{do not stop at } A_1, B_1, A_2 \text{ and } B_2\}), \quad (25)
\end{aligned}$$

where the probabilities are computed at $p=p_0$, and a somewhat arbitrary interval $[p * n_3 - 0.5, p * n_3 + 0.5]$ is defined at the third stage. If the experiment terminates at the third stage, the total number of successes outside of above range is regarded as “incorrect” or deviating from p_0 . The numerical values of his interval can be varied according to the availability of computer resource and/or other statistical considerations.

It might be anticipated that the solutions to Equation (25) are not unique (Colton and McPherson 1976). The plan with smallest Average Sample Number (ASN) is defined as the best choice. Such a subject has become a separate research area. Because this topic is not in the scope of my thesis, the detailed discussion is

omitted here. In later parts, ASN is denoted as $E(\hat{n})$ in the tables of simulation analysis, through Tables 6 to 10.

Terms in Equation (25) can be expanded and expressed further as the following,

$$P(\{\text{stop at } A_1\}) = \sum_{b_1+1}^{n_1} C_{n_1}^i p^i (1-p)^{n_1-i},$$

$$P(\{\text{stop at } B_1\}) = \sum_0^{a_1-1} C_{n_1}^i p^i (1-p)^{n_1-i},$$

$$P(\{\text{stop at } A_2\} \cap \{\text{do not stop at } A_1 \& B_1\})$$

$$= \sum_{i=b_2+1}^{b_1+n_2} p^i (1-p)^{n_2-i} \sum_{j=a_1}^{b_1} C_{n_2}^{i-j} C_{n_1}^j,$$

$$P(\{\text{stop at } B_2\} \cap \{\text{do not stop at } A_1 \& B_1\})$$

$$= \sum_{i=a_1}^{a_2-1} p^i (1-p)^{n_2-i} \sum_{j=a_1}^{\min(b_1, a_2)} C_{n_2}^{i-j} C_{n_1}^j$$

and

$$P(\{\text{stop at the third stage, not in the desired range}\} \\ \cap \{\text{do not stop at } A_1, B_1, A_2 \& B_2\})$$

$$= \sum_{i=\text{floor}(p(n_3))+1}^{b_2+n_3} p^i (1-p)^{n_3} \sum_{j=a_2}^{b_2} C_{n_3}^{i-j} \sum_{k=a_1}^{b_1} C_{n_2}^{j-k} C_{n_1}^k \\ + \sum_{i=a_2}^{\text{floor}(p(n_3))-1} p^i (1-p)^{n_3} \sum_{j=a_2}^{b_2} C_{n_3}^{i-j} \sum_{k=a_1}^{b_1} C_{n_2}^{j-k} C_{n_1}^k.$$

The straightforward yet time consuming way to numerically solve above set of

equations is to iterate n_1 , n_2 and n_3 . For each stage, sets of possible combinations of a and b are plugged into the equations respectively to identify those sets that satisfy Equation (25). If Equation (25) holds for a set of designing parameters, that set of parameter can be saved and utilized in practice.

However, the method above is somewhat a computationally intensive. Thus, in practice of my numerical implementation I use following approach to make the computation attainable. In a double sampling design, say there are totally 500 possible pairs of values for (a, b) , the design parameters, can be obtained by the computing resource given sample size n_1 for the first stage. I take 10,000 iterations for each pair of (a, b) to get a set of estimator of proportion, coverage ratio and the length of confidence interval. Among those sets of results, I keep the first 100 pairs of design parameters which maintain coverage ratios higher than $1 - \alpha$. The reasons for this criteria is that, there might be either more or less than 100 pairs satisfying the criteria, but it is unnecessary to have that many pairs eventually. Thus, 100 is a number chosen to save computing resource. If the potential pairs are less than 100, we just use all available pairs. For an extreme situation that none is revealed to be potential, we can use the pair that maintains the highest coverage ratio among all 500 pairs. However, I didn't encounter such an extreme issue in my numerical study. The extreme scenario might exist though, but we infer the possibility of occurrence is few. Then, I pick one pair of the (a, b) which is the best point estimator, in the sense of having the smallest mean squared error (MSE) and the shortest length of confidence interval. When these two features are not focused on one pair of (a, b) , we choose the pair with the smallest MSE as a compromise, because MSE is, I believe, a more persuadable criteria in representative of goodness of an estimator in general senses. By far, this pair of (a, b) is fixed as the pair of

design parameters of the first stage of this experiment. Because each of the 100 pairs of (a, b) maintains a coverage ratio higher than $1 - \alpha$, MSE and the length of confidence interval come to my main concerns of picking right design parameters. The same approach is applied to a triple sampling design.

Although my method may not be an ideal one, it indeed provides sensible results that differ among single sampling, double sampling and triple sampling schemes. Moreover, given the computation resource and the work load (much greater than 500×1000), my method is a compromise between accuracy and computability.

5.4 Programming

Program codes for double & triple sampling scheme using R program are given in Appendix II and Appendix III, respectively.

6. Comparisons & Analysis of Simulation Results

Based on the preparations in Section 5, the main simulation is implemented. In this section, the results of single sampling, double sampling and triple sampling schemes are presented and compared. Various characteristics (point estimate, average length of confidence intervals and the probability of which the expected proportion is in the computed confidence interval) are compared in order to evaluate the efficiency of different sampling schemes.

The analysis of simulation results for all of single, double and triple

sampling schemes are processed under two scenarios: $p=p_0$ and $p\neq p_0$, in my numerical study.

6.1 Simulation Analysis when $p = p_0$

6.1.1 Simulation Analysis for Single Sampling Scheme when $p = p_0$

The overall performance of single sampling scheme is good. The average length of confidence interval at small sample size ($n = 120$) is larger than that at large sample size ($n = 300$), which can be derived from the formula of Wald's interval. The most noticeable phenomenon is that the probability of expected proportion within the confidence interval is lower than the $1 - \alpha$ for all p and for almost all sample sizes, especially when $p_0 = 0.02$ and 0.99 . These results are consistent with the results given by Ryan (2009) that the probability of expected proportion within the resulting confidence interval is lower than the $1 - \alpha$ when $p_0 \in [0.2, 0.5]$. In addition, two proportions at extreme values are included in my numerical study.

