Title Page

Title of the manuscript: Estimating the number of usability problems affecting medical devices: modelling the discovery matrix

Author listing:

Vincent Vandewalle[a,b*], PhD, **Alexandre Caron**[a*], MD, MSc, **Coralie Delettrez**[c], MSc, **Renaud Périchon**[a], MSc, **Sylvia Pelayo**[a,d], PhD, **Alain Duhamel**[a,c], PhD, **Benoit Dervaux**[a,c], PhD.

*Equal contributions

Affiliations:

- [a]Univ. Lille, CHU Lille, ULR 2694 Evaluations des technologies de santé et des pratiques médicales, F-59000 Lille, France,
- [b]Inria, F-59000 Lille, France,
- [c]CHU Lille, Direction de la Recherche et de l'Innovation, F-59000 Lille, France
- [d]CIC-IT/Evalab 1403, CHU Lille, F-59000 Lille, France.

Corresponding Author: Alexandre Caron (alexandre.caron2@univ-lille.fr)

Word Count: 7355

Abstract

Background. Usability testing of medical devices are mandatory for market access. The testings' goal is to identify usability problems that could cause harm to the user or limit the device's effectiveness. In practice, human factor engineers study participants under actual conditions of use and list the problems encountered. This results in a binary discovery matrix in which each row corresponds to a participant, and each column corresponds to a usability problem. One of the main challenges in usability testing is estimating the total number of problems, in order to assess the completeness of the discovery process. Today's margin-based methods fit the column sums to a binomial model of problem detection. However, the discovery matrix actually observed is truncated because of undiscovered problems, which corresponds to fitting the marginal sums without the zeros. Margin-based methods fail to overcome the bias related to truncation of the matrix. The objective of the present study was to develop and test a matrix-based method for estimating the total number of usability problems.

Methods. The matrix-based model was based on the full discovery matrix (including unobserved columns) and not solely on a summary of the data (e.g. the margins). This model also circumvents a drawback of margin-based methods by simultaneously estimating the model's parameters and the total number of problems. Furthermore, the matrix-based method takes account of a heterogeneous probability of detection, which reflects a real-life setting. As suggested in the usability literature, we assumed that the probability of detection had a logit-normal distribution.

Results. We assessed the matrix-based method's performance in a range of settings reflecting real-life usability testing and with heterogeneous probabilities of problem detection. In our simulations, the matrix-based method improved the estimation of the

number of problems (in terms of bias, consistency, and coverage probability) in a wide range of settings. We also applied our method to five real datasets from usability testing.

Conclusions. Estimation models (and particularly matrix-based models) are of value in estimating and monitoring the detection process during usability testing. Matrix-based models have a solid mathematical grounding and, with a view to facilitating the decision-making process for both regulators and device manufacturers, should be incorporated into current standards.

Keywords: usability testing, medical device, missing data, Bayesian statistics, maximum likelihood

Main manuscript text

I. Background

A. Introduction

The usability testing is a cornerstone of medical device development, and proof of usability is mandatory for market access in both the European Union and the United States [1]. The overall objective of a usability assessment is to ensure that a medical device is designed and optimized for use by the intended users in the environment in which the device is likely to be used [2]. The goal is to identify problems (called “use errors”) that could cause harm to the user or impair medical treatment (e.g. an inappropriate number of inhalations, finger injection with an adrenaline pen, etc.) [3]. The detection of usability problems must be as comprehensive as possible because medical devices are safety-critical systems [4]. However, the total number of usability problems is never known in advance. The main challenge during the usability testing is thus to estimate this number, in order to assess the completeness of the problem discovery process [5].

In practice, participants are placed under actual conditions of use (real or simulated), and usability problems are observed and listed by human factor engineers. The experimental conditions are defined in a risk analysis that gathers together possible usability problems. Throughout the usability testing, problems are discovered and added to a discovery matrix - a binary matrix with the participants as the rows and the problems as the columns. The current approach involves estimating the total number of problems as the usability testing progresses, starting from the first sessions. The number is estimated iteratively as the sample size increases, until the objective of completeness has been achieved [6].

From a statistical perspective, the current estimation procedure is based on a model of how the usability problems are detected; this is considered to be a binomial process. The literature suggests that the total number of usability problems can be estimated from the discovery matrix's problem margin (the sum of the columns) [7-11]. However, this estimation is complicated by (i) the small sample size usually encountered in usability testing of medical devices [12] and (ii) as-yet unobserved problems that truncate the margin and bias estimates [13-15].

The objective of the present study was to develop a matrix-based estimation of the number of usability problems affecting a medical device. This new method is based on the likelihood of the discovery matrix (rather than the matrix's margins alone), so as to avoid a reduction in the level of information prior to modeling. The method's main targets are (i) regulatory agencies and notified bodies involved in the pre-market evaluation of medical devices, and (ii) medical device manufacturers (more specifically, the human factors engineers in charge of ensuring that the devices are usable).

B. Data collected during the usability testing: the discovery matrix

The human factor engineer collects the results of the usability testing in a problem-discovery matrix $\mathbb{d}$. Each row corresponds to a participant, and each column corresponds to a usability problem. The result is 1 if the participant discovered the problem and 0 if not. Considering that after the inclusion of n participants, j problems have been discovered, a $n \times j$ matrix is built. By way of an example, the discovery matrix obtained after $n = 8$ participants (in rows) might be the one presented below:

95 $$\mathbb{d} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

96 In this example, $j = 10$ different problems (in columns) have been detected so far. The
97 first participant discovered only one problem (column 1), whereas the second discovered
98 two new problems (columns 2 and 3), etc.

