ASSESSING A BINARY MEASUREMENT SYSTEM

Oana Danila, Stefan H. Steiner and R. Jock MacKay

Business and Industrial Statistics Research Group

Dept. of Statistics and Actuarial Sciences

University of Waterloo

Waterloo, N2L 3G1 Canada

This report was submitted as **Oana Danila**'s Masters Essay in Statistics

1. Introduction

Binary measurement systems (BMS) are widely used in industry and it is of interest to assess their performance in terms of misclassification rates. A BMS classifies items as conforming and passes them or as nonconforming and rejects them. Simultaneously, we may want to separate the effects of the measurement system and the rest of the process in producing defective items.

2. Credit card example

An example of a BMS is one in which blank credit cards are passed or rejected by an automated visual inspection system. The cards are checked for many defects, such as missing parts, surface scratches, bleeding of colors, fuzzy letters and numbers, etc. The system takes a digital picture of the front of each card and calculates hundreds of summary measures based on comparing the picture to a template of the ideal card. If any of the summary measures falls outside a pre-specified range, the card is rejected. Cards can also be classified by a human operator who carefully analyzes the credit cards and classifies them as conforming or nonconforming. We assume that the human inspector is the "gold standard" measurement system that determines the true status of the cards with no classification error.

3. Misclassification rates

If each item's true state is known, we can examine the performance of a BMS by considering the misclassification rates.

There are two kinds of mistakes a BMS could make. First, conforming items are mistakenly classified as nonconforming and second, nonconforming items are classified as conforming. Usually, for a production process, the latter misclassification probability is of a greater interest, as a high rate of classifying nonconforming items as conforming could mean that many defective items reach the customer.

To estimate misclassification rates we need to know both the BMS classification for a certain number of items and also the items' true state given by a "gold" standard measurement system. The cost of each measurement with the gold standard system is usually high. We can summarize the results of classifying N items by each measurement system as given in Table1.

Table 1: Data from assessing a BMS performance

	conform (C)	not conform ($\bar{C}$)	Total items
pass (P)	n_{PC}	$n_{P\bar{C}}$	n_P
reject ($\bar{P}$)	$n_{\bar{P}C}$	$n_{\bar{P}\bar{C}}$	$n_{\bar{P}}$
Total items	n_C	$n_{\bar{C}}$	N

Using the notation given in the table, we define $p_C = \Pr(C)$, the probability of a conforming item and $p_P = \Pr(P)$, the probability of passing an item. Note that p_C is determined by the quality of the production process and is not related to the measurement system performance.

We also define a set of conditional probabilities that characterize the quality of the measurement system. Let $\Pr(P/\bar{C}) = a$ be the probability of accepting a nonconforming item and $\Pr(\bar{P}/C) = b$ be the probability of rejecting a conforming item.

We assume the pass rate, p_P, is known. In other words, we assume the system has been operating for a while and information has been gathered to compute a good estimate of p_P. The assumption that p_P is known will be used in all the proposed methods for assessing BMS performance.

The assumption that p_P is known imposes restrictions on the conditional probabilities, a and b, and p_C.

In particular, we have $p_P = (1-b)p_C + a(1-p_C)$ and solving for p_C, we get: $p_C = \dfrac{p_P - a}{1-b-a}$. Since $0 \le p_C \le 1$, $0 \le \dfrac{p_P - a}{1-b-a} \le 1$. It is reasonable to assume that $1-b-a > 0$, as a and b are usually small. Therefore, we have two restrictions, $a \le p_P$ and $b \le 1-p_P$.

4. *Analysis goals*

We have mentioned above that p_C characterizes the performance of the production process and a and b quantify the performance of the measurement system. We consider two different goals when assessing a BMS. The first is to estimate p_C or the performance of the production process, and the second is to estimate a and b or the performance of the BMS itself.

The goal of this paper is to propose different approaches for estimating a , b and p_C , and then compare them using several criteria. Sampling plans differ from method to method and all approaches use Maximum Likelihood (ML) Estimation. To start, we describe each of the methods, focusing on the sampling scheme and variance calculation for the estimators of a , b and p_C under that particular scheme. For each method, parameters estimators and their corresponding standard deviations will be obtained using both theoretical and simulation methods. Then, we will present the advantages and disadvantages by comparing the efficiency of the estimators, the costs and the practicality of each method.

5. *Methods*

In practice, there are two kinds of BMS. There are systems that record whether an item is within some specification limits or not, so the binary output is the result of an underlying continuous output. There are also systems that classify items as acceptable or not using a large number of criteria.

The Automotive Industry Action Group (AIAG) proposed a method for assessing the first type of BMS's and this method is largely used in practice. Our goal is to estimate the performance of BMS for the second type of outputs. Therefore, the AIAG method is not applicable in this case. There is only one method that has been suggested for assessing the performance of a BMS from the second category. This method was proposed by Farnum in 1994 and from now on is going to be referred to as the "Farnum's Plan". The author did not assume that the passing rate, p_P , is known and originally this information was not used in estimation. Below we describe the proposed sampling plan and use p_P in estimating p_C .

Plan I (Farnum)

The sampling plan for this approach is to select two equally-sized independent samples of conforming and nonconforming items ($n_C = n_{\bar{C}}$) previously assessed by the gold standard measurement system. Then, the sampled items are evaluated by the BMS. Therefore, in *Table 1*, n_C , $n_{\bar{C}}$ and N are fixed and all the other quantities are random.

A strength of this sampling scheme is that we can directly estimate the misclassification probabilities.

The ML estimators for a , b and p_C are denoted $\hat{a}(1)$, $\hat{b}(1)$ and $\hat{p}_C(1)$, respectively. We obtain $\hat{a}(1) = \frac{n_{P\bar{C}}}{n_{\bar{C}}}$ and $\hat{b}(1) = \frac{n_{\bar{P}C}}{n_C}$. Both estimators are unbiased and their variances depend on $n_{\bar{C}}$ and n_C , respectively.

We also estimate p_C using the estimators for a and b , and the fact that p_P is known: $\hat{p}_C(1) = \frac{p_P - \hat{a}(1)}{1 - \hat{b}(1) - \hat{a}(1)}$.

However, Farnum's Plan is not practical, since finding a large sample of conforming and (especially) nonconforming items requires that the true state of many items be determined. In the case of a high conforming probability, getting a large number of nonconforming items would be unreasonably expensive.

We do not consider this method in further investigations and comparisons. We will propose another two methods that are easily applied and for which the costs involved are relatively lower.

Plan II (Binomial)

We select two independent samples, one containing n_P items from the population of passed items and another of size $n_{\bar{P}}$ from the population of rejected items. Then, all the selected items are inspecting using the "gold standard" measurement system and their true state – conforming or nonconforming - is determined. In *Table 1*, n_P, $n_{\bar{P}}$ and N are fixed and all the other quantities are random.

In this case, we cannot directly estimate the two misclassification probabilities, a and b. Instead, we start with the ML estimates: $\hat{\Pr}(\bar{C}/P) = \dfrac{n_{P\bar{C}}}{n_P}$ and $\hat{\Pr}(\bar{C}/\bar{P}) = \dfrac{n_{\bar{P}\bar{C}}}{n_{\bar{P}}}$.

By Bayes' Rule:

$$a = \frac{\Pr(P \cap \bar{C})}{\Pr(\bar{C})} = \frac{\Pr(\bar{C}/P)\Pr(P)}{\Pr(\bar{C}/P)\Pr(P) + \Pr(\bar{C}/\bar{P})\Pr(\bar{P})}$$

and

$$b = \frac{\Pr(\bar{P} \cap C)}{\Pr(C)} = \frac{\Pr(C/\bar{P})\Pr(\bar{P})}{\Pr(C/\bar{P})\Pr(\bar{P}) + \Pr(C/P)\Pr(P)}$$

We also have $p_C = \Pr(C \cap P) + \Pr(C \cap \bar{P}) = \Pr(C/P)\Pr(P) + \Pr(C/\bar{P})\Pr(\bar{P})$

Using the invariance property of the ML estimators we obtain the following estimates for a, b and p_C from Plan II:

$$\hat{a}(2) = \frac{p_P \dfrac{n_{P\bar{C}}}{n_P}}{p_P \dfrac{n_{P\bar{C}}}{n_P} + (1-p_P)\dfrac{n_{\bar{P}\bar{C}}}{n_{\bar{P}}}} = \frac{p_P n_{P\bar{C}} n_{\bar{P}}}{p_P n_{P\bar{C}} n_{\bar{P}} + (1-p_P) n_{\bar{P}\bar{C}} n_P}$$

P PC

P(

PC

n P

Therefore, $E(\frac{N_{P\bar{C}}}{n_P}) = \Pr(\bar{C}/P)$, $Var(\frac{N_{P\bar{C}}}{n_P}) = \frac{\Pr(\bar{C}/P)(1-\Pr(\bar{C}/P))}{n_P}$ and $E(\frac{N_{\bar{P}\bar{C}}}{n_{\bar{P}}}) = \Pr(\bar{C}/\bar{P})$,

$Var(\frac{N_{\bar{P}\bar{C}}}{n_{\bar{P}}}) = \frac{\Pr(\bar{C}/\bar{P})(1-\Pr(\bar{C}/\bar{P}))}{n_{\bar{P}}}$

The biases for both $\hat{a}(2)$ and $\hat{b}(2)$ are of order $1/N$, and an approximation given by the $d-method$ can be found in *Appendix 1*. Note that $\hat{p}_C(2)$ is an unbiased estimator.

We can also get approximations for estimators' variances in terms of a, b and p_P (known) using the $d-method$ (for details, see *Appendix 1*):

$$Var(\hat{a}(2)) \simeq \frac{a(1-a)(p_P-a)}{1-b-p_P}\left(\frac{1-a-b+ab}{n_P}+\frac{ab}{n_{\bar{P}}}\right)$$

$$Var(\hat{b}(2)) \simeq \frac{b(1-b)(1-b-p_P)}{p_P-a}\left(\frac{ab}{n_P}+\frac{1-b-a+ab}{n_{\bar{P}}}\right)$$

The variance of $\hat{p}_C(2)$ can be directly derived and the details are also included in *Appendix 1*.

$$Var(\hat{p}(2)) = \frac{(1-b-p_P)(p_P-a)}{(1-a-b)^2}\left(\frac{a(1-b)}{n_P}+\frac{b(1-a)}{n_{\bar{P}}}\right)$$

These approximations were computed for the general case of two independent samples of different sizes (n_P and $n_{\bar{P}}$). The purpose of considering first the case of different sample sizes is to find the optimal allocation of n_P and $n_{\bar{P}} = N - n_P$ for fixed N. Our goal is to minimize the standard deviations.

Figures 1 – 2 illustrate how the standard deviations of the estimators vary with the proportion n_P/N of the passed items, for some specific values of a, b and p_P.

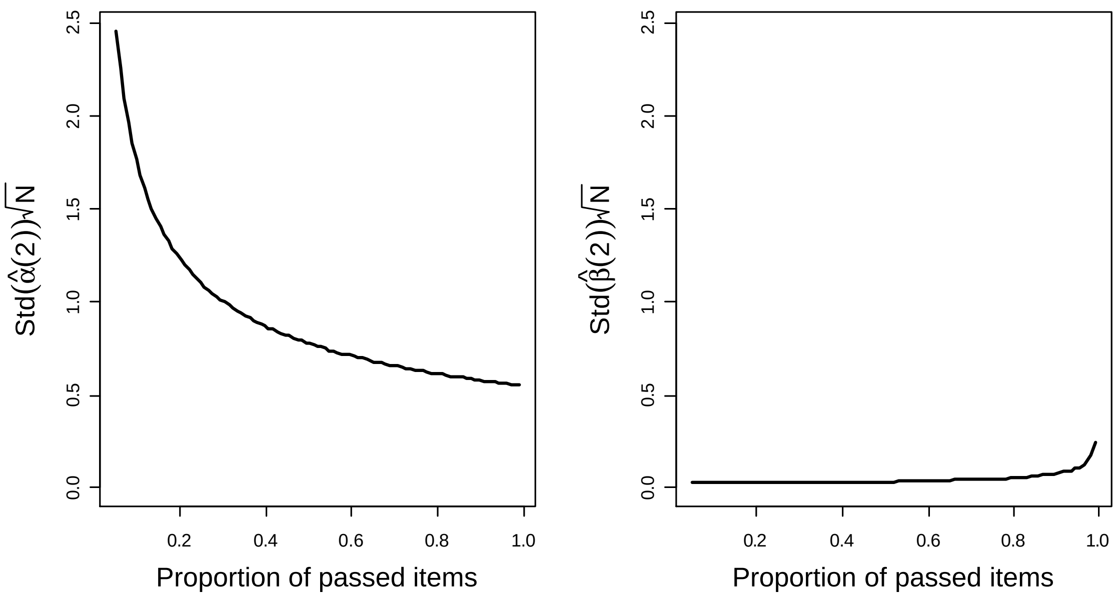

Figure1: $Std(\hat{a}(2))\sqrt{N}$ *and* $std(\hat{b}(2))\sqrt{N}$ *as functions of the proportion of passed items*
Total sample size is N *, and* $a = 0.01$, $b = 0.02$, $p_P = 0.95$

We notice that $Std(\hat{a}(2))\sqrt{N}$ decreases slowly - from 0.86 to 0.57 - over the interval $[0.4;1)$, the function being almost flat over this range. Therefore, to estimate a, selecting passed items anywhere from 40% to 99% gives roughly the same results.