Table 5 Results of Single Sampling Scheme when $p = p_0$

n	$1 - \alpha$	p_0	Coverage Ratio	\hat{p}	Length of C.I.
120	0.95	0.02	0.9087	0.020	0.046
		0.2	0.9409	0.200	0.142
		0.5	0.9459	0.501	0.178
		0.7	0.9371	0.700	0.163
		0.99	0.6893	0.990	0.028

180	0.95	0.02	0.8674	0.020	0.039
		0.2	0.9465	0.200	0.116
		0.5	0.9576	0.501	0.146
		0.7	0.9500	0.701	0.133
		0.99	0.8310	0.990	0.026
240	0.95	0.02	0.8548	0.020	0.034
		0.2	0.9394	0.200	0.101
		0.5	0.9552	0.499	0.126
		0.7	0.9492	0.702	0.116
		0.99	0.9068	0.990	0.023
300	0.95	0.02	0.9336	0.020	0.031
		0.2	0.9436	0.200	0.090
		0.5	0.9455	0.500	0.113
		0.7	0.9434	0.700	0.104
		0.99	0.8019	0.990	0.021

6.1.2 Simulation Analysis for Double Sampling Scheme when $p = p_0$

Compared with a single sampling, a double sampling scheme indeed improves the probability of expected proportion within the resulting confidence interval at the points where a single sampling scheme produces insufficient coverage probability. The most striking result is the significant improvement of coverage ratio when $p = 0.02, 0.99$. Such improvement was gained by compromising the length of confidence interval, which is reasonable in the first glance because larger length of

confidence interval would include more points. However, high coverage ratio does not equivalent to cover $E(p)$ as pointed out by Ryan (2009). Therefore, the increasing of length of confidence interval is not due to higher coverage probability. A double sampling scheme does improve the accuracy compared with a single sampling.

Table 6 Results of Double Sampling Scheme when $p = p_0$

n_1	n_2	$1 - \alpha$	p_0	a	b	Coverage Ratio	\hat{p}	Length of C.I.	Early stops	$E(\hat{n})$
80	40	0.95	0.02	0	2	0.962	0.022	0.062	215	111.40
			0.20	2	22	0.955	0.200	0.142	33	118.68
			0.50	19	47	0.951	0.502	0.177	37	118.52
			0.70	44	61	0.953	0.701	0.164	90	116.40
			0.99	71	80	0.975	0.990	0.043	0	120.00
120	60	0.95	0.02	0	60	0.971	0.019	0.044	0	180.00
			0.20	11	29	0.954	0.199	0.118	87	174.78
			0.50	26	67	0.965	0.502	0.146	72	175.68
			0.70	39	89	0.952	0.700	0.135	109	173.46
			0.99	13	119	0.975	0.991	0.035	317	160.98
160	80	0.95	0.02	0	8	0.972	0.020	0.037	4	239.68
			0.20	5	43	0.962	0.200	0.101	13	238.96

			0.50	26	91	0.955	0.501	0.126	33	237.36
			0.70	31	118	0.953	0.700	0.117	108	231.36
			0.99	13	160	0.969	0.991	0.028	0	240.00
200	100	0.95	0.02	0	26	0.953	0.020	0.033	0	300.00
			0.2	36	41	0.958	0.200	0.101	577	242.30
			0.5	96	103	0.953	0.500	0.126	571	242.90
			0.7	56	141	0.959	0.701	0.111	362	263.80
			0.99	4	198	0.957	0.990	0.027	390	261.00

6.1.3 Simulation Analysis for Triple Sampling Scheme when $p = p_0$

The coverage ratio of confidence interval is much more improved under a triple sampling scheme compared with that under a single sampling scheme at the expense of widening the length of confidence interval. The coverage ratios of confidence interval under a double sampling plan and a triple sampling plan are comparable. So are the lengths of confidence intervals. The most important conclusion from these results is that a triple sampling plan tends to stop early more often than a double sampling plan, which means smaller sample size can be used to obtain satisfactory results with desired statistical properties. It has been noticed that $a_1 = 0$ for almost all the cases. This is because I have experimented all possible pairs of design parameters from $a_1 = 0$, and it turns out that the wide opening of the first gate does not compromise the accuracy of the final results. In other words, a

double sampling plan is a good choice for designing the experiments because the parameter space is smaller than that of a triple sampling plan.

In the following, the comparison among single sampling, double sampling and triple sampling schemes will be presented.

Table 7 Results of Triple Sampling Scheme when $p = p_0$

n_1	n_2	n_3	$1 - \alpha$	a_1	b_1	a_2	b_2	p_0	Coverage Ratio	\hat{p}	Length of C.I.	Early stop	$E(\hat{n})$
60	30	30	0.95	0	2	0	2	0.02	0.968	0.0205	0.061	248	109.47
				0	12	8	9	0.2	0.979	0.204	0.179	995	77.88
				0	30	2	33	0.5	0.953	0.504	0.219	990	77.73
				0	42	7	27	0.7	0.959	0.706	0.201	1000	76.50
				0	60	23	87	0.99	0.964	0.99	0.054	946	91.62
90	45	45	0.95	0	2	2	1	0.02	0.969	0.021	0.058	1000	122.90
				0	18	5	11	0.2	0.958	0.204	0.148	999	115.29
				0	45	2	48	0.5	0.968	0.507	0.182	1000	114.17
				0	63	2	62	0.7	0.954	0.703	0.165	1000	115.61
				0	90	1	50	0.99	0.969	0.99	0.041	1000	135.00
120	60	60	0.95	0	3	1	56	0.02	0.969	0.021	0.042	221	215.46