99 At this stage, some problems might not have been detected, and the total number of
100 usability problems (m) is unknown. It should be noted that by definition, $m \geq j$ and $m -$
101 j problems remain undetected. Indeed, $\mathbb{d}$ comes from a complete but unobserved matrix
102 of dimensions $n \times m$. This matrix is denoted as $\mathbb{x}$. Thus, the "observed" matrix $\mathbb{d}$ is a
103 truncated version of the "complete" matrix $\mathbb{x}$; it lacks the columns corresponding to the
104 as-yet undetected problems. Hereafter, we use the following notation: $\mathbb{x} = (x_{il})_{1 \leq i \leq n, 1 \leq l \leq m}$
105 where $x_{il} = 1$ if the participant i experiences the problem l, and $x_{il} = 0$ otherwise.

106 $$\mathbb{x} = \begin{pmatrix} x_{11} & \cdots & x_{1l} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i1} & \cdots & x_{il} & \cdots & x_{im} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nl} & \cdots & x_{nm} \end{pmatrix}$$

107 The human factor engineer's goal is to estimate the total number of problems m from the
108 discovery matrix $\mathbb{d}$ and thus deduce $m - j$ - the number of problems that have not been
109 detected. The new method presented below addresses this goal.

C. Conventional estimation of m using a margin-based probabilistic model

In this section, we describe the margin-based methods currently employed to estimate the number of usability problems. As mentioned above, m is currently estimated by fitting a probabilistic (binomial) model to the discovery matrix's problems margin. More specifically, the probability with which a given usability problem is discovered by a participant is modelled by a Bernoulli trial with a probability of success (i.e. detection) p. For a given problem, the Bernoulli trial is considered to apply independently to each of the n participants in the usability testing. Thus, the problem margin sums can be considered as an independent, identically distributed sequence of Bernoulli trials, in which the number of times a given usability problem (a random variable X) has been observed after n participants follows a binomial distribution, $X \sim \mathcal{B}in(n, p)$. Considering the binomial distribution of the margin sums, the proportion of problems that has been discovered at least once after n participants is given by the cumulative function of the shifted geometric distribution [6, 16, 17]:

$$P(X > 0) = 1 - (1 - p)^n \quad (1)$$

The total number of problems m is then deduced from the following relationship:

$$j = (1 - (1 - p)^n) \times m \quad (2)$$

The discovery progress is thus assessed in two steps: the probability of detection p is first estimated and then plugged into Equation (2) to estimate the number of problems m. A wide range of literature methods are available for estimating the probability of problem detection. The simplest way involves computing the naive estimate (denoted as $\hat{p}$) using

the observed discovery matrix $\mathbb{d}$, considering that only j problems have been detected so far:

$$\hat{p} = \frac{\sum_{i=1}^{n}\sum_{l=1}^{j} x_{il}}{n * j} \quad (3)$$

As mentioned above, the naïve estimate is systematically biased - especially for small samples. Indeed, unobserved problems result in zero columns that shrink the probability space and lead to overestimation of p, particularly at the beginning of the process when $j \ll m$. Consequently, m is systematically underestimated, which generates safety concerns in the medical device field. In response, several strategies have been employed to overcome the truncated matrix problem.

In 2001, Hertzum and Jacobsen suggested normalizing the value of $\hat{p}$ [9]. This procedure considers that the lower boundary of the probability of detection estimated with n participants is $1/n$. For example, in a sample of 5 participants, $\hat{p} \in [0.2\,;1]$. Conversely, the normalized estimator $\hat{p}_{Norm} \in [0;1]$, and is computed as follows:

$$\hat{p}_{\text{Norm}} = \frac{\hat{p} - \frac{1}{n}}{1 - \frac{1}{n}} \quad (4)$$

However, the normalized approach suffers from a major limitation when estimating the total number of problems with Equation (4). In fact, if each participant has discovered only one problem and if each problem was discovered only once, $\hat{p} = \frac{1}{n}$, $\hat{p}_{Norm} = 0$, and the estimated number of problems $\widehat{m}$ is infinite. We will not discuss this estimation method further.

Turing and Good developed a discounting method for estimating the probability of unseen species on the basis of observed data [18]. Lewis suggested that the Good-Turing (GT)

adjustment could be used to reduce the magnitude of the overestimation of p by increasing the probability space and thus accounting for unobserved usability problems [8]. The GT adjustment is computed as the proportion of singletons relative to the total number of events (i.e. the proportion of problems discovered only once, $x_{il} = 1$), and is incorporated in the estimation as follows:

$$\hat{p}_{GT} = \frac{\hat{p}}{1 + GT} \quad (5)$$

However, Lewis observed that use of the GT estimator overestimated p. He empirically assessed the best adjustment for a small sample size by carrying out Monte Carlo simulations on a range of usability testing databases involving web or software user interfaces with known true values. Based on these simulations, Lewis concluded that the best method was to average the GT adjustment and a "double-deflation" term:

$$\hat{p}_{\text{double-deflation}} = \frac{1}{2}\left[\frac{\hat{p}}{1 + GT_{adj}}\right] + \frac{1}{2}\left[\left(\hat{p} - \frac{1}{n}\right) \times \left(1 - \frac{1}{n}\right)\right] \quad (6)$$

Nevertheless, the degree of adjustment of the probability space for unobserved problems is essentially empirical. The residual bias is not known to trend towards over- or underestimation.