$Std(\hat{b}(2))\sqrt{N}$ increases very slowly over the entire interval for the proportion of passed items. If the main goal of the study is to estimate a and b, we can select as many as 75% passed items and 25% rejected items and have almost the same precision for the estimates as when selecting 50% accepted and 50% rejected. This result makes Plan II very appealing, as it may be difficult to select a large number of rejected items, $1-p_P$ usually being small.

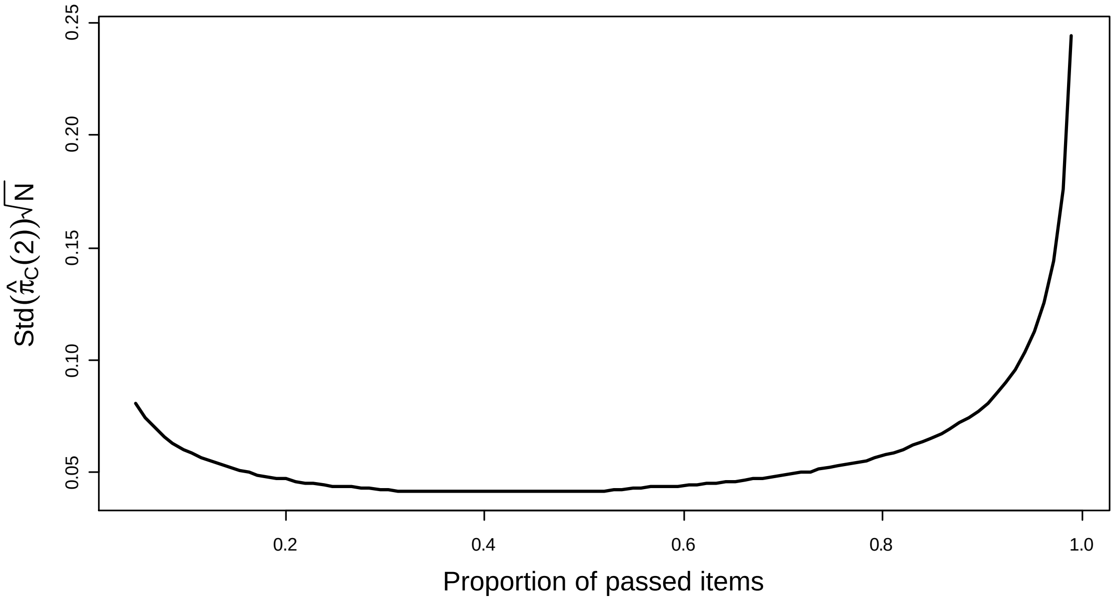

Figure2: $Std(\hat{p}_C(2))\sqrt{N}$ *as a function of the proportion of passed items sample size in the total sample size,* N *, when* $a = 0.01$, $b = 0.02$, $p_P = 0.95$

$Std(\hat{p}_C(2))\sqrt{N}$ varies from 0.047 to 0.057 over the interval $[0.2;0.8]$ for the proportion of passed items. The function is very flat from 0.25 to 0.75. Therefore, selecting 75% passed items gives almost the same precision as selecting 25% passed items. This fact also makes

For estimating a , b and p_C we use the ML method, the fact that p_P is known and also the invariance property of the ML estimators. The detailed derivations of the ML estimation using Lagrange Multipliers can be found in *Appendix 2*.

We obtain the following estimates for a , b and p_C:

$$\hat{a}(3) = \frac{p_P n_{P\bar{C}}(n_{\bar{P}C} + n_{\bar{P}\bar{C}})}{p_P n_{P\bar{C}}(n_{\bar{P}C} + n_{\bar{P}\bar{C}}) + (1-p_P) n_{\bar{P}\bar{C}}(n_{PC} + n_{P\bar{C}})}$$

$$\hat{b}(3) = \frac{(1-p_P) n_{\bar{P}C}(n_{PC} + n_{P\bar{C}})}{(1-p_P) n_{\bar{P}C}(n_{PC} + n_{P\bar{C}}) + p_P n_{PC}(n_{\bar{P}C} + n_{\bar{P}\bar{C}})}$$

and

$$\hat{p}_C(3) = \frac{p_P n_{PC}}{n_{PC} + n_{P\bar{C}}} + \frac{(1-p_P) n_{\bar{P}C}}{n_{\bar{P}C} + n_{\bar{P}\bar{C}}}$$

The a and b estimators given by this method have biases of order $1/N$, and the p_C estimator is unbiased. We use the $d-method$ to derive approximations of estimators' variances and biases. The derivations and resulting functions are included in *Appendix 2* and the Maple code used for these derivations is included in *Appendix 3*.

So far we have used the $d-method$ to get the approximations for estimators' biases and variances, for each of the proposed methods. We also ran a 10,000 trial simulation using the R-software and used the sample means as estimates for a , b and p_C and the sample variances as estimates of the variances of $\hat{a}(2)$, $\hat{b}(2)$ and $\hat{p}_C(2)$. As we will see later on, these simulation results agree with the theoretical approximations. The R-code used for simulation and graphs is included in *Appendix 4*.

6. Comparison of Plan II and Plan III

Both simulation and theoretical results are used to compare the proposed methods. If not otherwise stated, we use for Plan II the special case where the number of passed items equals the number of rejected items , i.e. $n_P = n_{\bar{P}} = N/2$.

First, we compare the distributions of each method's estimators as given by the R simulation. *Figures 3, 4* and *5* give these distributions for some specific values of a , b and p_P .

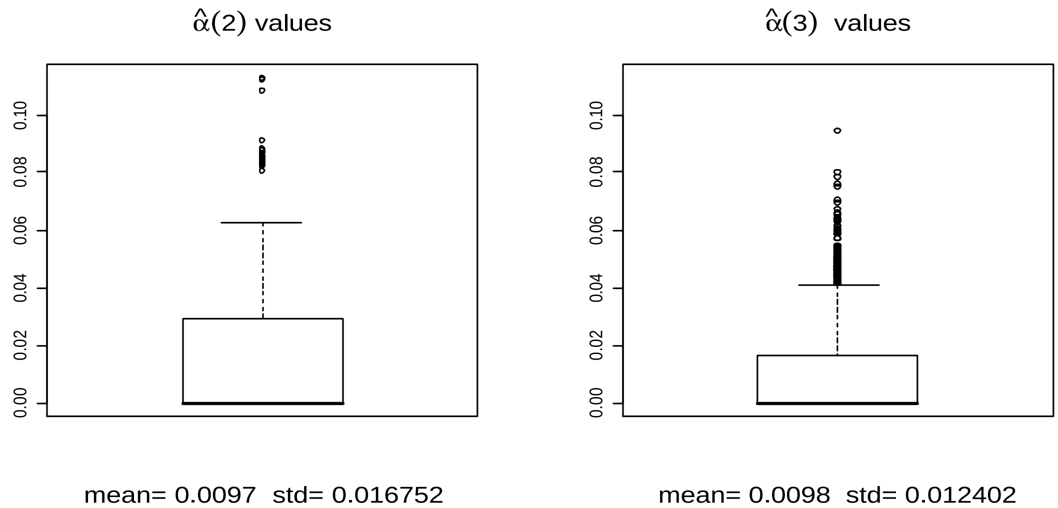

Figure 3: Simulation results for estimating a *using Plan II and Plan III,*
$a = 0.01$, $b = 0.02$, $p_P = 0.95$ *and the total sample size is 2,000 items*

The $d-method$ approximations for standard deviations are: $Std(\hat{a}(2)) = 0.01734$ and $Std(\hat{a}(3)) = 0.01261$. The theoretical and simulation results agree closely. Plan III gives the better estimator for a .

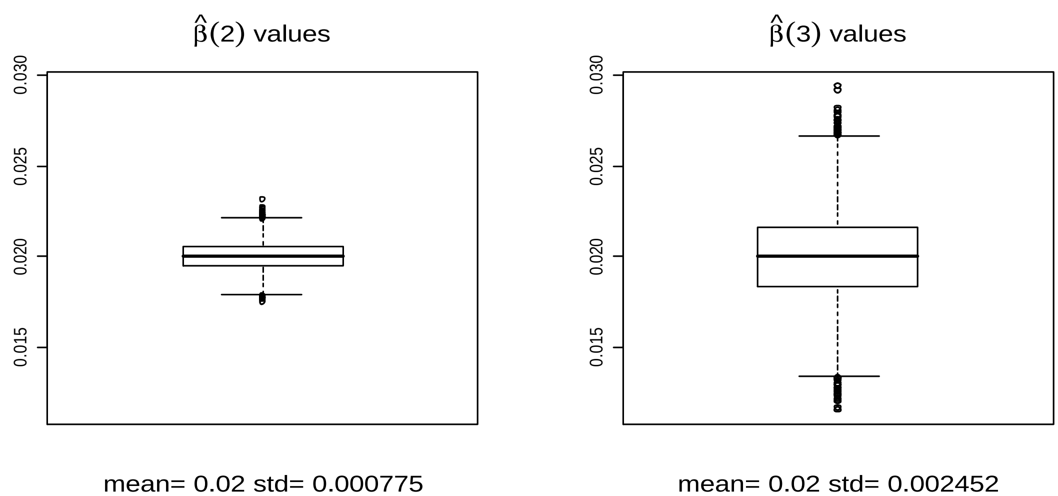

Figure 4: Simulation results for estimating b *using Plan II and Plan III,*
$a = 0.01$, $b = 0.02$, $p_P = 0.95$ *and the total sample size is 2,000 items*

We notice in *Figure 4.* that Plan II gives a more efficient estimator for b . The theoretical approximations for standard deviations are: $Std(\hat{b}(2)) = 0.000779$ and $Std(\hat{b}(3)) = 0.00246$.

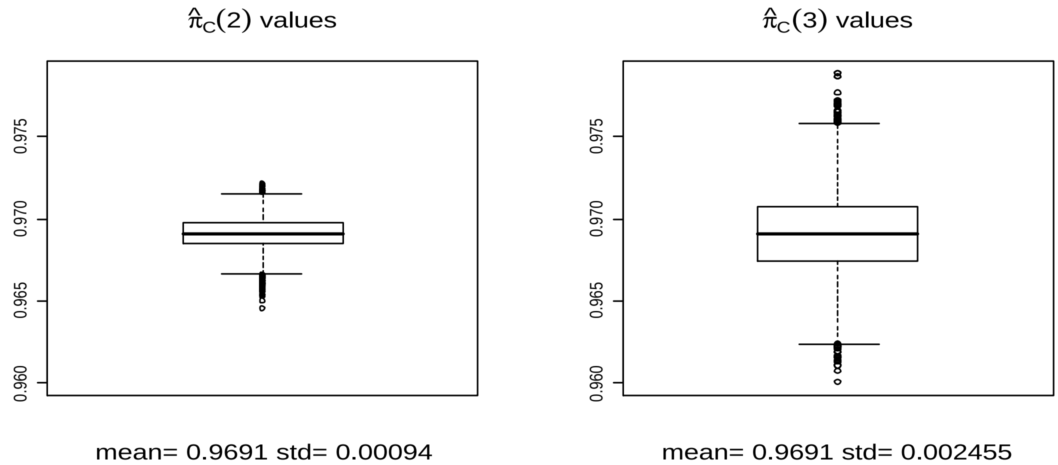

Figure 5: Simulation results for estimating π_C using Plan II and Plan III,
$\alpha = 0.01$, $\beta = 0.02$, $\pi_P = 0.95$ and the total sample size is 2,000 items

Plan II is also better than Plan III at estimating π_C. The theoretical approximations for estimators standard deviations are $Std(\hat{\pi}_C(2)) =$ 0.000941 and $Std(\hat{\pi}_C(3)) = 0.00246$, which are very close to the simulated values. We also notice that π_C estimators are unbiased.