				0	24	2	13	0.2	0.965	0.205	0.129	1000	152.64
				0	60	4	64	0.5	0.96	0.503	0.157	1000	154.62
				0	84	2	68	0.7	0.959	0.702	0.144	1000	153.66
				0	119	0	164	0.99	0.974	0.991	0.035	1000	161.34
150	75	75	0.95	0	3	4	4	0.02	0.969	0.021	0.041	836	211.73
				0	30	3	15	0.2	0.959	0.203	0.114	1000	193.65
				0	75	2	88	0.5	0.959	0.502	0.142	1000	192.00
				0	105	2	107	0.7	0.966	0.703	0.130	1000	191.48
				0	149	2	216	0.99	0.962	0.99	0.031	1000	209.70

6.2 Simulation Analysis when $p \neq p_0$

The result of simulation study, for the case when p and p_0 are not equal, is even more persuasive than the case of equivalent value. Under this situation, the true proportion p is different from the null hypothesis proportion p_0 . However, the point estimate is very close to the true proportion p . The coverage ratio of confidence interval is near the nominal level as well. These are shown as the simulation results in Tables 8, 9 and 10. These two pieces of evidence indicate the efficiency of double and triple sampling schemes, that is, even if p and p_0 are not equal the estimation mechanism following of the multi-sampling hypothesis testing procedure could adjust the point estimate and confidence interval. Also, the results indicate the goodness of the approach in the selection of design parameters that we

described in the last section.

6.2.1 Simulation Analysis for Single Sampling Scheme under $p \neq p_0$

Table 8 Results of Single Sampling Scheme under $p \neq p_0$

n	$1 - \alpha$	p_0	p	Coverage Ratio	\hat{p}	Length of C.I.
120	0.95	0.12	0.02	0.9086	0.021	0.046
		0.3	0.2	0.9410	0.200	0.142
		0.6	0.5	0.9458	0.500	0.178
		0.8	0.7	0.9371	0.700	0.163
		0.89	0.99	0.6893	0.990	0.028
180	0.95	0.12	0.02	0.8674	0.021	0.039
		0.3	0.2	0.9465	0.203	0.116
		0.6	0.5	0.9576	0.499	0.146
		0.8	0.7	0.9501	0.699	0.134
		0.89	0.99	0.8311	0.990	0.025
240	0.95	0.12	0.02	0.8547	0.020	0.034
		0.3	0.2	0.9394	0.200	0.101
		0.6	0.5	0.9552	0.504	0.126
		0.8	0.7	0.9492	0.702	0.116
		0.89	0.99	0.9068	0.990	0.023
300	0.95	0.12	0.02	0.9335	0.020	0.031
		0.3	0.2	0.9434	0.200	0.090

0.6	0.5	0.9455	0.500	0.113
0.8	0.7	0.9434	0.700	0.104
0.89	0.99	0.8018	0.990	0.021

6.2.2 Simulation Analysis for Double Sampling Scheme when $p \neq p_0$

Table 9 Results of Double Sampling Scheme when $p \neq p_0$

n_1	n_2	$1 - \alpha$	p_0	p	a	b	Coverage Ratio	\hat{p}	Length of C.I.	Early stops	$E(\hat{n})$
80	40	0.95	0.12	0.02	0	2	0.9681	0.022	0.062	226	110.96
			0.3	0.2	2	22	0.9791	0.202	0.143	49	118.04
			0.6	0.5	19	47	0.9532	0.502	0.177	57	117.72
			0.8	0.7	44	61	0.9592	0.706	0.164	126	114.96
			0.89	0.99	71	80	0.9641	0.991	0.044	0	120.00
120	60	0.95	0.12	0.02	0	60	0.9698	0.020	0.044	0	180.00
			0.3	0.2	11	29	0.9580	0.203	0.120	115	173.10
			0.6	0.5	26	67	0.9682	0.503	0.147	80	175.20
			0.8	0.7	39	89	0.9543	0.702	0.136	131	172.14
			0.89	0.99	13	119	0.9694	0.991	0.036	281	163.14

160	80	0.95	0.12	0.02	0	8	0.9695	0.020	0.037	0	240.00
			0.3	0.2	5	43	0.9657	0.201	0.101	23	238.16
			0.6	0.5	26	91	0.9660	0.502	0.126	33	237.36
			0.8	0.7	31	118	0.9591	0.704	0.118	139	228.88
			0.89	0.99	13	160	0.9741	0.990	0.028	0	240.00
200	100	0.95	0.12	0.02	0	26	0.9694	0.020	0.033	0	300
			0.3	0.2	36	41	0.9594	0.202	0.102	597	240.3
			0.6	0.5	96	103	0.9593	0.501	0.126	564	243.6
			0.8	0.7	56	141	0.9668	0.704	0.111	394	260.6
			0.89	0.99	4	198	0.9621	0.990	0.026	429	257.1

6.2.3 Simulation Analysis for Triple Sampling Scheme when $p \neq p_0$

Table 10 Results of Triple Sampling Scheme when $p \neq p_0$

n_1	n_2	n_3	$1-\alpha$	a_1	b_1	a_2	b_2	p_0	p	Coverage Ratio	\hat{p}	Length of C.I.	Early stop	$E(\hat{n})$
60	30	30	0.95	0	2	0	2	0.12	0.02	0.9683	0.0205	0.061	242	109.71
				0	12	8	9	0.3	0.2	0.9797	0.206	0.179	990	78.48
				0	30	2	33	0.6	0.5	0.9530	0.510	0.220	996	76.53