In 2009, Schmettow considered the problem margin sums in a zero-truncation framework [19]. Indeed, the distribution of the problems so far observed follows a binomial distribution with only a positive integer as support (i.e. a positive or conditional distribution). The distribution is zero-truncated because problems only appear in the discovery matrix once they have been discovered. The probability is then estimated using standard mathematical techniques, such as the maximum likelihood or moment estimator [20-22]. The probability mass function is:

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (7)$$

and zero truncation is achieved as follows:

$$P(X = k)_{\text{zt}} = \begin{cases} 0 & if\ k = 0 \\ \dfrac{P(X = k)}{1 - P(X = 0)} & if\ k > 0 \end{cases} \quad (8)$$

The probability of problem discovery is then estimated by using maximum likelihood techniques to fit the marginal sums to the zero-truncated binomial distribution. It should be noted that the expected probability of unobserved problems, $\Pr(X = 0)$, is deduced from the non-truncated function [19].

D. Methods taking account of a heterogeneous problem detection probability

All the methods presented above assume that the probability of detection is the same for all usability problems (i.e., the same p). However, this assumption is unrealistic and does not hold true in real-life usability testing. Schmettow showed that overdispersion was frequent in the problem margin sums, reflecting heterogeneity in the probability of detection [23]. Furthermore, erroneously ignoring the presence of heterogeneity by using a single, average value of p leads to overestimation of the completeness of the discovery process (Jensen's inequality) [24]. Schmettow tackled this problem by developing a model that incorporated heterogeneity. The probability of detection was considered to be a random variable, which enabled each problem to have its own probability of detection. Schmettow used the logit-normal distribution as a plugin distribution for the probability of detection. Formally, the logit of the probability of detection follows a normal distribution $\mathcal{N}(\mu, \sigma)$. In this model, the problem margin sums follows a logit-normal binomial distribution and the probability mass function is:

$$P(X=k)=\binom{n}{k}\frac{1}{\sqrt{2\pi}\sigma}\int_0^1(1-p)^{n-k-1}p^{k-1}\exp\left(-\frac{(logit(p)-\mu)^2}{2\sigma^2}\right)dp \qquad (9)$$

Using the zero truncation technique presented in equation (8), Schmettow developed the logit-normal binomial zero truncated (LNBzt) model and applied it to the usability of medical infusion pumps [25]. To the best of our knowledge, this model is the only one that accounts for both heterogeneity and unobserved problems.

E. Statistical limitations of margin-based methods

The primary limitation of the margin-based methods presented above is that they estimate the probability of detection only. The number of problems m is deduced but not estimated *per se*. It would be possible to estimate both m and p by summarizing the discovery matrix on the basis of the participants' margin. In such a case, each sum follows a binomial $\mathcal{B}in(m,p)$, thus enabling estimation of both the number of attempts and the probability of success in a binomial setting. However, DasGupta and Rubin established that there were no unbiased estimates for essentially any functions of either the number of attempts or the probability of success [26]. This problem was initially considered by Fisher and Haldane for estimating species abundance [27, 28]. It has also been considered by Olkin, Petkau, and Zidek, who developed both a moment and a maximum likelihood estimator, and by Carroll and Lombard, who proposed an estimator in a Bayesian setting (leading to a beta-binomial distribution) [29, 30]. Hall also considered this problem in an asymptotic framework [31].

The second limitation of margin-based methods is information loss, relative to the initially available data. For example, j and the number of singletons were the only data used in the GT estimates. In the same way, the zero-truncated method considered only the column sums for the problems and omitted the pattern of detection (i.e., the users).

Here, we tackle these problems by directly modelling the full discovery matrix (including unobserved columns) and not only a summary of the data (e.g. the margins). In the Methods section, we describe the statistical basis of the matrix-based method and detail a Bayesian approach for estimating the number of problems. In the Results section, we compare the matrix-based method's statistical properties with those of existing models in a simulation study and then in actual usability studies. Lastly, we discuss the implications of our results with regard to estimation of the number of problems in usability testing.

II. Methods

We first specify the statistical basis underpinning the matrix-based method, and the principle of column permutation in particular. Next, we present our estimation of the number of problems in a Bayesian setting. The last part is dedicated to the methods used to assess the matrix-based model's performance.

A. The matrix-based method

We first present the matrix-based method. For the sake of clarity, we simplified the problem by considering that the probability of problem detection was homogeneous. The concept of heterogeneous probability will be introduced in the second part of this section, along with the Bayesian estimation.

1. Presentation of the method

Consider the complete discovery matrix $\mathbb{x}$. The probability of $\mathbb{x}$ can be written as follows:

$$P(\mathbb{x}|p,m) = p^{\mathbb{x}_{\bullet\bullet}}(1-p)^{nm-\mathbb{x}_{\bullet\bullet}} \quad (10)$$

where $\mathbb{x}_{\bullet\bullet} = \sum_{i=1}^{n}\sum_{l=1}^{m} x_{il}$ is the total number of problems observed by n participants.