We are also interested in analyzing how the estimators' standard deviations compare over a large range of values for α and β, for different values of π_P. Figures 6. – 8. give a graphical representation of the ratio of the standard deviations, for each parameter.

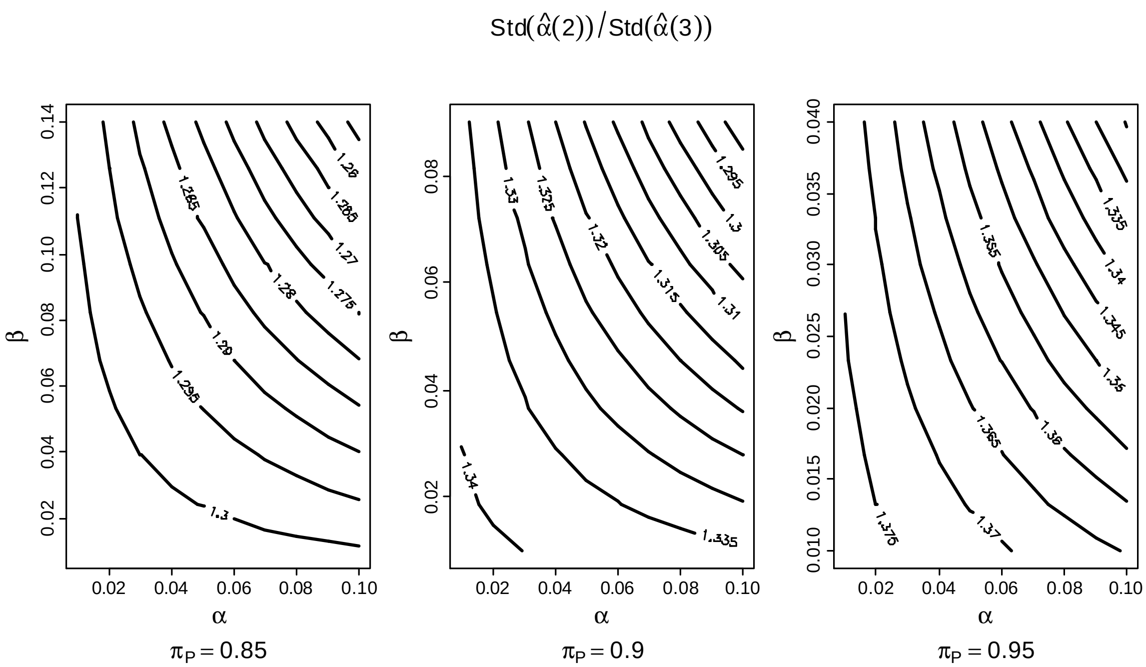

Figure 6: Contour plots for $Std(\hat{\alpha}(2))/Std(\hat{\alpha}(3))$

In the graph above we notice that Plan III gives a consistently better estimator for a over all values of a and b, and different values of p_P. Also, as p_P gets larger, the ratio gets larger for the same values of a, and b has a smaller influence on the ratio.

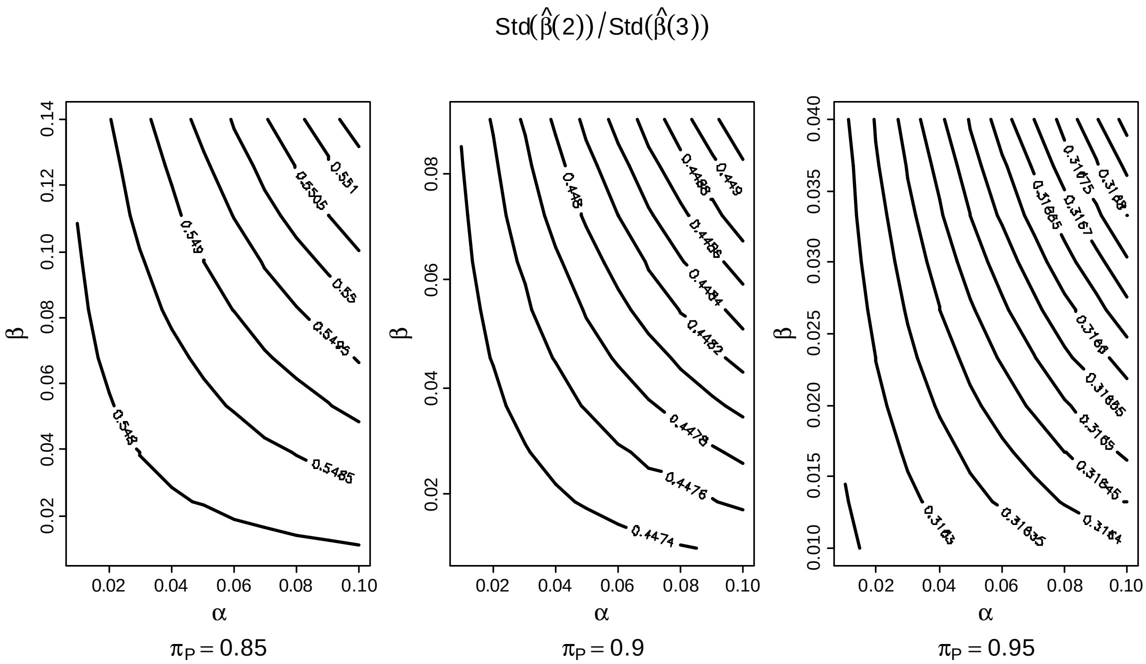

Figure 7: Contour plots for $Std(\hat{b}(2))/Std(\hat{b}(3))$

Plan II gives a more precise estimator of b than Plan III, as the ratio of the standard deviations is uniformly smaller than 1 over the whole range of a, b and p_P. Also the ratio varies only by a little over a and b ranges. This ratio becomes less sensitive to b values for larger values of p_P (e.g. 0.95).

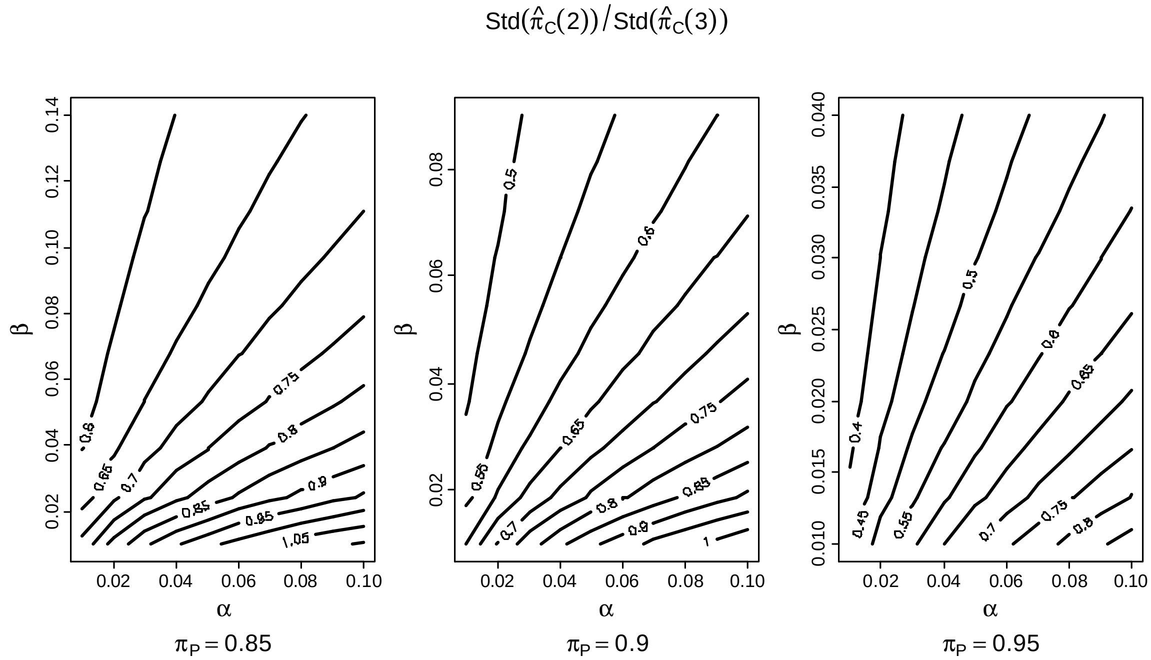

Figure 8: Contour plots for $Std(\hat{p}_C(2))/Std(\hat{p}_C(3))$

In *Figure 8* we notice that Plan II gives a more precise estimator for p_P, for small values of p_P and a. When $p_P = 0.95$, b has little influence on the ratio, especially for a less than 0.05.

It is also of an interest to compare the precision of the estimators when we use different sample sizes for passed and failed items in Plan II, i.e. $n_P \neq n_{\overline{P}}$. The graph below illustrates how the ratio of the standard deviations for the two estimators varies with the proportion of the passed items, for each parameter.

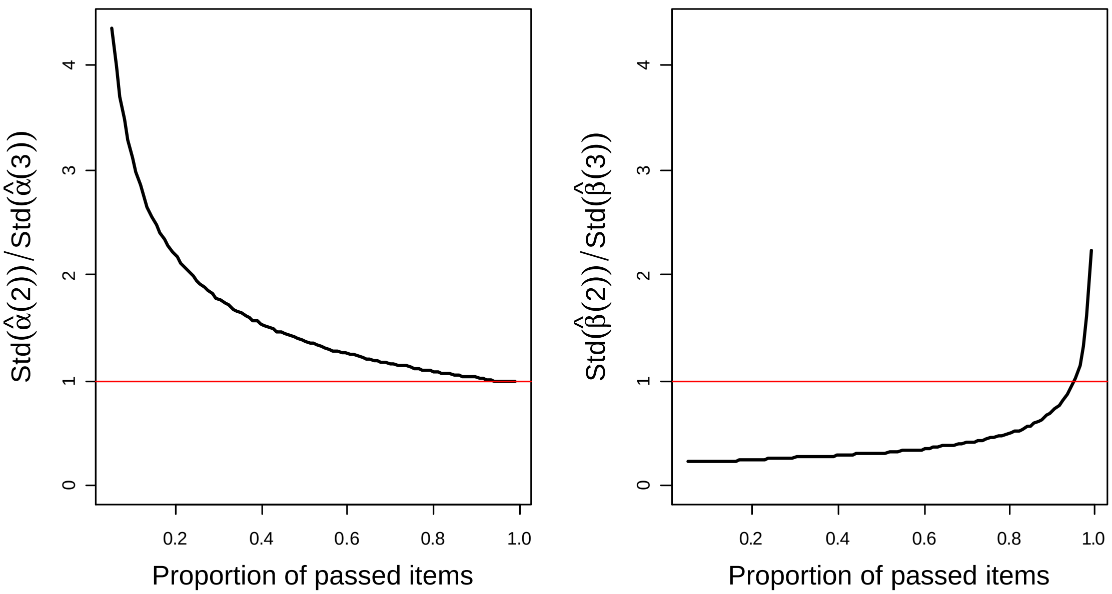

Figure 9: Standard deviations' ratios for a *and* b *estimators, when* $a = 0.01$, $b = 0.02$ and $p_P = 0.95$

Plan III gives a better estimator for a, regardless of the proportion of the passed items sample size. The ratio of the standard deviations is greater than 1 for the whole proportion interval $[0.05;1)$. In b case, Plan II gives the more precise estimator.

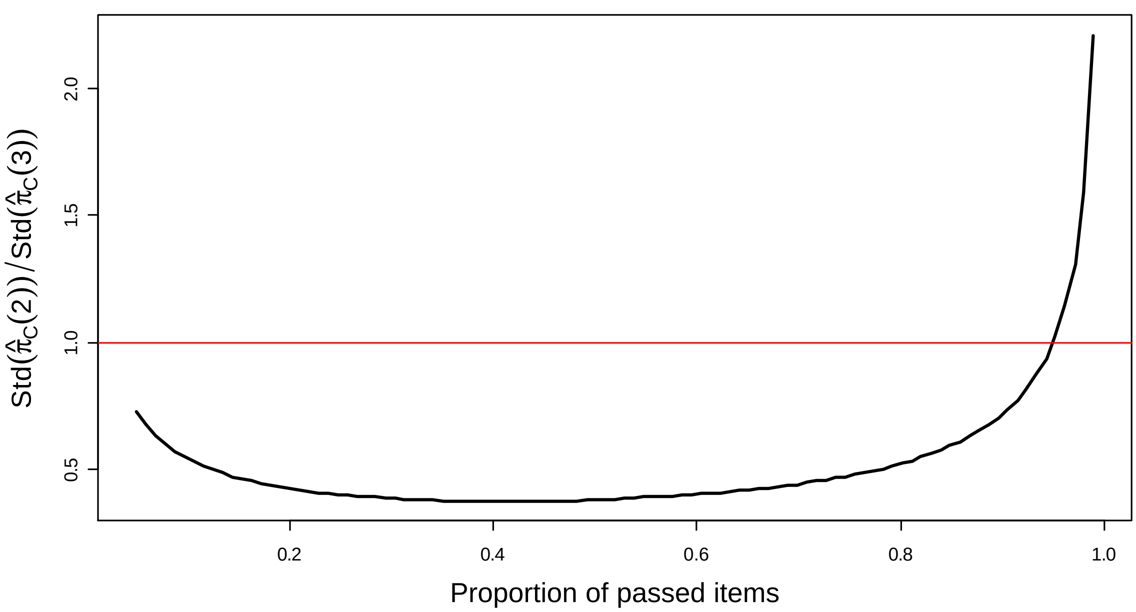

Figure 10: Standard deviations' ratios for p_C *estimators, when* $a = 0.01$, $b = 0.02$ and $p_P = 0.95$

In p_C case, we notice that the ratio is smaller that 1 for almost all the values of the proportion, but becomes larger than 1 for values close to 0.9.