				0	42	7	27	0.8	0.7	0.9596	0.711	0.200	1000	76.14
				0	60	23	87	0.89	0.99	0.9641	0.990	0.054	938	91.86
90	45	45	0.95	0	2	2	1	0.12	0.02	0.9691	0.021	0.057	1000	124.47
				0	18	5	11	0.3	0.2	0.9583	0.205	0.148	1000	115.65
				0	45	2	48	0.6	0.5	0.9687	0.506	0.181	1000	115.15
				0	63	2	62	0.8	0.7	0.9548	0.708	0.166	1000	113.72
				0	90	1	50	0.89	0.99	0.9697	0.990	0.042	1000	135
120	60	60	0.95	0	3	1	56	0.12	0.02	0.9695	0.022	0.044	257	211.02
				0	24	2	13	0.3	0.2	0.9654	0.204	0.128	1000	154.62
				0	60	4	64	0.6	0.5	0.9601	0.505	0.159	1000	151.74
				0	84	2	68	0.8	0.7	0.9591	0.706	0.144	1000	152.04
				0	119	0	164	0.89	0.99	0.9741	0.991	0.035	1000	162.24
150	75	75	0.95	0	3	4	4	0.12	0.02	0.9693	0.022	0.044	862	207.23
				0	30	3	15	0.3	0.2	0.9592	0.203	0.115	1000	192.15
				0	75	2	88	0.6	0.5	0.9592	0.505	0.143	1000	188.93
				0	105	2	107	0.8	0.7	0.9666	0.704	0.130	1000	190.5
				0	149	2	216	0.89	0.99	0.9624	0.990	0.031	998	208.05

One more observation is that compared to a single sampling scheme, the coverage ratios of double sampling and triple sampling schemes are higher. Such

observation supports the idea that double sampling and triple sampling plans are more robust than a single sampling plan. However, the general lower coverage ratios compared with nominal confidence levels indicate that the design specified in this study has room for improvements.

6.3 Discussions

The first objective of my study is to compare the accuracy of point estimator under different sampling schemes. Under a single sampling scheme all the results are very close to the expected proportion. Double sampling and triple sampling schemes exhibit the similar characteristic, though the estimates under triple sampling design tend to be slightly higher than the expected proportion due to possible early stopping. In general, the point estimate under a triple sampling plan produces good approximation of expected proportion.

The probability with which the expected proportion is in the calculated confidence interval is another factor to assess the efficiency and accuracy of sampling scheme. Under a single sampling scheme, above probability appears close to $1 - \alpha$ when $p = 0.2, 0.5, 0.7$. But, the coverage ratio deviates a lot when $p = 0.02, 0.99$. Such insufficient has been greatly compensated in double and triple sampling schemes. Thus, double and triple sampling designs provide higher confidence level over that under a single sampling design. When compared with a double sampling design, a triple sampling plan also gives higher achieved confidence level.

The average length of confidence intervals is implemented in the R. Generally speaking, the length of confidence interval is larger under a triple sampling

scheme than that under a single sampling scheme. The difference might be the correction for the sampling bias when experiment terminates at first stage or second stage (Ryan 2009). But, such an insignificant increase of length of confidence interval can be neglected given the significant improvement of coverage ratio.

7. Conclusions

A Triple sampling scheme provides good prediction of the expected proportion in most cases. The achieved confidence level is higher than $1 - \alpha$ under a triple sampling scheme for $p \in [0.3, 0.5]$ where the outcomes under a single sampling turn out to be not as good. Compared with a double sampling scheme, a triple sampling scheme has better coverage ratio and comparable length of confidence interval.

Furthermore, without compromise of the length of confidence interval and coverage ratio, the sample size of a triple sampling scheme is typically much smaller than the sample size of single sampling scheme or double sampling scheme. For example, when $p = 0.5$, in a single sampling design, $n = 300$, coverage ratio = 0.9455 and length of confidence interval = 0.113; in a double sampling design, $n = 242.9$, coverage ratio = 0.953 and length of confidence interval = 0.126; while in a triple sampling plan, $n = 192$, coverage ratio = 0.959 and length of confidence interval = 0.142. In this sense, a triple sampling plan possesses the feature of efficiency and economy.

The negative side of a triple sampling plan is that a triple sampling plan tends to produce biased, often higher, estimator of proportion p , which is also mentioned in

Ryan (2009). But, given that a triple sampling plan tends to terminate early more often than a double sampling plan, a triple sampling plan still provides cost benefits over a double sampling plan. A narrower first gate might produce more accurate estimator of p .

In summary, a triple sampling scheme is a good alternative of single sampling or double sampling plans. Because a triple sampling scheme terminates early more often than a double sampling scheme, triple sampling scheme is recommended for the practice of statistics.

8. References

- Agrawal, M.C. and Jain, Nirmal. (1988). Predictive Estimation in Double Sampling Procedures. *The American Statistician*, Vol. 42, No. 3, (Aug 1988), pp.184-186.
- Brown, Lawrence D. ; Cai, T. Tony and DasGupta, Anirban. (2001). Interval Estimation for a Binomial Proportion. *Statistical Science*, Vol 16, Issue 2 (2001), 101-133.
- Colton, Theodore; McPherson, Klim. (1976). Two-Stage Plans Compared with Fixed-Sample-Size and Wald SPRT Plans. *Journal of the American Statistical Association*, Vol. 71, No. 353 (Mar., 1976), pp. 80-86.
- Duffy, Diane E. and Santner, Thomas J. (1987). Confidence Intervals for a Binomial Parameter Based on Multistage Tests. *Biometrics*, Vol. 43, No. 1 (Mar., 1987), pp. 81-93.
- Hamaker, H.C. and Strik, R.Van. (1955). The Efficiency of Double Sampling for Attributes. *Journal of the American Statistical Association*, Vol. 50, No. 271 (Sep. 1955), pp.830-849.
- Jennison, Christopher and Turnbull, Bruce W. (1983). Confidence Intervals for a Binomial Parameter Following a Multistage Test with Application to MIL-STD 105D and Medical Trials. *Technometrics*, Vol. 25, No. 1 (Feb., 1983), pp. 49-58.
- Jung, Sin-Ho; Kim, Kyung Mann. (2004). On the Estimation of the Binomial Probability in Multistage Clinical Trials. *Statistics in Medicine*, Vol.23, Issue 6, pp.881 – 896.
- O'Brien, Peter C. and Fleming, Thomas R. (1979). A Multiple Testing Procedure for Clinical Trials. *Biometrics*, Vol. 35 (Sep. 1979), pp.549-556.
- Pocock, Stuart J. (1977). Group Sequential Methods in the Design and Analysis of Clinical Trials. *Biometrika*, Vol. 64, No. 2 (1977), pp. 191-199, printed in

Great Britain.