An example of a possible matrix $\mathbb{x}$ obtained from two participants during a usability testing of a medical device with $m = 3$ problems is given below (with users in rows and problems in columns):

$$\mathbb{x} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad (11)$$

As seen above, the complete discovery matrix $\mathbb{x}$ is never observed, and the discovery matrix $\mathbb{d}$ is the only one available. It is similar to the matrix $\mathbb{x}$, except that unobserved problems are missing. Considering the above example, neither of the users observed the second problem, and the resulting observed discovery matrix $\mathbb{d}$ would be:

$$\mathbb{d} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (12)$$

It should be noted that if the total number of problems m is known, then the complete matrix $\mathbb{x}$ could be reconstituted (with permutation), based on the matrix $\mathbb{d}$. For instance, if we take the matrix $\mathbb{x}$ and consider (wrongly, in this case) that the number of problems $m = 5$, then the reconstituted complete matrix denoted by $\hat{\mathbb{x}}^m$ would be obtained by padding the matrix $\mathbb{d}$ with columns of zeros (corresponding to as-yet unobserved problems):

$$\hat{\mathbb{x}}^{m=5} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad (13)$$

Thus, noting that $\mathbb{x}_{\bullet\bullet} = \mathbb{d}_{\bullet\bullet}$, it is possible to compute the likelihood of the complete matrix $\hat{\mathbb{x}}^m$ on the basis of the discovery matrix $\mathbb{d}$. This likelihood is given by the following equation:

$$P(\hat{\mathbb{x}}^m | p, m) = p^{\mathbb{x}_{\bullet\bullet}} (1-p)^{nm - \mathbb{x}_{\bullet\bullet}} \quad (14)$$

Note that the definition of $\mathbb{x}^m$ depends on the value m, which is unknown. Thus, any inference based on $\mathbb{x}^m$ will induce some bias. For instance, a maximum likelihood estimation of (p, m) based on $\mathbb{x}^m$ (consisting in maximizing $p(\mathbb{x}^m|p, m)$ with respect to m and p) leads to $\hat{m} = j$ (where j is the number of problems observed so far) and $p = \frac{\mathbb{x}_{\bullet\bullet}}{nj}$, which are known to be biased. We tackled this issue by modeling the distribution of the observed discovery matrix $p(\mathbb{d}|p, m)$.

It should be noted that the matrix $\mathbb{d}$ is defined in a lexicographic order, which simply means that the problems are ordered in the order of detection. For instance, the six possible complete matrices $\mathbb{x}$ leading to the previous matrix $\mathbb{d}$ if $m = 3$ are presented in Table 1.

Table 1: Six possible complete matrices $\mathbb{x}^{m=3}$ leading to the observed discovery matrix $\mathbb{d} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

Possibility 1	Possibility 2	Possibility 3
$\mathbb{x}_1^{m=3} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\mathbb{x}_2^{m=3} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\mathbb{x}_3^{m=3} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$
Possibility 4	**Possibility 5**	**Possibility 6**
$\mathbb{x}_4^{m=3} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$	$\mathbb{x}_5^{m=3} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$	$\mathbb{x}_6^{m=3} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$

In fact, if we could consider the label (the name of the usability problem) associated with each column, only one matrix $\mathbb{x}$ could lead to the matrix $\mathbb{d}$. However, since we have no means of finding the names of the columns in the initial matrix $\mathbb{x}$, we will consider that the matrix $\mathbb{d}$ has unnamed columns. Removing these column names allows us to consider the matrix $\mathbb{d}$ for the observed data (for which the definition does not vary as a function of the model's definition of the model – in contrast to $\mathbb{x}^m$). Thus:

$$P(\mathbb{d}|m=3,p)=\sum_{h=1}^{6}P(\hat{\mathbb{x}}_h^{m=3}|m=3,p) \quad (15)$$

and more generally

$$P(\mathbb{d}|m,p)=\sum_{h=1}^{H(\mathbb{d},m)}P(\hat{\mathbb{x}}_h^{m}|m,p) \quad (16)$$

where $H(\mathbb{d},m)$ is the number of different matrices $\hat{\mathbb{x}}_h^m$ with m columns leading to the same discovery matrix $\mathbb{d}$.

In the simple example presented above (Table 1), $H(\mathbb{d},m)=6$ and each matrix $\hat{\mathbb{x}}_h^m$ has the same probability, i.e. $p^2(1-p)^4$. It follows that:

$$P(\mathbb{d}|m=3,p)=\mathrm{H}(\mathbb{d},\mathrm{m}=3)\times P(\hat{\mathbb{x}}_h^{m=3}|m=3,p)= 6\times p^2(1-p)^4=A_3^2\times p^2(1-p)^4 \quad (17)$$

More generally, the number of matrices $\mathbb{x}$ with m columns associated with an observed discovery matrix $\mathbb{d}$ is:

$$\mathrm{H}(\mathbb{d},\mathrm{m})=\frac{m!}{(m-j)!\,j_1!\ldots j_r!}=\frac{1}{j_1!\ldots j_r!}\times A_m^j \quad (18)$$

where r is the number of different columns of $\mathbb{d}$, and j_h $(1\le h\le r)$ is the number of repetitions of the column of type h. Of course, $j=j_1+\cdots+j_r$. Here, we recognize a familiar equation: that associated with the number of anagrams of a word in which each type of column corresponds to a different letter, including the null column (repeated $m-j$ times).