7. *Sample Size Calculation*

So far we have assumed a certain value for the total sample size or have analyzed the ratio of the standard deviations, which does not depend on N. Now, suppose that we are interested in getting a certain precision for a parameter's estimator. Then, the objective is to find the minimum sample size for which we can achieve this precision. The functions that give the sample size for Plan II when we want a certain precision and assume certain values for a, b and know p_P are given below:

$$N_0(a(2)) = \frac{(p_P - a)(1-a)(1-f-b+bf-a+af+ab)}{[std_0(\hat{a}(2))]^2 f(1-f)(1-b-p_P)}$$

$$N_0(b(2)) = \frac{(ab+f-bf-af)b(1-b)(1-b-p_P)}{[std_0(\hat{b}(2))]^2 f(1-f)(p_P-a)}$$

$$N_0(p_C(2)) = \frac{(a+bf-ab-af)(1-b-p_P)}{[std_0(\hat{p}_C(2))]^2 f(1-f)(1-b-a)^2},$$

where f is the proportion of the passed items in the total sample size. The R-code for these functions is included in *Appendix 3*.

If we are interested in using Plan III, the functions that give the required sample sizes for certain values of the standard deviations are included in *Appendix 2.* We can either use these functions or we can use an indirect method that involves less computation. With the indirect method we look at the contour plots described above and find the standard deviation ratio for the assumed a and b values and the known p_P. Once we have the ratio, we can derive the corresponding standard deviation given by Plan II and finally get the total sample size, N_0.

For example, suppose that our goal is to estimate a with a precision of 0.0126 ($Std(\hat{a}) = 0.0126$). The assumed values for a and b are 0.01 and 0.02, respectively, and $p_P = 0.95$. Plan III gives the best estimator for a and now we are interested in finding the minimum total sample size, N_0, for this method. We look at *Figure 6* and find that, for the given values of a, b and p_P, the standard deviations ratio is 1.375. From this ratio we can compute $Std(\hat{a}(2)) = 0.01734$ and the minimum sample size to achieve that, using the above functions. We finally get $N_0(0.01734)\text{=}2000$, where the passed and failed items sample sizes are equal ($f = 0.5$).

8. *Conclusions*

The goal of this paper was to propose new methods for estimating the misclassification probabilities that characterize the quality of the measurement system and the probability of having a conforming item, which is a measure of the process quality.

The unique feature is that we assumed that the passing probability, p_P, is known, and this information is used in all methods.

We started by reviewing an already proposed method, i.e. Farnum's Plan, and concluded that this method is very difficult to implement. Then, we proposed two new methods that are more practical and less costly.

Neither of the two methods was found to be consistently better than the other for estimating all the parameters. The best method depends on the objective of the study.

Therefore, if we are interested in estimating a, the probability of passing a nonconforming item we recommend Plan III (Multinomial). This method gives a more precise estimator than the other one for a large range of a, b and p_P values. If we want to achieve a certain precision for this estimator, we can estimate the total required sample size.

If the study objective is to assess the second misclassification probability, the probability of rejecting a conforming item, or to assess the process quality, i.e. estimating p_C, it is better to use Plan II (Binomial). We have mentioned before that the standard deviation of the b estimator given by this method varies very slowly as we change the proportion of the passed items sample size. Therefore, we can use a very convenient sample structure by selecting unequal samples with a larger sample size for the passed items. This fact makes Plan II very convenient, as it is usually difficult to get a large number of rejected items. The standard deviation of p_C was almost constant when the proportion of the passed items sample size varied from 0.25 to 0.75. Therefore, we can select as many as $0.75N$ passed items and $0.25N$ rejected items and still have a precise p_C estimator.

References

Automotive Industry Action Group (AIAG) (1995), *Measurement Systems Analysis*, 2nd edition. Southfield MI.

Casella G. and Berger R. L. (2002), *Statistical Inference,* 2nd Edition, Brooks/Cole Publishing Company, Pacific Grove, California.

Farnum, Nicholas R. (1994), *Modern Statistical Quality Control and Improvement,* Duxbury Press, Belmont California.

Appendix I

Variance and bias approximations for Plan II (Binomial)

The Delta Method – variance and bias approximations for a function of random variables

Consider $X = (X_1, X_2, ..., X_n)$ a vector of n random variables with means $q = (q_1, q_2, ..., q_n)$ and a function $g(X)$ for which we want to estimate the bias and the variance. If $x = (x_1, x_2, ..., x_n)$ is a vector of realizations of the n random variables the Taylor's series expansion of $g(x)$ around $q = (q_1, q_2, ..., q_n)$ is:

$$g(x) = g(q) + \sum_{i=1}^{n}(x_i - q_i)\frac{\partial}{\partial x_i}g(x)\Big|_{x=q} + \frac{1}{2!}\Big[\sum_{i=1}^{n}(x_i - q_i)^2\frac{\partial^2}{\partial x_i^2}g(x)\Big|_{x=q}$$

$$+2\sum_{i}\sum_{j\neq i}(x_i - q_i)(x_j - q_j)\frac{\partial^2}{\partial x_i \partial x_j}g(x)\Big|_{x=q}\Big] + remainder$$

After taking the expectations we get:

$$E(g(X)) \simeq g(q)$$

$$Var(g(X)) \simeq \sum_{1}^{n}\Big[\frac{\partial}{\partial x_i}g(x)\Big|_{x=q}\Big]^2 Var(X_i) + \sum_{i}\sum_{j\neq i}\frac{\partial}{\partial x_i}g(x)\Big|_{x=q}\frac{\partial}{\partial x_j}g(x)\Big|_{x=q}Cov(X_i, X_j)$$

$$Bias(g(X)) = E(g(X)) - g(q) \simeq \frac{1}{2}\sum_{1}^{n}\frac{\partial^2}{\partial x_i^2}g(x)\Big|_{x=q}Var(X_i) + \sum_{i}\sum_{j\neq i}\frac{\partial^2}{\partial x_i \partial x_j}g(x)\Big|_{x=q}Cov(X_i, X_j)$$

The above method is often referred to as the $d - method$ (see G. Casella and R. L. Berger, "Statistical Inference", Second Edition, p. 241).

For Plan II, we select two independent samples of passed and failed items. In order to make the derivation easier to follow we use some new notation:

$g = \Pr(\bar{C}/P)$, $\hat{g} = \dfrac{n_{P\bar{C}}}{n_P}$ (MLE), $E(\hat{g}) = g$ and $Var(\hat{g}) = g(1-g)/n_P$

$d = \Pr(\bar{C}/\bar{P})$, $\hat{d} = \dfrac{n_{\bar{P}\bar{C}}}{n_{\bar{P}}}$ (MLE), $E(\hat{d}) = d$, $Var(\hat{d}) = d(1-d)/n_{\bar{P}}$ and $Cov(\hat{g}, \hat{d}) = 0$

We know that: $\hat{a}(2) = \dfrac{\hat{g}p_P}{\hat{g}p_P + \hat{d}(1-p_P)}$, $\hat{b}(2) = \dfrac{(1-\hat{d})(1-p_P)}{(1-\hat{d})(1-p_P) + (1-\hat{g})p_P}$

and $\hat{p}_C = (1-\hat{d})(1-p_P) + (1-\hat{g})p_P$

Variances approximations

Using the above formulae for variance approximation we get:

$$Var(\hat{a}(2)) \simeq \frac{p_P^2(1-p_P)^2 dg}{(p_P g + (1-p_P)d)^4}\Big(\frac{d(1-g)}{n_P} + \frac{g(1-d)}{n_{\bar{P}}}\Big)$$

$$Var(\hat{b}(2)) \simeq \frac{p_P^{\,2}(1-p_P)^2(1-d)(1-g)}{((1-d)(1-p_P)+(1-g)p_P)^4}\left(\frac{(1-d)g}{n_P}+\frac{(1-g)d}{n_{\overline{P}}}\right)$$

The variance of $\hat{p}_C$ can be derived directly:

$$Var(\hat{p}_C(2)) = Var((1-d)(1-p_P)+(1-g)p_P) = (1-p_P)^2 Var(d) + p_P^{\,2} Var(g)$$
$$= \frac{(1-p_P)^2 d(1-d)}{n_{\overline{P}}} + \frac{p_P^{\,2} g(1-g)}{n_P}$$

Using the identities:

$g = \dfrac{a(1-b-p_P)}{p_P(1-a-b)}$ and $d = \dfrac{(1-a)(1-b-p_P)}{(1-p_P)(1-a-b)}$ we re-express the above approximations in terms of a, b and p_P as follows:

$$Var(\hat{a}(2)) \simeq \frac{a(1-a)(p_P-a)}{1-b-p_P}\left(\frac{1-a-b+ab}{n_P}+\frac{ab}{n_{\overline{P}}}\right)$$
$$Var(\hat{b}(2)) \simeq \frac{b(1-b)(1-b-p_P)}{p_P-a}\left(\frac{ab}{n_P}+\frac{1-b-a+ab}{n_{\overline{P}}}\right)$$

and

$$Var(\hat{p}(2)) = \frac{(1-b-p_P)(p_P-a)}{(1-a-b)^2}\left(\frac{a(1-b)}{n_P}+\frac{b(1-a)}{n_{\overline{P}}}\right)$$

Biases approximations

Using the above identities and the formulae for bias approximation and we get:

$$Bias(\hat{a}(2)) \simeq \frac{a(1-a)(p_P-a)(n_P b + n_{\overline{P}} b - n_{\overline{P}})}{n_P n_{\overline{P}}(1-b-p_P)}$$
$$Bias(\hat{b}(2)) \simeq \frac{b(1-b)(1-b-p_P)(n_P a + n_{\overline{P}} a - n_P)}{n_P n_{\overline{P}}(p_P-a)}$$

$Bias(\hat{p}_C(2)) = 0$, since

$$E(\hat{p}_C(2)) = E[(1-\hat{d})(1-p_P)+(1-\hat{g})p_P] = (1-d)(1-p_P)+(1-g)p_P = p_C$$

Appendix 2

Variance and bias approximations for Plan III (Multinomial)

For Plan III we select N items from the total population of items previously measured by the BMS and then the selected items are classified as conforming or nonconforming by the "gold standard" system.

We denote $\Pr(P \cap C) = q_1$, $\Pr(\overline{P} \cap C) = q_2$, $\Pr(P \cap \overline{C}) = q_3$, $\Pr(\overline{P} \cap \overline{C}) = q_4$ and we have the following identities:

$$q_1 = \frac{(1-b)(p_P - a)}{1-a-b}, \quad q_2 = \frac{b(p_P - a)}{1-a-b}, \quad q_3 = \frac{a(1-b-p_P)}{1-a-b} \text{ and } q_4 = \frac{(1-a)(1-b-p_P)}{1-a-b}$$

Note that the initial information about p_P can be translated into a restriction for q_1 and q_3, i.e.$q_1 + q_3 = p_P$. First, we express a, b and p_C in terms of q_1, q_2, q_3 and q_4:

$$a = \frac{\Pr(P \cap \overline{C})}{\Pr(\overline{C})} = \frac{q_3}{q_3 + q_4}$$

$$b = \frac{\Pr(\overline{P} \cap C)}{\Pr(C)} = \frac{q_2}{q_2 + q_1}$$

$$p_C = q_1 + q_2$$

Then, we estimate q_1, q_2, q_3 and q_4 using the Maximum Likelihood Method and Lagrange Multipliers to include the additional restriction $q_1 + q_3 = p_P$.