Ryan, Kenneth J. (2009). Approximate Confidence Interval for p When Double Sampling. *The American Statistician*, Vol. 63, No. 2 (May 2009), pp.132-140.

Wolfowitz, J. (1947). The Efficiency of Sequential Estimates and Wald's Equation for Sequential Processes. *The Annals of Mathematical Statistics*, Vol 18, No.2 (1947), pp. 215-230.

Appendix I R Program Codes for Single Sampling Scheme

```
Simulation.Single<-function(repeats, N, alpha, p){
  intherange <- 0
  pstorage <- NULL
  kappa <- qnorm(1-alpha/2)
  kappa2 <- kappa*kappa
  length <- 0
  for (i in 1: repeats){
    u <- runif(N)
    success <- 0
    for (j in 1: N){
      if (u[j] <= p){
        success <- success+1
      }
    }
    pestimator <- success/N
    pstorage[i] <- pestimator

    success_ <- success+kappa2/2
    N_ <- N+kappa2
    p_ <- success_/N_

    s1 <- sqrt(pestimator*(1-pestimator)+kappa2/4/N)
    s2 <- kappa*sqrt(N)/N_

    plower <- p_-s1*s2
    pupper <- p_+s1*s2
    if (p >= plower && p <= pupper){
      intherange <- intherange+1
      length <- length+pupper-plower
    }
  }
  confidence <- intherange/repeats
  pave <- mean(pstorage)
  length <- length/intherange
  print(confidence)
```

```
    print(pave)
    print(length)
}
```

```
repeats <- 10000
N <- 17000
alpha <- 0.01
p <- 0.99
```

```
Simulation.Single(repeats, N, alpha, p)
```

* In R program codes above:
repeats = times of repetitions, N = sample size,
alpha = confidence level, p = success proportion.

Appendix II R Program Codes for Computations of Table 3 and Table 4

```
Integral<-function(N1, N2, a1, b1, a2, b2, p){
  temp1 <- pbinom(b1+1, N1,p)
  A1 <- 1-temp1
  B1 <- pbinom(a1-1, N1, p)

  ### a1+N2 > b2+1
  A2 <- 0
  for (s in (b2+1): (b1+N2)){
    for (ps in a1 : b1){
      A2 <- A2+dbinom(ps, N1, p)*dbinom(s-ps, N2, p)
    }
  }

  ### a2-1 < b1
  B2 <- 0
  for (s in a1 : a2-1){
    for (ps in a1: a2-1){
      B2 <- B2+dbinom(ps, N1, p)*dbinom(s-ps, N2, p)
    }
  }
  C <- 0
  for (s in a1: a2-1){
    for (ps in a1: b1){
      C <- C+dbinom(ps, N1, p)*dbinom(s-ps, N2, p)
    }
  }
  sum <- A1+B1+A2+B2
  sum2 <- A1+B1+C
  print(A1)
  print(B1)
  print(A2)
  print(B2)
  print(sum)
  print(C)
```

```
print(sum2)
print(A1+B1)
}
```

```
N1 <- 12
N2 <- 12
a1 <- 11
b1 <- 12
a2 <- 14
b2 <- 16
p <- 0.99
```

```
Integral(N1, N2, a1, b1, a2, b2, p)
```

Appendix III R Program Codes for Double Sampling Scheme

```
boundsolver<-function(N1, N2, a, b, p, alpha, success, indicator){
  tolerance <- 1e-9
  tail <- alpha/2
  dif <- 1
  if (indicator == 1){
    high <- p; low <- 0; plower <- low

    pa <- 1-pnorm(b, mean = N1*plower, sd = N1*plower*(1-plower))
    f <- function(x) dnorm(x, mean = N1*plower, sd =
N1*plower*(1-plower))*(1-pnorm(success-x, mean = N2*plower, sd =
N2*plower*(1-plower)))
    pb <- integrate(Vectorize(f), a, b)$value
    pr <- pa+pb-tail

    while((abs(pr) > tolerance) && (abs(dif) > 1e-3)){
      pah <- 1-pnorm(b, mean = N1*high, sd = N1*high*(1-high))
      g <- function(x) dnorm(x, mean = N1*high, sd =
N1*high*(1-high))*(1-pnorm(success-x, mean = N2*high, sd = N2*high*(1-high)))
      pbh <- integrate(Vectorize(g), a, b)$value
      prh <- pah+pbh-tail

      pre <- plower
      if (pr*prh < 0){
        low <- plower
        plower <- (high+low)/2
        pa <- 1-pnorm(b, mean = N1*plower, sd = N1*plower*(1-plower))
        f <- function(x) dnorm(x, mean = N1*plower, sd =
N1*plower*(1-plower))*(1-pnorm(success-x, mean = N2*plower, sd =
N2*plower*(1-plower)))
        pb <- integrate(Vectorize(f), a, b)$value
        pr <- pa+pb-tail
      }else{
        high <- plower
        plower <- (high+low)/2
        pa <- 1-pnorm(b, mean = N1*plower, sd = N1*plower*(1-plower))
```

```

        f <- function(x) dnorm(x, mean = N1*plower, sd =
N1*plower*(1-plower))*(1-pnorm(success-x, mean = N2*plower, sd =
N2*plower*(1-plower)))
        pb <- integrate(Vectorize(f), a, b)$value
        pr <- pa+pb-tail
    }
    pcurr <- plower
    dif <- pcurr-pre
}
plower
}else{
    high <- 1; low <- p; pupper <- high

    pa <- pnorm(a, mean = N1*pupper, sd = N1*pupper*(1-pupper))
    f <- function(x) dnorm(x, mean = N1*pupper, sd =
N1*pupper*(1-pupper))*pnorm(success-x, mean = N2*pupper, sd =
N2*pupper*(1-pupper))
    pb <- integrate(Vectorize(f), a, b)$value
    pr <- pa+pb-tail

    while((abs(pr) > tolerance) && (abs(dif) > 1e-3)){
        pah <- pnorm(a, mean = N1*high, sd = N1*high*(1-high))
        g <- function(x) dnorm(x, mean = N1*high, sd =
N1*high*(1-high))*pnorm(success-x, mean = N2*high, sd = N2*high*(1-high))
        pbh <- integrate(Vectorize(g), a, b)$value
        prh <- pah+pbh-tail

        pre <- pupper
        if (pr*prh < 0){
            low <- pupper
            pupper <- (high+low)/2
            pa <- pnorm(a, mean = N1*pupper, sd = N1*pupper*(1-pupper))
            f <- function(x) dnorm(x, mean = N1*pupper, sd =
N1*pupper*(1-pupper))*pnorm(success-x, mean = N2*pupper, sd =
N2*pupper*(1-pupper))
            pb <- integrate(Vectorize(f), a, b)$value
            pr <- pa+pb-tail
        }else{