Lastly, since each matrix $\hat{\mathbb{x}}_h^m$ has the same probability, we obtain the likelihood of $\mathbb{d}$ as follows:

$$P(\mathbb{d}|p,m) = \frac{1}{j_1! \dots j_r!} \times A_m^j \times P(\widehat{\mathbb{x}}_h^m|m,p) \quad (19)$$

In practice, the computation of $\frac{1}{j_1! \dots j_r!}$ has no impact on the estimation, since it is the same for all values of m and p. This result is not limited to the homogenous setting and would remain valid for any probability of $\mathbb{x}$ with a column-wise exchangeability property.

In the particular case of the homogeneous setting, we obtain:

$$P(\mathbb{d}|p,m) = \frac{1}{j_1! \dots j_r!} \times A_m^j \times p^{\mathbb{x}_{\bullet\bullet}}(1-p)^{nm-\mathbb{x}_{\bullet\bullet}}. \quad (20)$$

In the homogeneous setting, our matrix-based approach could be extended to perform maximum likelihood inference or Bayesian inference on the parameters. However, as explained above, this setting is unrealistic in practice and so a heterogeneous probability of detection should be considered in the following section.

2. Heterogeneity and Bayesian estimation

We considered a heterogeneous probability of detection; i.e. each problem l has its own probability of detection p_l. In line with Schmettow's method, we assume that the probabilities of detection are independent and follow a logit-normal distribution, i.e. $\text{logit}(p_l) \sim \mathcal{N}(\mu, \sigma)$. The model's parameters are m, μ and σ. Note that $p_1, \dots, p_m$ are considered as latent random variables - like random effects in the mixed model.

Given these parameters, the likelihood of the discovery matrix $\mathbb{d}$ can be written as

$$P(\mathbb{d}|\mu,\sigma,m) = \int_0^1 \dots \int_0^1 P(\mathbb{d}|p_1, \dots, p_m, m) f(p_1, \dots, p_m|\mu,\sigma) dp_1 \dots dp_m \quad (21)$$

where $f(p_1, p_2, \dots, p_m|\mu,\sigma)$ is the probability density function of $p_1, p_2, \dots, p_m$. Given that the columns are exchangeable, we can also write

$$P(\mathbb{d}|\mu,\sigma,m) = \frac{1}{j_1!\dots j_r!} \times A_m^j \times P(\hat{\mathbb{x}}_h^m|\mu,\sigma,m) \quad (22)$$

which will be useful for subsequent computations.

We now consider a Bayesian framework [32] for estimation of the parameters. This framework has good theoretical properties and can include prior knowledge about the problem's parameters. Indeed, the distribution of the parameters $P(\mu,\sigma,m)$ must first be defined. Moreover, assuming the prior independence of μ, σ and m, $P(\mu,\sigma,m) = P(\mu)P(\sigma)P(m)$. We assume a prior uniform distribution for m:

$$P(m) = \frac{1}{M} \forall m \in \{1,\dots,M\} \quad (23)$$

The value of M is the pre-determined upper boundary for m, and should be chosen by the human factor engineer according to the expected maximum possible number of problems. To prevent underestimation, a high value should be used. However, if M is unnecessarily high, it will lead to an increase in the computing time.

Since our goal here is to estimate the number of problems, our main interest is $P(m|\mathbb{d})$, which is obtained using Bayes' theorem:

$$P(m|\mathbb{d}) = \frac{\mathrm{P}(m) \times P(\mathbb{d}|m)}{\sum_{m'=1}^{M} \mathrm{P}(m') \times P(\mathbb{d}|m')} \quad (24)$$

Thus, we need to compute $P(\mathbb{d}|m)$ for each possible value of m in $\{1,\dots,M\}$. This computation requires computation of the integrated likelihood $P(\mathbb{d}|m)$, as follows

$$P(\mathbb{d}|m) = \int_0^{+\infty} \int_{-\infty}^{+\infty} P(\mathbb{d}|\mu,\sigma,m)P(\mu)P(\sigma)d\mu\, d\sigma \quad (25)$$

The choice of prior distributions for $P(\mu)$ and $P(\sigma)$ is discussed below. $P(\mathbb{d}|m)$ can be computed by approximating this integral with Markov chain Monte Carlo (MCMC) techniques.

Even though $P(m|\mathbb{d})$ is the main quantity of interest, $P(\mu|\mathbb{d})$ and $P(\sigma|\mathbb{d})$ are also of interest because they can be used as prior distributions for future studies; this will decrease the sample size and improve early estimates as part of an early control strategy.

3. Computational aspects

From a computational perspective, and since $P(\mathbb{d}|\mu,\sigma,m) = \frac{1}{j_1!\dots j_r!} \times A_m^j \times P(\hat{\mathbb{x}}_h^m|\mu,\sigma,m)$, we will first focus on the computation based on $\hat{\mathbb{x}}_h^m$ and will then deduce the results for $\mathbb{d}$.