We get the following estimates:

$$\hat{q}_1 = \frac{p_P n_{PC}}{n_{PC} + n_{P\overline{C}}}, \quad \hat{q}_2 = \frac{(1-p_P) n_{\overline{P}C}}{n_{\overline{P}C} + n_{\overline{P}\overline{C}}}, \quad \hat{q}_3 = \frac{p_P n_{P\overline{C}}}{n_{PC} + n_{P\overline{C}}} \text{ and } \hat{q}_4 = \frac{(1-p_P) n_{\overline{P}\overline{C}}}{n_{\overline{P}C} + n_{\overline{P}\overline{C}}}$$

Using the invariance property of the ML estimators we get the estimates for a, b and p_C:

$$\hat{a}(3) = \frac{p_P n_{P\overline{C}} (n_{\overline{P}C} + n_{\overline{P}\overline{C}})}{p_P n_{P\overline{C}} (n_{\overline{P}C} + n_{\overline{P}\overline{C}}) + (1-p_P) n_{\overline{P}\overline{C}} (n_{PC} + n_{P\overline{C}})}$$

$$\hat{b}(3) = \frac{(1-p_P) n_{\overline{P}C} (n_{PC} + n_{P\overline{C}})}{(1-p_P) n_{\overline{P}C} (n_{PC} + n_{P\overline{C}}) + p_P n_{PC} (n_{\overline{P}C} + n_{\overline{P}\overline{C}})}$$

$$\hat{p}_C(3) = \frac{p_P n_{PC}}{n_{PC} + n_{P\overline{C}}} + \frac{(1-p_P) n_{\overline{P}C}}{n_{\overline{P}C} + n_{\overline{P}\overline{C}}}$$

Variances approximations

We use again the $d-method$ to approximate estimators' variances. We also use the fact that the marginal distributions for a multinomial distribution are binomial. Thus, $N_{PC} \sim BIN(N, q_1)$, $N_{\overline{P}C} \sim BIN(N, q_2)$, $N_{P\overline{C}} \sim BIN(N, q_3)$ and $N_{\overline{P}\overline{C}} \sim BIN(N, q_4)$. We also know their covariances: $Cov(N_{PC}, N_{\overline{P}C}) = -Nq_1q_2$, $Cov(N_{PC}, N_{P\overline{C}}) = -Nq_1q_3$ etc. We have:

$$Var(\hat{a}(3)) \simeq \frac{p_P^2 (1-p_P)^2 q_4 q_3 (q_2 + q_4)(q_1 + q_3)(q_3^2 q_2 + q_1 q_3 q_2 + q_4^2 q_1 + q_2 q_4 q_1)}{N (p_P q_3 q_2 + q_4 q_1 + q_4 q_3 - p_P q_4 q_1)^4}$$

$$Var(\hat{b}(3)) \simeq \frac{p_P^2 (1-p_P)^2 q_2 q_1 (q_2 + q_4)(q_1 + q_3)(q_3 q_4 q_2 + q_4 q_3 q_1 + q_2^2 q_3 + q_1^2 q_4)}{N (p_P q_3 q_2 - q_2 q_1 - q_2 q_3 - p_P q_4 q_1)^4}$$

$$
\begin{aligned}
Var(\hat{p}_C(3)) \simeq{} & (q_3 p_P^2 q_1 q_4^3 + 3q_3 p_P^2 q_1 q_2 q_4^2 + q_2 q_4 q_1^3 + q_3^3 q_4 q_2 + 3q_4 q_3 q_2 p_P^2 q_1^2 \\
& + 3q_3 q_4 q_2 q_1^2 + 3q_4 q_3^2 q_2 p_P^2 q_1 + q_4 q_2 p_P^2 q_1^3 - 2q_4 q_2 p_P q_1^3 \\
& - 6q_4 q_3 q_2 p_P q_1^2 - 2q_4 q_3^3 q_2 p_P + q_4 q_3^3 q_2 p_P^2 + 3q_4 q_3 p_P^2 q_1 q_2^2 \\
& - 6q_4 q_3^2 q_2 p_P q_1 + 3q_3^2 q_4 q_2 q_1 + q_3 p_P^2 q_1 q_2^3) / (N(q_2 + q_4)^3 (q_1 + q_3)^3)
\end{aligned}
$$

Biases approximations

$$
\begin{aligned}
Bias(\hat{a}(3)) \simeq{} & p_P(1-p_P) q_4 q_3 (2p_P q_3 q_2 q_1 + q_2 p_P q_1^2 + q_1 p_P q_2^2 + p_P q_3^2 q_2 + 2q_4 q_1 p_P q_2 \\
& + q_4^2 q_1 p_P - 2q_2 q_1 q_3 - q_2 q_1^2 - q_2 q_3^2) / N(q_4 p_P q_1 - p_P q_3 q_2 - q_4 q_1 - q_4 q_3)^3 \\
Bias(\hat{b}(3)) \simeq{} & p_P(1-p_P) q_2 q_1 (q_3 q_4^2 p_P - 2q_4 q_3 q_1 + q_4 p_P q_3^2 + q_2^2 q_3 p_P + q_4 p_P q_1^2 \\
& - q_4 q_3^2 - q_4 q_1^2 + 2q_4 p_P q_3 q_1 + 2q_3 q_2 q_4 p_P) / N(q_4 p_P q_1 + q_2 q_1 + q_3 q_2 - p_P q_3 q_2)^3
\end{aligned}
$$

We can prove that $\hat{p}_C(3)$ is an unbiased estimator of p_C using the conditional distributions:

$N_{PC} \mid (N_{PC} + N_{P\bar{C}}) \sim BIN(N_{PC} + N_{P\bar{C}}, \frac{q_1}{p_P})$ and $N_{\bar{P}C} \mid (N_{\bar{P}C} + N_{\bar{P}\bar{C}}) \sim BIN(N_{\bar{P}C} + N_{\bar{P}\bar{C}}, \frac{q_2}{1-p_P})$

and the conditional mean identity $E(E(X \mid Y)) = E(X)$.

We have:

$$
E(\hat{p}(3)) = E[p_P \frac{N_{PC}}{N_{PC} + N_{P\bar{C}}} + (1-p_P)\frac{N_{\bar{P}C}}{N_{\bar{P}C} + N_{\bar{P}\bar{C}}}] = p_P E(\frac{N_{PC}}{N_{PC} + N_{P\bar{C}}}) + (1-p_P) E(\frac{N_{\bar{P}C}}{N_{\bar{P}C} + N_{\bar{P}\bar{C}}})
$$

From the above conditional distributions we get:

$E(\frac{N_{PC}}{N_{PC} + N_{P\bar{C}}} \mid N_{PC} + N_{P\bar{C}}) = \frac{q_1}{p_P}$ and $E(\frac{N_{\bar{P}C}}{N_{\bar{P}C} + N_{\bar{P}\bar{C}}} \mid N_{\bar{P}C} + N_{\bar{P}\bar{C}}) = \frac{q_2}{1-p_P}$

$$
E[E(\frac{N_{PC}}{N_{PC} + N_{P\bar{C}}} \mid N_{PC} + N_{P\bar{C}})] = E(\frac{N_{PC}}{N_{PC} + N_{P\bar{C}}}) = \frac{q_1}{p_P}
$$

and

$$
E[E(\frac{N_{\bar{P}C}}{N_{\bar{P}C} + N_{\bar{P}\bar{C}}} \mid N_{\bar{P}C} + N_{\bar{P}\bar{C}})] = E(\frac{N_{\bar{P}C}}{N_{\bar{P}C} + N_{\bar{P}\bar{C}}}) = \frac{q_2}{1-p_P}
$$

Therefore,

$$
E[\hat{p}_C(3)] = p_P E(\frac{N_{PC}}{N_{PC} + N_{P\bar{C}}}) + (1-p_P) E(\frac{N_{\bar{P}C}}{N_{\bar{P}C} + N_{\bar{P}\bar{C}}}) = p_P \frac{q_1}{p_P} + (1-p_P)\frac{q_2}{(1-p_P)} = q_1 + q_2 = p_C
$$

Appendix 3
Maple-code for variance and bias approximations

Variances approximation

Method II (n1 passed items sample size and n2 failed items sample size)

```
alpha2:=gamma*q/(gamma*q+delta*(1-q));
beta2:=(1-delta)*(1-q)/((1-delta)*(1-q)+(1-gamma)*q);
phat2:=(1-gamma)*q+(1-delta)*(1-q);

vargamma:=gamma*(1-gamma)/n1;
vardelta:=delta*(1-delta)/n2;

diffa2gam:=simplify(diff(alpha2,gamma));
diffa2delt:=simplify(diff(alpha2,delta));
varalpha2:=diffa2gam^2*vargamma+diffa2delt^2*vardelta;
valpha2:=simplify(subs(delta=(1-alpha)*(1-beta-q)/((1-q)*(1-beta-
alpha)),subs(gamma=alpha*(1-beta-q)/(q*(1-beta-alpha)),varalpha2)));

diffb2gam:=simplify(diff(beta2,gamma));
diffb2delt:=simplify(diff(beta2,delta));
varbeta2:=diffb2gam^2*vargamma+diffb2delt^2*vardelta;
vbeta2:=simplify(subs(delta=(1-alpha)*(1-beta-q)/((1-q)*(1-beta-
alpha)),subs(gamma=alpha*(1-beta-q)/(q*(1-beta-alpha)),varbeta2)));

diffp2gam:=simplify(diff(phat2,gamma));
diffp2delt:=simplify(diff(phat2,delta));
varp2:=diffp2gam^2*vargamma+diffp2delt^2*vardelta;
vp2:=simplify(subs(delta=(1-alpha)*(1-beta-q)/((1-q)*(1-beta-alpha)),subs(gamma=alpha*(1-
beta-q)/(q*(1-beta-alpha)),varp2)));
```

Method II (f*N passed items sample size and (1-f)*N failed items sample size)

```
alpha2:=gamma*q/(gamma*q+delta*(1-q));
beta2:=(1-delta)*(1-q)/((1-delta)*(1-q)+(1-gamma)*q);
phat2:=(1-gamma)*q+(1-delta)*(1-q);

vargamma:=gamma*(1-gamma)/(f*N);
vardelta:=delta*(1-delta)/((1-f)*N);

diffa2gam:=simplify(diff(alpha2,gamma));
diffa2delt:=simplify(diff(alpha2,delta));
varalpha2:=diffa2gam^2*vargamma+diffa2delt^2*vardelta;
valpha2:=simplify(subs(delta=(1-alpha)*(1-beta-q)/((1-q)*(1-beta-
alpha)),subs(gamma=alpha*(1-beta-q)/(q*(1-beta-alpha)),varalpha2)));
ratioalphaf:=simplify(subs(delta=(1-alpha)*(1-beta-q)/((1-q)*(1-beta-
alpha)),subs(gamma=alpha*(1-beta-q)/(q*(1-beta-alpha)),varalpha2/valpha4)));

diffb2gam:=simplify(diff(beta2,gamma));
diffb2delt:=simplify(diff(beta2,delta));
varbeta2:=diffb2gam^2*vargamma+diffb2delt^2*vardelta;
vbeta2:=simplify(subs(delta=(1-alpha)*(1-beta-q)/((1-q)*(1-beta-
alpha)),subs(gamma=alpha*(1-beta-q)/(q*(1-beta-alpha)),varbeta2)));
```

```
ratiobetaf:=simplify(subs(delta=(1-alpha)*(1-beta-q)/((1-q)*(1-beta-alpha)),subs(gamma=alpha*(1-beta-q)/(q*(1-beta-alpha)),varbeta2/vbeta4)));

diffp2gam:=simplify(diff(phat2,gamma));
diffp2delt:=simplify(diff(phat2,delta));
varp2:=diffp2gam^2*vargamma+diffp2delt^2*vardelta;
vp2:=simplify(subs(delta=(1-alpha)*(1-beta-q)/((1-q)*(1-beta-alpha)),subs(gamma=alpha*(1-beta-q)/(q*(1-beta-alpha)),varp2)));
ratiopf:=simplify(subs(delta=(1-alpha)*(1-beta-q)/((1-q)*(1-beta-alpha)),subs(gamma=alpha*(1-beta-q)/(q*(1-beta-alpha)),varp2/vp4hat)));
```