```



```

        high <- pupper
        pupper <- (high+low)/2
        pa <- pnorm(a, mean = N1*pupper, sd = N1*pupper*(1-pupper))
        f <- function(x) dnorm(x, mean = N1*pupper, sd =
N1*pupper*(1-pupper))*pnorm(success-x, mean = N2*pupper, sd =
N2*pupper*(1-pupper))
        pb <- integrate(Vectorize(f), a, b)$value
        pr <- pa+pb-tail
    }
    pcurr <- pupper
    dif <- pcurr-pre
}
pupper
}
}

```

```
Simulation.Double<-function(repeats, N1, N2, a, b, alpha, p){
```

```
### Some initialization
```

```

intherange <- 0
pstorage <- NULL

```

```

kappa <- qnorm(1-alpha/2)
kappa2 <- kappa*kappa
lower <- 1
upper <- 2
stagecount <- 0
totallength <- 0

```

```
### The main simulation part
```

```

for (i in 1: repeats){
  print(i)
  u1 <- runif(N1)
  success <- 0
  for (j in 1: N1){
    if (u1[j] <= p){
      success <- success+1
    }
  }
}

```

```

    }
  }
  pestimator <- success/N1
  stage1 <- 1

  if (success <= b && success >= a){
    u2 <- runif(N2)
    for (k in 1: N2){
      if (u2[k] <= p){
        success <- success+1
      }
    }
    pestimator <- success/(N1+N2)
    stage1 <- 0
  }
  pstorage[i] <- pestimator
  if (stage1 == 1){
    stagecount <- stagecount+1
    success_ <- success+kappa2/2
    N_ <- N1+kappa2
    p_ <- success_/N_
    s1 <- sqrt(pestimator*(1-pestimator)+kappa2/4/N1)
    s2 <- kappa*sqrt(N1)/N_
    plower <- p_-s1*s2
    pupper <- p_+s1*s2
  }else{
    plower <- boundsolver(N1, N2, a, b, pestimator, alpha, success, lower)
    pupper <- boundsolver(N1, N2, a, b, pestimator, alpha, success, upper)
  }
  totallength <- totallength+pupper-plower
  if (p >= plower && p <= pupper){
    intherange <- intherange+1
  }
}
print(totallength/repeats)
confidence <- intherange/repeats
pave <- mean(pstorage)
print(confidence)

```

```
print(pave)
print("Stop at 1st stage")
print(stagecount)
}
```

```
repeats <- 10000
```

```
N1 <- 12
```

```
N2 <- 12
```

```
a <- 0
```

```
b <- 5
```

```
alpha <- 0.01
```

```
p <- 0.2
```

```
Simulation.Double(repeats, N1, N2, a, b, alpha, p)
```

Appendix IV R Program Codes for Triple Sampling Scheme

```
boundsolver<-function(N1, N2, a, b, p, alpha, success, indicator){
  tolerance <- 1e-9
  tail <- alpha/2
  dif <- 1
  if (indicator == 1){
    high <- p; low <- 0; plower <- low

    pa <- 1-pnorm(b, mean = N1*plower, sd = N1*plower*(1-plower))
    f <- function(x) dnorm(x, mean = N1*plower, sd =
N1*plower*(1-plower))*(1-pnorm(success-x, mean = N2*plower, sd =
N2*plower*(1-plower)))
    pb <- integrate(Vectorize(f), a, b)$value
    pr <- pa+pb-tail

    while((abs(pr) > tolerance) && (abs(dif) > 1e-3)){
      pah <- 1-pnorm(b, mean = N1*high, sd = N1*high*(1-high))
      g <- function(x) dnorm(x, mean = N1*high, sd =
N1*high*(1-high))*(1-pnorm(success-x, mean = N2*high, sd = N2*high*(1-high)))
      pbh <- integrate(Vectorize(g), a, b)$value
      prh <- pah+pbh-tail

      pre <- plower
      if (pr*prh < 0){
        low <- plower
        plower <- (high+low)/2
        pa <- 1-pnorm(b, mean = N1*plower, sd = N1*plower*(1-plower))
        f <- function(x) dnorm(x, mean = N1*plower, sd =
N1*plower*(1-plower))*(1-pnorm(success-x, mean = N2*plower, sd =
N2*plower*(1-plower)))
        pb <- integrate(Vectorize(f), a, b)$value
        pr <- pa+pb-tail
      }else{
        high <- plower
        plower <- (high+low)/2
        pa <- 1-pnorm(b, mean = N1*plower, sd = N1*plower*(1-plower))
```

```

        f <- function(x) dnorm(x, mean = N1*plower, sd =
N1*plower*(1-plower))*(1-pnorm(success-x, mean = N2*plower, sd =
N2*plower*(1-plower)))
        pb <- integrate(Vectorize(f), a, b)$value
        pr <- pa+pb-tail
    }
    pcurr <- plower
    dif <- pcurr-pre
}
plower
}else{
    high <- 1; low <- p; pupper <- high

    pa <- pnorm(a, mean = N1*pupper, sd = N1*pupper*(1-pupper))
    f <- function(x) dnorm(x, mean = N1*pupper, sd =
N1*pupper*(1-pupper))*pnorm(success-x, mean = N2*pupper, sd =
N2*pupper*(1-pupper))
    pb <- integrate(Vectorize(f), a, b)$value
    pr <- pa+pb-tail

    while((abs(pr) > tolerance) && (abs(dif) > 1e-3)){
        pah <- pnorm(a, mean = N1*high, sd = N1*high*(1-high))
        g <- function(x) dnorm(x, mean = N1*high, sd =
N1*high*(1-high))*pnorm(success-x, mean = N2*high, sd = N2*high*(1-high))
        pbh <- integrate(Vectorize(g), a, b)$value
        prh <- pah+pbh-tail

        pre <- pupper
        if (pr*prh < 0){
            low <- pupper
            pupper <- (high+low)/2
            pa <- pnorm(a, mean = N1*pupper, sd = N1*pupper*(1-pupper))
            f <- function(x) dnorm(x, mean = N1*pupper, sd =
N1*pupper*(1-pupper))*pnorm(success-x, mean = N2*pupper, sd =
N2*pupper*(1-pupper))
            pb <- integrate(Vectorize(f), a, b)$value
            pr <- pa+pb-tail
        }else{