Let now consider the choice of a prior distribution for μ and σ. Since μ and σ are Gaussian distribution parameters and in the absence of additional information (e.g. from previous usability studies), we chose the following flat priors:

- $\mu \sim \mathcal{N}(0;\mathcal{A})$: a Gaussian distribution with a high variance $\mathcal{A}$, (e.g. $\mathcal{A} = 10^8$), mimicking a uniform distribution on $\mathbb{R}$,
- $\sigma^2 \sim \text{inv}-\chi_\nu^2$: an inverse chi-squared distribution with ν degrees of freedom (typically $\nu = 1$).

When the data has a Gaussian distribution, choosing the above priors leads to a conjugated posterior distribution. However, a logistic-normal distribution of the probabilities of detection means that conjugacy cannot be obtained. Thus, estimation of the posterior distribution required the use of MCMC methods. This consisted in drawing μ and σ for each possible value of m, $m \in 1, \dots, M$ according to their posterior distribution

$P(\mu, \sigma|m, \mathbb{d})$, and deducing a numerical approximation of $P(\mathbb{d}|m)$ from the Monte-Carlo sample. Lastly, $P(m|\mathbb{d})$ was computed using Bayes' theorem.

For a fixed value of m, we consider sampling from $P(\mu, \sigma|\hat{\mathbb{x}}_h^m, m)$, computing the integrated likelihood $P(\hat{\mathbb{x}}_h^m|m)$ with bridge sampling [33], and deducing $P(\mathbb{d}|m)$.

The parameters μ and σ (given $\hat{\mathbb{x}}_h^m$ and m) are sampled using the parameter space augmented by $p_1, \ldots, p_m$, i.e. the discovery probabilities associated with each column of $\hat{\mathbb{x}}_h^m$. Thus, we will now sample from $\mu, \sigma, p_1, \ldots, p_m|\hat{\mathbb{x}}_h^m$, using stan software (adaptative Hamiltonian Monte Carlo algorithm).

B. Assessment of the performance of the matrix-based method

We compared the performance of five methods (naïve, GT, double-deflation, LNBzt, and matrix-based methods) first in a simulation study and then using literature data from actual usability studies.

1. Simulation study

Each simulation consisted in generating an observed discovery matrix $\mathbb{d}$ from the usability testing of a hypothetical medical device with a known total number of usability problems m and a sample size n. The probability of detection was normally distributed ($\mathcal{N}(\mu, \sigma)$) on a logit scale. The combinations of parameters used in the simulations are specified in Table 2. The values were chosen to reflect a wide range of parameters encountered in usability testing of medical devices.

Table 2: Combinations of parameters for the simulation testing with homogeneous and heterogeneous probabilities of detection.

Parameter	Values
Total number of usability problems	$m = 20,50,100$
Sample size	$n = 15,20,30,40,50$
Probability of problem detection	$\mu = logit(0.1), logit(0.2)$ $\sigma = 0.5, 1, 2$
Number of combinations tested	**90**

In each setting (i.e. for each combination of m, μ, σ and n), we simulated $S = 2 \times 10^4$ complete discovery matrices, $\mathbb{x}_{m,\mu,\sigma,n,i}$, $i \in \{1,2,\dots,S\}$.. The matrices $\mathbb{d}$ were obtained by truncation of the zero columns (problems not yet discovered). We averaged the estimates of m over the S simulations and computed the 95% fluctuation interval (0.025 and 0.975 quantiles). We also calculated the prediction's root mean square error (RMSE) as the square root of the mean square difference between the predicted and true values of m:

$$RMSE(m) = \sqrt{\frac{1}{S}\sum_{i=1}^{S}(m - \hat{m}_i)^2} \qquad (26)$$

When the sample is small, little information is available; a tight credible interval might reflect overconfidence rather than a good estimation. Thus, to gauge the level of confidence that human factor engineers can place in each method, we computed the coverage probability. In each setting, this is the proportion of 95% confidence intervals for the simulated $\hat{m}_i$ that include the true value of m. The confidence intervals for $\hat{m}_i$ were computed using 1000 parametric bootstrap repetitions with the parameters $(\hat{m}_i, \hat{\mu}_i, \hat{\sigma}_i, n)$. For the matrix-based method, we were able to directly compute the 95% confidence interval of the posterior distribution of each simulation, which saved substantial computation time.

2. Application to actual usability studies

We applied the above-described methods to the discovery matrices of five published usability studies. Four did not involve a medical device: the EDU3D dataset encompassed 119 problems discovered by 20 participants during the evaluation of virtual environments [34], the MACERR dataset encompassed 145 problems discovered by 15 participants during a scenario-driven usability testing of an integrated office system [35],

the MANTEL dataset encompassed 30 problems submitted by 76 expert participants evaluating the specifications of a computer program, and the SAVINGS dataset encompassed 48 usability problems discovered by 34 participants on voice response systems MANTEL and SAVINGS comes from the same experiment on heuristic evaluations [36]. These four studies were included because they have been used in important publications in this field [8] and they enabled us to address heterogeneity in the probability of discovery, in particular [23]. The fifth usability testing involved a medical device: INFPUMP encompassed 107 usability problems discovered by 34 participants (intensive care unit nurses and anesthesiologist) evaluating a prototype medical infusion pump [25].