Method IV

```
theta1:=q*x1/(x1+x3);
theta2:=(1-q)*x2/(x2+x4);
theta3:=q*x3/(x1+x3);
theta4:=(1-q)*x4/(x2+x4);

alpha4:=theta3/(theta3+theta4);
beta4:=theta2/(theta1+theta2);
p4hat:=theta1+theta2;

dalpha4x1:=simplify(diff(alpha4,x1));
dalpha4x2:=simplify(diff(alpha4,x2));
dalpha4x3:=simplify(diff(alpha4,x3));
dalpha4x4:=simplify(diff(alpha4,x4));

dbeta4x1:=simplify(diff(beta4,x1));
dbeta4x2:=simplify(diff(beta4,x2));
dbeta4x3:=simplify(diff(beta4,x3));
dbeta4x4:=simplify(diff(beta4,x4));

dp4x1:=diff(p4hat,x1);
dp4x2:=diff(p4hat,x2);
dp4x3:=diff(p4hat,x3);
dp4x4:=diff(p4hat,x4);

varx1:=p1*(1-p1)*N;
varx2:=p2*(1-p2)*N;
varx3:=p3*(1-p3)*N;
varx4:=p4*(1-p4)*N;
covx1x2:=-N*p1*p2;
covx1x3:=-N*p1*p3;
covx1x4:=-N*p1*p4;
covx2x3:=-N*p2*p3;
covx2x4:=-N*p2*p4;
covx3x4:=-N*p3*p4;
```

```
varalpha4_1:=(dalpha4x1^2)*varx1+(dalpha4x2^2)*varx2+(dalpha4x
3^2)*varx3+(dalpha4x4^2)*varx4;

varalpha4_2:=2*(dalpha4x1*dalpha4x2*covx1x2+dalpha4x1*dalpha4x
3*covx1x3+dalpha4x1*dalpha4x4*covx1x4+dalpha4x2*dalpha4x3*covx
2x3+dalpha4x2*dalpha4x4*covx2x4+dalpha4x3*dalpha4x4*covx3x4);

varalpha4:=varalpha4_1+varalpha4_2;
valpha4:=subs(x4=N*p4,subs(x3=N*p3,subs(x2=N*p2,subs(x1=N*p1,varalpha4))));

varbeta4_1:=(dbeta4x1^2)*varx1+(dbeta4x2^2)*varx2+(dbeta4x3^2)
*varx3+(dbeta4x4^2)*varx4;

varbeta4_2:=2*(dbeta4x1*dbeta4x2*covx1x2+dbeta4x1*dbeta4x3*cov
x1x3+dbeta4x1*dbeta4x4*covx1x4+dbeta4x2*dbeta4x3*covx2x3+dbeta
4x2*dbeta4x4*covx2x4+dbeta4x3*dbeta4x4*covx3x4);

varbeta4:=varbeta4_1+varbeta4_2;
vbeta4:=subs(x4=N*p4,subs(x3=N*p3,subs(x2=N*p2,subs(x1=N*p1,va
rbeta4))));

varp4hat_1:=(dp4x1^2)*varx1+(dp4x2^2)*varx2+(dp4x3^2)*varx3+(d
p4x4^2)*varx4;

varp4hat_2:=2*(dp4x1*dp4x2*covx1x2+dp4x1*dp4x3*covx1x3+dp4x1*d
p4x4*covx1x4+dp4x2*dp4x3*covx2x3+dp4x2*dp4x4*covx2x4+dp4x3*dp4
x4*covx3x4);

varp4hat:=varp4hat_1+varp4hat_2;
vp4hat:=subs(x4=N*p4,subs(x3=N*p3,subs(x2=N*p2,subs(x1=N*p1,va
rp4hat))));
```

Biases Approximation

Method II (n1 passed items sample size and n2 failed items sample size)

```
alpha2:=gamma*q/(gamma*q+delta*(1-q));
beta2:=(1-delta)*(1-q)/((1-delta)*(1-q)+(1-gamma)*q);
phat2:=(1-gamma)*q+(1-delta)*(1-q);

vargamma:=gamma*(1-gamma)/n1;
vardelta:=delta*(1-delta)/n2;

diffa2gam2:=simplify(diff(diff(alpha2,gamma),gamma));
diffa2delt2:=simplify(diff(diff(alpha2,delta),delta));
diffa2gamdelt:=simplify(diff(diff(alpha2,gamma),delta));

biasalpha2:=simplify((1/2)*(diffa2gam2*vargamma+diffa2delt2*vardelta));
```

```
biasalpha2:=simplify(subs(delta=(1-alpha)*(1-beta-q)/((1-q)*(1-beta-
alpha)),subs(gamma=alpha*(1-beta-q)/(q*(1-beta-alpha)),biasalpha2)));

diffb2gam2:=simplify(diff(diff(beta2,gamma),gamma));
diffb2delt2:=simplify(diff(diff(beta2,delta),delta));

biasbeta2:=simplify((1/2)*(diffb2gam2*vargamma+diffb2delt2*vardelta));
biasbeta2:=simplify(subs(delta=(1-alpha)*(1-beta-q)/((1-q)*(1-beta-
alpha)),subs(gamma=alpha*(1-beta-q)/(q*(1-beta-alpha)),biasbeta2)));
```

Method IV

```
dalpha4x1x1:=simplify(diff(diff(alpha4,x1),x1));
dalpha4x2x2:=simplify(diff(diff(alpha4,x2),x2));
dalpha4x3x3:=simplify(diff(diff(alpha4,x3),x3));
dalpha4x4x4:=simplify(diff(diff(alpha4,x4),x4));
dalpha4x1x2:=simplify(diff(diff(alpha4,x1),x2));
dalpha4x1x3:=simplify(diff(diff(alpha4,x1),x3));
dalpha4x1x4:=simplify(diff(diff(alpha4,x1),x4));
dalpha4x2x3:=simplify(diff(diff(alpha4,x2),x3));
dalpha4x2x4:=simplify(diff(diff(alpha4,x2),x4));
dalpha4x3x4:=simplify(diff(diff(alpha4,x3),x4));
biasalpha4_1:=1/2*(dalpha4x1x1*varx1+dalpha4x2x2*varx2+dalpha4x3x3*varx3+dalpha4x
4x4*varx4);
biasalpha4_2:=dalpha4x1x2*covx1x2+dalpha4x1x3*covx1x3+dalpha4x1x4*covx1x4+dalph
a4x2x3*covx2x3+dalpha4x2x4*covx2x4+dalpha4x3x4*covx3x4;
biasalpha4:=biasalpha4_1+biasalpha4_2;
bialpha4:=subs(x4=N*p4,subs(x3=N*p3,subs(x2=N*p2,subs(x1=N*p1,biasalpha4))));
factor(simplify(bialpha4));

dbeta4x1x1:=simplify(diff(diff(beta4,x1),x1));
dbeta4x2x2<-simplify(diff(diff(beta4,x2),x2));
dbeta4x2x2:=simplify(diff(diff(beta4,x2),x2));
dbeta4x3x3:=simplify(diff(diff(beta4,x3),x3));
dbeta4x4x4:=simplify(diff(diff(beta4,x4),x4));
dbeta4x1x2:=simplify(diff(diff(beta4,x1),x2));
dbeta4x1x3:=simplify(diff(diff(beta4,x1),x3));
dbeta4x1x4:=simplify(diff(diff(beta4,x1),x4));
dbeta4x2x3:=simplify(diff(diff(beta4,x2),x3));
dbeta4x2x4:=simplify(diff(diff(beta4,x2),x4));
dbeta4x3x4:=simplify(diff(diff(beta4,x3),x4));
biasbeta4_1:=1/2*(dbeta4x1x1*varx1+dbeta4x2x2*varx2+dbeta4x3x3*varx3+dbeta4x4x4*v
arx4);
biasbeta4_2:=dbeta4x1x2*covx1x2+dbeta4x1x3*covx1x3+dbeta4x1x4*covx1x4+dbeta4x2x
3*covx2x3+dbeta4x2x4*covx2x4+dbeta4x3x4*covx3x4;
biasbeta4:=biasbeta4_1+biasbeta4_2;
bibeta4:=subs(x4=N*p4,subs(x3=N*p3,subs(x2=N*p2,subs(x1=N*p1,biasbeta4))));
```

```
factor(simplify(bibeta4));

dfp4x1x1:=simplify(diff(diff(p4hat,x1),x1));
dfp4x2x2:=simplify(diff(diff(p4hat,x2),x2));
dfp4x3x3:=simplify(diff(diff(p4hat,x3),x3));
dfp4x4x4:=simplify(diff(diff(p4hat,x4),x4));
dfp4x1x2:=simplify(diff(diff(p4hat,x1),x2));
dfp4x1x3:=simplify(diff(diff(p4hat,x1),x3));
dfp4x1x4:=simplify(diff(diff(p4hat,x1),x4));
dfp4x2x3:=simplify(diff(diff(p4hat,x2),x3));
dfp4x2x4:=simplify(diff(diff(p4hat,x2),x4));
dfp4x3x4:=simplify(diff(diff(p4hat,x3),x4));
biasp4_1:=1/2*(dfp4x1x1*varx1+dfp4x2x2*varx2+dfp4x3x3*varx3+dfp4x4x4*varx4);
biasp4_2:=dfp4x1x2*covx1x2+dfp4x1x3*covx1x3+dfp4x1x4*covx1x4+dfp4x2x3*covx2x3
+dfp4x2x4*covx2x4+dfp4x3x4*covx3x4;
biasp4:=biasp4_1+biasp4_2;
bip4:=subs(x4=N*p4,subs(x3=N*p3,subs(x2=N*p2,subs(x1=N*p1,biasp4))));
simplify(bip4);
```

Appendix 4
R-code for simulations, functions and graphs

NOTE:

The following R-code uses a slightly different notation for parameters and their estimates. Thus, p_P is denoted by q, p_C by p, a by "alpha" and b by "beta". The estimates are denoted by the name of the estimated quantity (alpha, beta and p) and a number corresponding to the method. For Plan II the specific number that denotes the estimates is 2 and for Plan III the specific number is 4.

#Simulation results for estimating alpha, beta and p

```
#assign alpha, beta, q and N
#in most of the essay examples the assigned values were:
N<-2000
q=0.95
beta=0.02
alpha=0.01
p<-(q-alpha)/(1-beta-alpha)
#p should be smaller than 1
p<1
```

#Plan II (Binomial)

```
gamma<-(alpha*(1-p))/(alpha*(1-p)+(1-beta)*p)
delta<-((1-alpha)*(1-p))/((1-alpha)*(1-p)+beta*p)
gamma
```

```
delta
N1<-trunc(N/2)
N1
N<-2*N1
gammahat<-rbinom(10000,N1,gamma)/N1
deltahat<-rbinom(10000,N1,delta)/N1
gamma*q+delta*(1-q)
1-p
#they should be the same
(1-q)*(1-delta)+(1-gamma)*q
p
#they should be the same
alpha2<-q*gammahat/(q*gammahat+(1-q)*deltahat)
mean(alpha2)
sd(alpha2)
beta2<-((1-deltahat)*(1-q))/((1-deltahat)*(1-q)+(1-gammahat)*q)
mean(beta2)
sd(beta2)
phat2<-(1-gammahat)*q+(1-deltahat)*(1-q)
mean(phat2)
sd(phat2)
```

#Plan III (Multinomial)

```
p1<-p*(1-beta)
p1
p2<-beta*p
p2
p3<-(1-p)*alpha
p3
p4<-(1-p)*(1-alpha)
p4
p1+p2+p3+p4
method3<-rmultinom(10000,size=N,prob=c(p1,p2,p3,p4))
x1<-method3[1,]
x2<-method3[2,]
x3<-method3[3,]
x4<-method3[4,]
p1hat<-(q*x1)/(x1+x3)
p3hat<-(q*x3)/(x1+x3)
p2hat<-((1-q)*x2)/(x2+x4)
p4hat<-((1-q)*x4)/(x2+x4)
alpha4<-p3hat/(p3hat+p4hat)
mean(alpha4)
sd(alpha4)
beta4<-p2hat/(p1hat+p2hat)
mean(beta4)
```

```
sd(beta4)
phat4<-p1hat+p2hat
mean(phat4)
sd(phat4)
```

#Box Plots

#Alpha

```
par(mfrow=c(1,2))
par(mar=c(8,5,5,2)+0.1)
boxplot(alpha2,ylim=c(min(alpha2,alpha4),max(alpha2,alpha4)),main=expression(paste(hat(a
lpha)(2), " values")),
sub=paste("mean=",round(mean(alpha2),digit=4),"
std=",round(sd(alpha2),digit=6)),cex.lab=1,cex.sub=1.5,cex.main=1.5)
boxplot(alpha4,ylim=c(min(alpha2,alpha4),max(alpha2,alpha4)),main=expression(paste(hat(a
lpha)(3),"  values")), sub=paste("mean=",round(mean(alpha4),digit=4),"
std=",round(sd(alpha4),digit=6)),cex.lab=1,cex.sub=1.5,cex.main=1.5)
```