```

```

        high <- pupper
        pupper <- (high+low)/2
        pa <- pnorm(a, mean = N1*pupper, sd = N1*pupper*(1-pupper))
        f <- function(x) dnorm(x, mean = N1*pupper, sd =
N1*pupper*(1-pupper))*pnorm(success-x, mean = N2*pupper, sd =
N2*pupper*(1-pupper))
        pb <- integrate(Vectorize(f), a, b)$value
        pr <- pa+pb-tail
    }
    pcurr <- pupper
    dif <- pcurr-pre
}
pupper
}
}

```

```

boundsolverT<-function(N1, N2, N3, a1, b1, a2, b2, p, alpha, success, indicator){
  tolerance <- 1e-9
  tail <- alpha/2
  dif <- 1
  if (indicator == 1){
    high <- p; low <- 0; plower <- low

    pa <- 1-pnorm(b1, mean = N1*plower, sd = N1*plower*(1-plower))
    f <- function(x) dnorm(x, mean = N1*plower, sd =
N1*plower*(1-plower))*(1-pnorm(b2-x, mean = N2*plower, sd =
N2*plower*(1-plower)))
    pb <- integrate(Vectorize(f), a1, b1)$value
    pc <- integrate(function(y){
      sapply(y, function(y){
        integrate(function(x) dnorm(x, mean=N1*plower,
sd=N1*plower*(1-plower) )*dnorm(y-x, mean=N2*plower,
sd=N2*plower*(1-plower)), a1, b1)$value*(1-pnorm(success-y, mean = N3*plower,
sd = N3*plower*(1-plower)))
      })), a2, b2)$value

    pr <- pa+pb+pc-tail
  }
}

```

```

while((abs(pr) > tolerance) && (abs(dif) > 1e-3)){
  pah <- 1-pnorm(b1, mean = N1*high, sd = N1*high*(1-high))
  ff <- function(x) dnorm(x, mean = N1*high, sd =
N1*high*(1-high))*(1-pnorm(b2-x, mean = N2*high, sd = N2*high*(1-high)))
  pbh <- integrate(Vectorize(ff), a1, b1)$value
  pch <- integrate(function(y){
    sapply(y, function(y){
      integrate(function(x) dnorm(x, mean=N1*high,
sd=N1*high*(1-high) )*dnorm(y-x, mean=N2*high, sd=N2*high*(1-high)), a1,
b1)$value*(1-pnorm(success-y, mean = N3*high, sd = N3*high*(1-high)))
    })), a2, b2)$value

  prh <- pah+pbh+pch-tail

  pre <- plower
  if (pr*prh < 0){
    low <- plower
    plower <- (high+low)/2
    pa <- 1-pnorm(b1, mean = N1*plower, sd = N1*plower*(1-plower))
    f <- function(x) dnorm(x, mean = N1*plower, sd =
N1*plower*(1-plower))*(1-pnorm(b2-x, mean = N2*plower, sd =
N2*plower*(1-plower)))
    pb <- integrate(Vectorize(f), a1, b1)$value
    pc <- integrate(function(y){
      sapply(y, function(y){
        integrate(function(x) dnorm(x, mean=N1*plower,
sd=N1*plower*(1-plower) )*dnorm(y-x, mean=N2*plower,
sd=N2*plower*(1-plower)), a1, b1)$value*(1-pnorm(success-y, mean = N3*plower,
sd = N3*plower*(1-plower)))
      })), a2, b2)$value
    pr <- pa+pb+pc-tail
  }else{
    high <- plower
    plower <- (high+low)/2
    pa <- 1-pnorm(b1, mean = N1*plower, sd = N1*plower*(1-plower))
    f <- function(x) dnorm(x, mean = N1*plower, sd =
N1*plower*(1-plower))*(1-pnorm(b2-x, mean = N2*plower, sd =
N2*plower*(1-plower)))

```

```

pb <- integrate(Vectorize(f), a1, b1)$value
pc <- integrate(function(y){
  sapply(y, function(y){
    integrate(function(x) dnorm(x, mean=N1*plover,
sd=N1*plover*(1-plover) )*dnorm(y-x, mean=N2*plover,
sd=N2*plover*(1-plover)), a1, b1)$value*(1-pnorm(success-y, mean = N3*plover,
sd = N3*plover*(1-plover)))
  })), a2, b2)$value
pr <- pa+pb+pc-tail
}
pcurr <- plover
dif <- pcurr-pre
}
plover
}else{
high <- 1; low <- p; pupper <- high

pa <- pnorm(a1, mean = N1*pupper, sd = N1*pupper*(1-pupper))
f <- function(x) dnorm(x, mean = N1*pupper, sd =
N1*pupper*(1-pupper))*pnorm(a2-x, mean = N2*pupper, sd =
N2*pupper*(1-pupper))
pb <- integrate(Vectorize(f), a1, b1)$value
pc <- integrate(function(y){
  sapply(y, function(y){
    integrate(function(x) dnorm(x, mean=N1*pupper,
sd=N1*pupper*(1-pupper) )*dnorm(y-x, mean=N2*pupper,
sd=N2*pupper*(1-pupper)), a1, b1)$value*pnorm(success-y, mean = N3*pupper, sd
= N3*pupper*(1-pupper))
  })), a2, b2)$value

pr <- pa+pb+pc-tail

while((abs(pr) > tolerance) && (abs(dif) > 1e-3)){
  pah <- pnorm(a1, mean = N1*high, sd = N1*high*(1-high))
  g <- function(x) dnorm(x, mean = N1*high, sd =
N1*high*(1-high))*pnorm(success-x, mean = N2*high, sd = N2*high*(1-high))
  pbh <- integrate(Vectorize(g), a1, b1)$value
  pch <- integrate(function(y){