For each of the five datasets, we computed the estimates and the 95% confidence intervals for the final data. When a sufficient number of participants had been included (i.e. for MANTEL, SAVINGS, and INFPUMP), we addressed the change in the estimates as a function of the sample size.

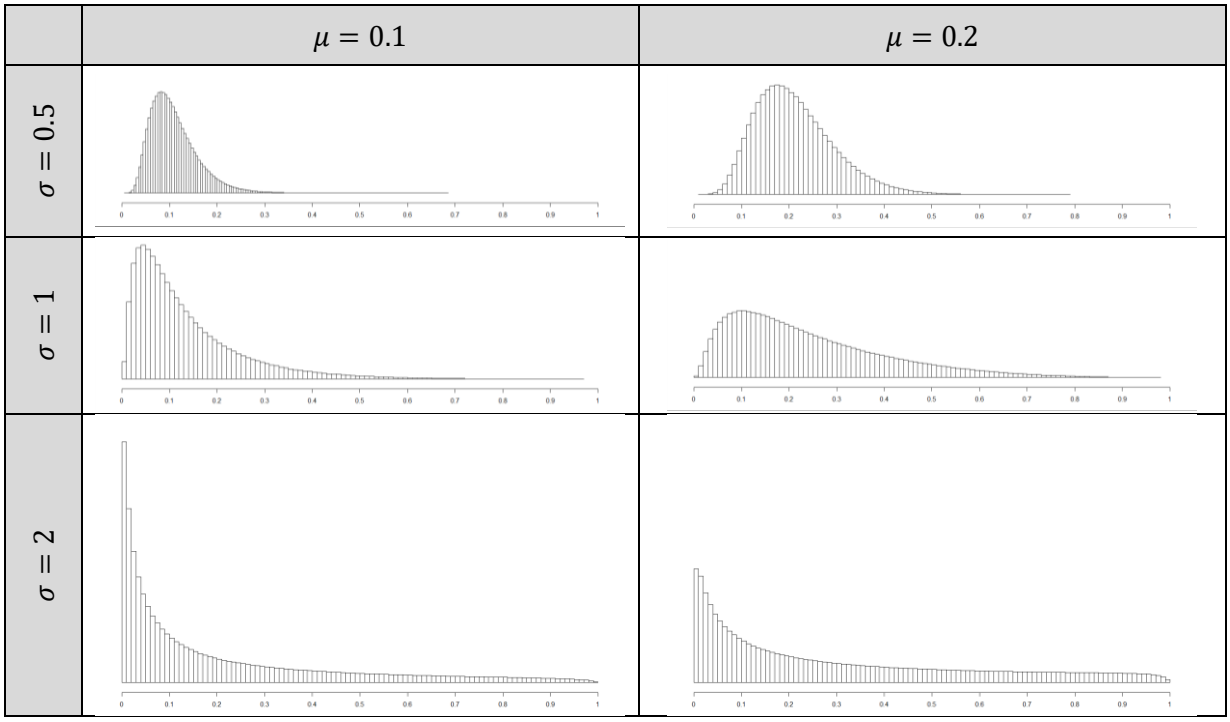
All the analyses were carried out running R software (version 3.6.1) on several servers equipped with 12-core Intel® Xeon® E5-2650 v4 processors (http://hpc.univ-lille.fr/cluster-hpc-htc). The MCMC was performed using the *Stan* library (http://mc-stan.org) via the *rstan* package [37]. The integrated likelihood was obtained using the *bridge_sampler* function of the *bridgesampling* package [38]. The code [see Additional file 1] and a detailed tutorial [see Additional file 2] are provided as supplementary material. In order to facilitate the matrix-based method's application in practice, a short step-by-step tutorial is provided at the end of this manuscript.

III. Results

A. The simulation study

The distributions of the probability of detection for each setting are summarized in Table 3. The distribution shifted to a highest average probability of detection when μ increased. It is noteworthy that a higher dispersion (σ) not only flattened the distribution but also led to an increase in probability of very rare problems.

Table 3: Distribution of the probability of detection as a function of μ and σ. The probability of detection followed a logit-normal distribution: $logit(p_i) \sim \mathcal{N}(\mu, \sigma)$.



The results of the simulation are presented for the five methods (naïve, GT, double-deflation, LNBzt, and matrix-based). The prediction error of m as a function of the sample size n are presented in Figure 1. The RMSE is presented in Figure 2. A tabulated version of these data is also provided as supplementary material (S-Table 5 and S-Table 6). As mentioned by Schmettow, extreme estimates of m can be obtained with the LNBzt method when the number of singletons is high. We decided to discard any results with

439 $\widehat{m}_{LNBzt} > 500$, to avoid penalizing the method with estimates that would not be realistic
440 in real life [19].

441 As expected, the accuracy of the estimation of the number of problems increased with the
442 sample size for all estimates, with less bias and greater consistency (i.e. the RMSE tended
443 towards zero as the sample size increased). Likewise, the estimates were better as the
444 number of problems to discover m increased. For all methods, the bias was higher as the
445 number of "rare" problems increased (i.e. for a higher σ).