#Beta

```
par(mfrow=c(1,2))
par(mar=c(8,5,5,2)+0.1)
boxplot(beta2,ylim=c(min(beta2,beta4),max(beta2,beta4)),main=expression(paste(hat(beta)(2
), " values")),
xlab=paste("mean=",round(mean(beta2),digit=4),"std=",round(sd(beta2),digit=6)),cex.lab=1.5
,cex.main=1.5)
boxplot(beta4,ylim=c(min(beta2,beta4),max(beta2,beta4)),main=expression(paste(hat(beta)(3
), " values")),
xlab=paste("mean=",round(mean(beta4),digit=4),"std=",round(sd(beta4),digit=6)),cex.lab=1.5
,cex.main=1.5)
```

#p

```
par(mfrow=c(1,2))
par(mar=c(8,5,5,2)+0.1)
boxplot(phat2,ylim=c(min(phat2,phat4),max(phat2,phat4)),main=expression(paste(hat(pi)[C](
2), " values")),
xlab=paste("mean=",round(mean(phat2),digit=4),"std=",round(sd(phat2),digit=6)),cex.lab=1.
5,cex.main=1.5)
boxplot(phat4,ylim=c(min(phat2,phat4),max(phat2,phat4)),main=expression(paste(hat(pi)[C](
3), " values")),
xlab=paste("mean=",round(mean(phat4),digit=4),"std=",round(sd(phat4),digit=6)),cex.lab=1.
5,cex.main=1.5)
```

#Graphs of the theoretical approximations of alpha2, beta2 and p2 standard deviations* sqrt(N) as functions of f (proportion of passed items sample size)

#alpha2 and beta2

```
varalpha2N<-function(x)
```

```
{((1-x-beta+beta*x-alpha+alpha*x+alpha*beta)*(q-alpha)*alpha*(1-alpha))/((1-x)*x*(1-beta-
q))}

varbeta2N<-function(x)
{((alpha*beta+x-alpha*x-beta*x)*(1-beta)*(1-beta-q)*beta)/((1-x)*x*(q-alpha))}

par(mfrow=c(1,2))
par(mar=c(5,4.75,4,2)+0.1)

curve(sqrt(varalpha2N(x)),from=0.05, to=0.99,
ylim=c(0,max(max(sqrt(varalpha2N(seq(0.05,0.99,by=0.02)))),max(sqrt(varbeta2N(
seq(0.05,0.99,by=0.02)))))), xlab=paste("Proportion of passed items"),ylab=
expression(paste(Std(hat(alpha)(2)),sqrt(N))),cex.lab=1.5,cex.axis=1,lwd=3)

curve(sqrt(varbeta2N(x)),from=0.05, to=0.99,
ylim=c(0,max(max(sqrt(varalpha2N(seq(0.05,0.99,by=0.02)))),max(sqrt(varbeta2N(
seq(0.05,0.99,by=0.02)))))), xlab=paste("Proportion of passed items"),ylab=
expression(paste(Std(hat(beta)(2)),sqrt(N))),cex.lab=1.5,cex.axis=1,lwd=3)
```

#p2

```
varp2N<-function(x)
{((q-alpha)*(alpha-alpha*x-alpha*beta+beta*x)*(1-beta-q))/((1-x)*(1-beta-alpha)^2*x)}
par(mar=c(5,4.75,4,2)+0.1)
curve(sqrt(varp2N(x)),from=0.05, to=0.99,xlab=paste("Proportion of passed items"),ylab=
expression(paste(Std(hat(pi)[C](2)),sqrt(N))),cex.lab=1.5,cex.axis=1,lwd=3)
```

#Contour plots of std(alpha2)/std(alpha4), std(beta2)/std(beta4), std(p2)/std(p4)

#Alpha

```
q<-0.85
upper.beta<-(1-q)-0.01
upper.beta
alpha.grid<-rep(seq(0.01,0.1,l=10),10)
beta.grid<-rep(seq(0.01,upper.beta,l=10),rep(10,10))
grid<-data.frame(cbind(alpha.grid,beta.grid))
p<-(q-alpha.grid)/(1-beta.grid-alpha.grid)
p1<-p*(1-beta.grid)
p2<-beta.grid*p
p3<-(1-p)*alpha.grid
p4<-(1-p)*(1-alpha.grid)
p1+p2+p3+p4
ratio.alpha.sd<-sqrt((2*(-q*p3*p2-p4*p1-p4*p3+p4*q*p1)^4*(1-beta.grid-
alpha.grid+2*alpha.grid*beta.grid)*(q-alpha.grid)*alpha.grid*(-
1+alpha.grid))/(q^2*(p2+p4)*(p1+p3)*(p1*p3*p2+p2*p4*p1+p4^2*p1+p3^2*p2)*p3*p4*(-
1+beta.grid+q)*(-1+q)^2))
sdalpham<-matrix(ratio.alpha.sd,nrow=10,ncol=10)
```

```
par(mfrow=c(1,3))
par(mar=c(8,4.75,8,1)+0.1)

plot(0,0,xlim=c(0.01,0.1),ylim=c(0.01,upper.beta),xlab=expression(alpha),ylab=expression(b
eta),type="n",cex.axis=1.5,cex.lab=2)
title(sub=bquote(pi[P]==.(q)),line=6,cex.sub=2)
contour(x=unique(grid[,1]),y=unique(grid[,2]),sdalpham,add=T,lwd=3,labcex=0.75)

q<-0.9
upper.beta<-(1-q)-0.01
upper.beta
alpha.grid<-rep(seq(0.01,0.1,l=10),10)
beta.grid<-rep(seq(0.01,upper.beta,l=10),rep(10,10))
grid<-data.frame(cbind(alpha.grid,beta.grid))
p<-(q-alpha.grid)/(1-beta.grid-alpha.grid)
p1<-p*(1-beta.grid)
p2<-beta.grid*p
p3<-(1-p)*alpha.grid
p4<-(1-p)*(1-alpha.grid)
p1+p2+p3+p4
ratio.alpha.sd<-sqrt((2*(-q*p3*p2-p4*p1-p4*p3+p4*q*p1)^4*(1-beta.grid-
alpha.grid+2*alpha.grid*beta.grid)*(q-alpha.grid)*alpha.grid*(-
1+alpha.grid))/(q^2*(p2+p4)*(p1+p3)*(p1*p3*p2+p2*p4*p1+p4^2*p1+p3^2*p2)*p3*p4*(-
1+beta.grid+q)*(-1+q)^2))
sdalpham<-matrix(ratio.alpha.sd,nrow=10,ncol=10)


plot(0,0,xlim=c(0.01,0.1),ylim=c(0.01,upper.beta),xlab=expression(alpha),ylab=expression(b
eta),type="n" ,cex.axis=1.5,cex.lab=2)
title(main=expression(Std(hat(alpha)(2))/Std(hat(alpha)(3))),
sub=bquote(pi[P]==.(q)),line=6,cex.sub=2,cex.main=2)
contour(x=unique(grid[,1]),y=unique(grid[,2]),sdalpham,add=T,lwd=3,labcex=0.75)

q<-0.95
upper.beta<-(1-q)-0.01
upper.beta
alpha.grid<-rep(seq(0.01,0.1,l=10),10)
beta.grid<-rep(seq(0.01,upper.beta,l=10),rep(10,10))
grid<-data.frame(cbind(alpha.grid,beta.grid))
p<-(q-alpha.grid)/(1-beta.grid-alpha.grid)
p1<-p*(1-beta.grid)
p2<-beta.grid*p
p3<-(1-p)*alpha.grid
p4<-(1-p)*(1-alpha.grid)
p1+p2+p3+p4
ratio.alpha.sd<-sqrt((2*(-q*p3*p2-p4*p1-p4*p3+p4*q*p1)^4*(1-beta.grid-
alpha.grid+2*alpha.grid*beta.grid)*(q-alpha.grid)*alpha.grid*(-
```

```
1+alpha.grid))/(q^2*(p2+p4)*(p1+p3)*(p1*p3*p2+p2*p4*p1+p4^2*p1+p3^2*p2)*p3*p4*(-
1+beta.grid+q)*(-1+q)^2))
sdalpham<-matrix(ratio.alpha.sd,nrow=10,ncol=10)

plot(0,0,xlim=c(0.01,0.1),ylim=c(0.01,upper.beta),xlab=expression(alpha),ylab=expression(b
eta),type="n",cex.axis=1.5,cex.lab=2)
title(sub=bquote(pi[P]==.(q)),line=6,cex.sub=2)
contour(x=unique(grid[,1]),y=unique(grid[,2]),sdalpham,add=T,lwd=3,labcex=0.75)
```

#Beta

```
q<-0.85
upper.beta<-(1-q)-0.01
upper.beta
alpha.grid<-rep(seq(0.01,0.1,l=10),10)
beta.grid<-rep(seq(0.01,upper.beta,l=10),rep(10,10))
grid<-data.frame(cbind(alpha.grid,beta.grid))
p<-(q-alpha.grid)/(1-beta.grid-alpha.grid)
p1<-p*(1-beta.grid)
p2<-beta.grid*p
p3<-(1-p)*alpha.grid
p4<-(1-p)*(1-alpha.grid)
p1+p2+p3+p4
ratio.beta.sd<-sqrt((2*(p4*q*p1+p2*p1+p2*p3-q*p3*p2)^4*(1-beta.grid-
alpha.grid+2*alpha.grid*beta.grid)*(-1+beta.grid+q)*(-
1+beta.grid)*beta.grid)/(q^2*(p2+p4)*(p1+p3)*(p1^2*p4+p1*p3*p4+p2^2*p3+p2*p4*p3)*p
1*p2*(-1+q)^2*(q-alpha.grid)))
sdbetamc<-matrix(ratio.beta.sd,nrow=10,ncol=10)

par(mfrow=c(1,3))
par(mar=c(8,4.75,8,1)+0.1)

plot(0,0,xlim=c(0.01,0.1),ylim=c(0.01,upper.beta),xlab=expression(alpha),ylab=expression(b
eta),type="n",cex.axis=1.5,cex.lab=2)
title(sub=bquote(pi[P]==.(q)),line=6,cex.sub=2)
contour(x=unique(grid[,1]),y=unique(grid[,2]),sdbetamc,add=T,lwd=3,labcex=0.75)

q<-0.9
upper.beta<-(1-q)-0.01
upper.beta
alpha.grid<-rep(seq(0.01,0.1,l=10),10)
beta.grid<-rep(seq(0.01,upper.beta,l=10),rep(10,10))
grid<-data.frame(cbind(alpha.grid,beta.grid))
p<-(q-alpha.grid)/(1-beta.grid-alpha.grid)
p1<-p*(1-beta.grid)
p2<-beta.grid*p
p3<-(1-p)*alpha.grid
p4<-(1-p)*(1-alpha.grid)
```

```
p1+p2+p3+p4
ratio.beta.sd<-sqrt((2*(p4*q*p1+p2*p1+p2*p3-q*p3*p2)^4*(1-beta.grid-
alpha.grid+2*alpha.grid*beta.grid)*(-1+beta.grid+q)*(-
1+beta.grid)*beta.grid)/(q^2*(p2+p4)*(p1+p3)*(p1^2*p4+p1*p3*p4+p2^2*p3+p2*p4*p3)*p
1*p2*(-1+q)^2*(q-alpha.grid)))
sdbetamc<-matrix(ratio.beta.sd,nrow=10,ncol=10)

plot(0,0,xlim=c(0.01,0.1),ylim=c(0.01,upper.beta),xlab=expression(alpha),ylab=expression(b
eta),type="n" ,cex.axis=1.5,cex.lab=2)
title(main=expression(Std(hat(beta)(2))/Std(hat(beta)(3))),
sub=bquote(pi[P]==.(q)),line=6,cex.sub=2,cex.main=2)
contour(x=unique(grid[,1]),y=unique(grid[,2]),sdbetamc,add=T,lwd=3,labcex=0.75)


q<-0.95
upper.beta<-(1-q)-0.01
upper.beta
alpha.grid<-rep(seq(0.01,0.1,l=10),10)
beta.grid<-rep(seq(0.01,upper.beta,l=10),rep(10,10))
grid<-data.frame(cbind(alpha.grid,beta.grid))
p<-(q-alpha.grid)/(1-beta.grid-alpha.grid)
p1<-p*(1-beta.grid)
p2<-beta.grid*p
p3<-(1-p)*alpha.grid
p4<-(1-p)*(1-alpha.grid)
p1+p2+p3+p4
ratio.beta.sd<-sqrt((2*(p4*q*p1+p2*p1+p2*p3-q*p3*p2)^4*(1-beta.grid-
alpha.grid+2*alpha.grid*beta.grid)*(-1+beta.grid+q)*(-
1+beta.grid)*beta.grid)/(q^2*(p2+p4)*(p1+p3)*(p1^2*p4+p1*p3*p4+p2^2*p3+p2*p4*p3)*p
1*p2*(-1+q)^2*(q-alpha.grid)))
sdbetamc<-matrix(ratio.beta.sd,nrow=10,ncol=10)

plot(0,0,xlim=c(0.01,0.1),ylim=c(0.01,upper.beta),xlab=expression(alpha),ylab=expression(b
eta),type="n",cex.axis=1.5,cex.lab=2)
title(sub=bquote(pi[P]==.(q)),line=6,cex.sub=2)
contour(x=unique(grid[,1]),y=unique(grid[,2]),sdbetamc,add=T,lwd=3,labcex=0.75)
```