```



```

        sapply(y, function(y){
            integrate(function(x) dnorm(x, mean=N1*high,
sd=N1*high*(1-high) )*dnorm(y-x, mean=N2*high, sd=N2*high*(1-high)), a1,
b1)$value*pnorm(success-y, mean = N3*high, sd = N3*high*(1-high))
        })), a2, b2)$value

prh <- pah+pbh+pch-tail

pre <- pupper
if (pr*prh < 0){
    low <- pupper
    pupper <- (high+low)/2
    pa <- pnorm(a1, mean = N1*pupper, sd = N1*pupper*(1-pupper))
    f <- function(x) dnorm(x, mean = N1*pupper, sd =
N1*pupper*(1-pupper))*pnorm(a2-x, mean = N2*pupper, sd =
N2*pupper*(1-pupper))
    pb <- integrate(Vectorize(f), a1, b1)$value
    pc <- integrate(function(y){
        sapply(y, function(y){
            integrate(function(x) dnorm(x, mean=N1*pupper,
sd=N1*pupper*(1-pupper) )*dnorm(y-x, mean=N2*pupper,
sd=N2*pupper*(1-pupper)), a1, b1)$value*pnorm(success-y, mean = N3*pupper, sd
= N3*pupper*(1-pupper))
        })), a2, b2)$value

    pr <- pa+pb+pc-tail
}else{
    high <- pupper
    pupper <- (high+low)/2
    pa <- pnorm(a1, mean = N1*pupper, sd = N1*pupper*(1-pupper))
    f <- function(x) dnorm(x, mean = N1*pupper, sd =
N1*pupper*(1-pupper))*pnorm(a2-x, mean = N2*pupper, sd =
N2*pupper*(1-pupper))
    pb <- integrate(Vectorize(f), a1, b1)$value
    pc <- integrate(function(y){
        sapply(y, function(y){
            integrate(function(x) dnorm(x, mean=N1*pupper,
sd=N1*pupper*(1-pupper) )*dnorm(y-x, mean=N2*pupper,

```

```

sd=N2*pupper*(1-pupper)), a1, b1)$value*pnorm(success-y, mean = N3*pupper, sd
= N3*pupper*(1-pupper))
      }}, a2, b2)$value

      pr <- pa+pb+pc-tail
    }
    pcurr <- pupper
    dif <- pcurr-pre
  }
  pupper
}
}

```

```
Simulation.Triple<-function(repeats, N1, N2, N3, a1, b1, a2, b2, alpha, p){
```

```
### Some initialization
```

```

intherange <- 0
pstorage <- NULL

```

```

kappa <- qnorm(1-alpha/2)
kappa2 <- kappa*kappa
lower <- 1
upper <- 2
stagecount <- 0
totallength <- 0

```

```
###
```

```
### The main simulation part
```

```

for (i in 1: repeats){
  print(i)
  u1 <- runif(N1)
  success <- 0
  for (j in 1: N1){
    if (u1[j] <= p){
      success <- success+1
    }
  }
}

```

```

}
pestimator <- success/N1
stage <- 1

if (success <= b1 && success >= a1){
  u2 <- runif(N2)
  for (k in 1: N2){
    if (u2[k] <= p){
      success <- success+1
    }
  }
  pestimator <- success/(N1+N2)
  stage <- 2

  if (success <= b2 && success >= a2){
    u3 <- runif(N3)
    for (t in 1: N3){
      if (u2[t] <= p){
        success <- success+1
      }
    }
    pestimator <- success/(N1+N2+N3)
    stage <- 3
  }
}

pstorage[i] <- pestimator
if (stage == 1){
  stagecount <- stagecount+1
  success_ <- success+kappa2/2
  N_ <- N1+kappa2
  p_ <- success_/N_
  s1 <- sqrt(pestimator*(1-pestimator)+kappa2/4/N1)
  s2 <- kappa*sqrt(N1)/N_
  plower <- p_-s1*s2
  pupper <- p_+s1*s2
}

```

```

    if (stage == 2){
      stagecount <- stagecount+1
      plower <- boundsolver(N1, N2, a1, b1, pestimator, alpha, success, lower)
      pupper <- boundsolver(N1, N2, a1, b1, pestimator, alpha, success, upper)
    }

    if (stage == 3){
      plower <- boundsolverT(N1, N2, N3, a1, b1, a2, b2, pestimator, alpha,
success, lower)
      pupper <- boundsolverT(N1, N2, N3, a1, b1, a2, b2, pestimator, alpha,
success, upper)
    }
    if (p >= plower && p <= pupper){
      totallength <- totallength+pupper-plower
      intherange <- intherange+1
    }
  }
  print(totallength/intherange)
  confidence <- intherange/repeats
  pave <- mean(pstorage)
  print(confidence)
  print(pave)
  print("Stop at 1st stage")
  print(stagecount)
}

```

```
repeats <- 10000
```

```
N1 <- 96
```

```
N2 <- 96
```

```
N3 <- 96
```

```
a1 <- 0
```

```
b1 <- 10
```

```
a2 <- 0
```

```
b2 <- 20
```

```
alpha <- 0.01
```

```
p <- 0.02
```

```
Simulation.Triple(repeats, N1, N2, N3, a1, b1, a2, b2, alpha, p)
```