#p

```
q<-0.85
upper.beta<-(1-q)-0.01
upper.beta
alpha.grid<-rep(seq(0.01,0.1,l=10),10)
beta.grid<-rep(seq(0.01,upper.beta,l=10),rep(10,10))
grid<-data.frame(cbind(alpha.grid,beta.grid))
p<-(q-alpha.grid)/(1-beta.grid-alpha.grid)
p1<-p*(1-beta.grid)
p2<-beta.grid*p
```

```
p3<-(1-p)*alpha.grid
p4<-(1-p)*(1-alpha.grid)
p1+p2+p3+p4
ratio.p.sd<-sqrt((2*(-1+beta.grid+q)*(-alpha.grid+2*alpha.grid*beta.grid-beta.grid)*(q-
alpha.grid)*(p1+p3)^3*(p2+p4)^3)/((-
2*p1^3*p2*p4*q+p2*p4*p1^3+p1^3*p2*q^2*p4+3*p3*p4*p2*p1^2+3*p1^2*p2*q^2*p4*p
3-6*p1^2*p2*p4*q*p3+3*p1*p2^2*q^2*p3*p4-
6*p1*p2*p4*q*p3^2+3*p1*p2*q^2*p3*p4^2+3*p3^2*p4*p2*p1+p1*p2^3*q^2*p3+3*p1*p
2*q^2*p4*p3^2+p1*q^2*p3*p4^3+p2*q^2*p4*p3^3+p3^3*p4*p2-2*p2*p4*q*p3^3)*(-
1+beta.grid+alpha.grid)^2))
sdpm<-matrix(ratio.p.sd,nrow=10,ncol=10)

par(mfrow=c(1,3))
par(mar=c(8,4.75,8,1)+0.1)

plot(0,0,xlim=c(0.01,0.1),ylim=c(0.01,upper.beta),xlab=expression(alpha),ylab=expression(b
eta),type="n",cex.axis=1.5,cex.lab=2)
title(sub=bquote(pi[P]==.(q)),line=6,cex.sub=2)
contour(x=unique(grid[,1]),y=unique(grid[,2]),sdpm,add=T,lwd=3, labcex=0.75)

q<-0.9
upper.beta<-(1-q)-0.01
upper.beta
alpha.grid<-rep(seq(0.01,0.1,l=10),10)
beta.grid<-rep(seq(0.01,upper.beta,l=10),rep(10,10))
grid<-data.frame(cbind(alpha.grid,beta.grid))
p<-(q-alpha.grid)/(1-beta.grid-alpha.grid)
p1<-p*(1-beta.grid)
p2<-beta.grid*p
p3<-(1-p)*alpha.grid
p4<-(1-p)*(1-alpha.grid)
p1+p2+p3+p4
ratio.p.sd<-sqrt((2*(-1+beta.grid+q)*(-alpha.grid+2*alpha.grid*beta.grid-beta.grid)*(q-
alpha.grid)*(p1+p3)^3*(p2+p4)^3)/((-
2*p1^3*p2*p4*q+p2*p4*p1^3+p1^3*p2*q^2*p4+3*p3*p4*p2*p1^2+3*p1^2*p2*q^2*p4*p
3-6*p1^2*p2*p4*q*p3+3*p1*p2^2*q^2*p3*p4-
6*p1*p2*p4*q*p3^2+3*p1*p2*q^2*p3*p4^2+3*p3^2*p4*p2*p1+p1*p2^3*q^2*p3+3*p1*p
2*q^2*p4*p3^2+p1*q^2*p3*p4^3+p2*q^2*p4*p3^3+p3^3*p4*p2-2*p2*p4*q*p3^3)*(-
1+beta.grid+alpha.grid)^2))
sdpm<-matrix(ratio.p.sd,nrow=10,ncol=10)

plot(0,0,xlim=c(0.01,0.1),ylim=c(0.01,upper.beta),xlab=expression(alpha),ylab=expression(b
eta),type="n" ,cex.axis=1.5,cex.lab=2)
title(main=expression(Std(hat(pi)[C](2))/Std(hat(pi)[C](3))),
sub=bquote(pi[P]==.(q)),line=6,cex.sub=2,cex.main=2)
contour(x=unique(grid[,1]),y=unique(grid[,2]),sdpm,add=T,lwd=3,labcex=0.75)

q<-0.95
```

```
upper.beta<-(1-q)-0.01
upper.beta
alpha.grid<-rep(seq(0.01,0.1,l=10),10)
beta.grid<-rep(seq(0.01,upper.beta,l=10),rep(10,10))
grid<-data.frame(cbind(alpha.grid,beta.grid))
p<-(q-alpha.grid)/(1-beta.grid-alpha.grid)
p1<-p*(1-beta.grid)
p2<-beta.grid*p
p3<-(1-p)*alpha.grid
p4<-(1-p)*(1-alpha.grid)
p1+p2+p3+p4
ratio.p.sd<-sqrt((2*(-1+beta.grid+q)*(-alpha.grid+2*alpha.grid*beta.grid-beta.grid)*(q-
alpha.grid)*(p1+p3)^3*(p2+p4)^3)/((-
2*p1^3*p2*p4*q+p2*p4*p1^3+p1^3*p2*q^2*p4+3*p3*p4*p2*p1^2+3*p1^2*p2*q^2*p4*p
3-6*p1^2*p2*p4*q*p3+3*p1*p2^2*q^2*p3*p4-
6*p1*p2*p4*q*p3^2+3*p1*p2*q^2*p3*p4^2+3*p3^2*p4*p2*p1+p1*p2^3*q^2*p3+3*p1*p
2*q^2*p4*p3^2+p1*q^2*p3*p4^3+p2*q^2*p4*p3^3+p3^3*p4*p2-2*p2*p4*q*p3^3)*(-
1+beta.grid+alpha.grid)^2))
sdpm<-matrix(ratio.p.sd,nrow=10,ncol=10)

plot(0,0,xlim=c(0.01,0.1),ylim=c(0.01,upper.beta),xlab=expression(alpha),ylab=expression(b
eta),type="n",cex.axis=1.5,cex.lab=2)
title(sub=bquote(pi[P]==.(q)),line=6,cex.sub=2)
contour(x=unique(grid[,1]),y=unique(grid[,2]),sdpm,add=T,lwd=3, labcex=0.75)
```

#Ratios of std(alpha2)/std(alpha4), std(beta2)/std(beta4) and std(p2)/std(p4) as functions of f (proportion of passed items)

#assign alpha, beta and q

```
alpha
beta
q
p<-(q-alpha)/(1-beta-alpha)
#p should be smaller than 1
p<1
p1<-p*(1-beta)
p1
p2<-beta*p
p2
p3<-(1-p)*alpha
p3
p4<-(1-p)*(1-alpha)
p4
p1+p2+p3+p4
```

#Alpha

```
ratio.alpha.sd.f<-function(x)
```

```
{ratio.alpha1=sqrt(((1-x-beta+beta*x-alpha+alpha*x+alpha*beta)*(q-alpha)*alpha*(1-
alpha))/((1-x)*x*(1-beta-q)))
ratio.alpha2=sqrt(((-
1+q)^2*p4*q^2*p3*(2*p4*p2*p1*p3^2+p1*p3*p2^2*p4+2*p1*p3*p2*p4^2+p4*p2*p1^2*p
3+p2^2*p3^3+p1^2*p4^3+p4*p2*p3^3+p1^2*p4*p2^2+2*p1^2*p4^2*p2+p2^2*p3*p1^2+2
*p2^2*p3^2*p1+p1*p3*p4^3))/((q*p3*p2+p4*p1+p4*p3-p4*q*p1)^4));
ratio.alpha1/ratio.alpha2}
```

#Beta

```
ratio.beta.sd.f<-function(x)
{ratio.beta1=sqrt(((alpha*beta+x-alpha*x-beta*x)*(1-beta)*(1-beta-q)*beta)/((1-x)*x*(q-
alpha)))
ratio.beta2=sqrt((q^2*(-
1+q)^2*p2*p1*(2*p4^2*p1^2*p3+p4*p2*p1*p3^2+p4^2*p1*p3^2+p4*p2*p1^3+p1*p3*p2^
3+2*p1*p3*p2^2*p4+p1*p3*p2*p4^2+2*p4*p2*p1^2*p3+p4^2*p1^3+p3^2*p2^3+2*p3^2*
p2^2*p4+p3^2*p2*p4^2))/((-p4*q*p1-p2*p1-p3*p2+q*p3*p2)^4));
ratio.beta1/ratio.beta2}

par(mfrow=c(1,2))
par(mar=c(5,4.75,4,2)+0.1)

curve(ratio.alpha.sd.f(x),from=0.05, to=0.99,
ylim=c(0,max(max(ratio.alpha.sd.f(seq(0.05,0.99,by=0.02))),max(ratio.beta.sd.f(
seq(0.05,0.99,by=0.02))))), xlab=paste("Proportion of passed items"),ylab=
expression(paste(Std(hat(alpha)(2))/Std(hat(alpha)(3)))),cex.lab=1.5,cex.axis=1,lwd=3)
abline(h=1,col=”red”)
curve(ratio.beta.sd.f(x),from=0.05, to=0.99,
ylim=c(0,max(max(ratio.alpha.sd.f(seq(0.05,0.99,by=0.02))),max(ratio.beta.sd.f(
seq(0.05,0.99,by=0.02))))), xlab=paste("Proportion of passed items"),ylab=
expression(paste(Std(hat(beta)(2))/Std(hat(beta)(3)))),cex.lab=1.5,cex.axis=1,lwd=3)
abline(h=1,col=”red”)
```

#p

```
ratio.p.sd.f<-function(x)
{ratiop1=sqrt(((q-alpha)*(alpha-alpha*x-alpha*beta+beta*x)*(1-beta-q))/((1-x)*(1-beta-
alpha)^2*x))
ratiop2=sqrt((p3*q^2*p1*p2^3+3*p3*q^2*p1*p4*p2^2-6*p2*p3^2*p4*q*p1-
6*p2*p3*p4*q*p1^2+p4*p2*p1^3-
2*p2*p4*q*p1^3+3*p2*p3^2*p4*q^2*p1+p2*p4*q^2*p1^3+3*p4*p2*p1^2*p3+3*p2*p3*p
4*q^2*p1^2+p4*p2*p3^3+p2*p3^3*p4*q^2-
2*p2*p3^3*p4*q+3*p4*p2*p1*p3^2+3*p2*p3*q^2*p1*p4^2+p3*q^2*p1*p4^3)/((p2+p4)^3
*(p1+p3)^3));
ratiop1/ratiop2}
par(mar=c(5,4.75,4,2)+0.1)
```

```
curve(ratio.p.sd.f(x),from=0.05, to=0.99,xlab=paste("Proportion of passed items"),ylab=
expression(paste(Std(hat(pi)[C](2))/Std(hat(pi)[C](3)))),cex.lab=1.5,cex.axis=1,lwd=3)
abline(h=1,col=”red”)
```

 # Minimum Sample Size for Plan II

```
sample.size.alpha2<-function(x)
{round((q-alpha)*alpha*(1-alpha)*(1-f-beta+beta*f-alpha+alpha*f+alpha*beta)/((x^2)*f*(1-
f)*(1-beta-q)))}

sample.size.beta2<-function(x)
{round(((alpha*beta+f-beta*f-alpha*f)*beta*(1-beta)*(1-beta-q))/((x^2)*f*(1-f)*(q-alpha)))}

sample.size.p2<-function(x)
{round(((alpha+beta*f-alpha*beta-alpha*f)*(1-beta-q)*(q-alpha))/((x^2)*f*(1-f)*(1-beta-
alpha)^2))}
```