446 **Methods accounting for heterogeneity: the matrix-based and LNBzt estimates**

447 The matrix-based method showed less bias overall; the bias ranged from -8.5% to +14.7%
448 for the 90 simulated combinations. This range was narrower (from -5.1 to +1.2%) when
449 the participant sample size was 30 or more. In contrast, the LNBzt method displayed
450 systematic upward bias; although the lower boundary was -0.1%, the upper boundary
451 was 54.7%. This bias was still observed for 30 participants, with an upper boundary of
452 23.8%.

453 When $\sigma = 2$, the matrix-based method underestimated the number of problems.
454 However, this underestimation was less than -5.1% for $n \geq 30$. For lower values of σ, the
455 matrix-based method's bias ranged from -2.6% to +1.2% for $n \geq 30$. The bias associated
456 with the LNBzt method was high for $\sigma = 2$. Although the bias decreased with n, it was still
457 +11.8% for $n = 50$. For a lower value of σ, the bias associated with the LNBzt method
458 ranged from -2.6% to +1.2% for $n \geq 30$.

459 The matrix-based method gave the lowest RMSE in all settings. This was particularly true
460 when the number of "rare" problems was high ($\sigma > 0.5$). The LNBzt gave the highest
461 average RMSE. As mentioned in the Methods, this bias resulted from a few very high

estimates of m, which increased the average RMSE dramatically. This was true for the lowest average probability of detection (i.e. $\mu = \text{logit}(0.1)$) and the highest variance (i.e. $\sigma = 2$).

Methods not accounting for heterogeneity: the naïve, GT, and double-deflation estimates

The estimates that did not take account of heterogeneity showed the strongest bias. The naïve estimate was the worst; it systematically underestimated the true value of m (range: -33.2% to -0.2%). This underestimation was slightly lower for the GT estimate, especially when σ was low. However, the range was still broad: from -32.2% to -0.2%. The double-deflation method compensated even more for underestimation but sometimes led to overestimation (range: -32.0% to +8.6%).

When σ was lower (i.e. 0.5 or 1), the trend towards underestimation was less pronounced for the double-deflation and the GT methods (with lower boundaries of -14.1% and -17.2, respectively) than for the naïve method (lower boundary: -22.8%. The bias persisted for larger sample sizes: it was still as high as -6.4% for the three methods for $n = 50$.

The naïve RMSE estimate was again the worst of the methods that did not take account of heterogeneity. Although the GT and the double-deflation methods gave acceptable RMSEs, this feature must be interpreted with caution. In fact, the acceptable RMSEs resulted essentially from systematic underestimation, which in turn limited the range of possible $\widehat{m}$ (which can never be lower than j). Hence, the interpretation of the RMSE was limited for these methods.

Coverage probability.

As explained in the Methods, human factor engineers do not know the variables for the usability testing they are carrying out. The coverage probability enables them to study the reliability of the estimate (and its 95% confidence interval). A tabulated version of the data is provided as supplementary material (S-Table 7).

For the matrix-based method, the coverage probability was always over 80% (except for $m = 100, n = 15, \mu = logit(0.1)$, and $\sigma = 0.5$, where the probability of coverage dropped to 72%) with an average of 94% over the range of settings tested in the simulations study. The probability was at least 81% for $n \geq 20$ and at least 88% for $n \geq 30$. The LNBzt method's coverage probability was always over 80%, with an average of 92%. The LNBzt performed particularly well for small sample sizes, with a minimum coverage of 89% for $n = 15$, of 86% for $n = 20$, and of 82% for $n = 30$. Indeed, the LNBzt method provided the broadest confidence intervals of the five methods studied here. It is noteworthy that the LNBzt method was the only one that sometimes failed to fit the data (in 33% of cases). However, it was impossible to adjust the method's parameter for each individual simulation. In practice, changing the optimization function's starting values would avoid most of the fitting failures.

The methods not taking account of heterogeneity provided a low, erratic coverage probability in most settings. On average, the coverage probabilities were 17.9%, 31.5% and 33.7% for the naïve, GT, and double-deflation methods, respectively. Furthermore, the three methods frequently yielded excessively high estimated levels of confidence - especially for high values of m.

Lessons learned from the simulation study

From the human factor engineer's point of view, the matrix-based and LNBzt methods are the only reliable ones; they gave a good coverage probability in almost any setting and for

almost any sample size. Conversely, the methods not taking account of heterogeneity were unreliable and so could not be trusted.

B. Application to real data from published usability studies

The estimated number of problems computed from the discovery matrices of five published usability studies are presented in Table 4. Although the real number of problems is not known, we can compare the matrix-based method's predictions with those of the other methods (and especially the LNBzt method).

Table 4: The estimated number of problems for five real datasets from published usability studies.

	n*	j**	naïve	Good-Turing	double deflation	LNBzt	matrix-based
EDU3D	**20**	**119**					
$\hat{m}$			120	121	122	155	152
95%CI			117 – 121	118 – 125	120 – 129	132 – 195	135 – 167
MACERR	**15**	**145**					
$\hat{m}$			156	178	184	449	382
95%CI			146 – 160	171 – 207	192 – 245	256 – 1301	346 – 440
MANTEL	**76**	**30**					
$\hat{m}$			30	30	30	31	30
95%CI			30 – 30	30 – 30	30 – 30	31 – 35	30 – 37
SAVINGS	**34**	**44**					
$\hat{m}$			44	44	44	46	45
95%CI			44 – 45	44 – 45	44 – 45	42 – 50	44 – 51
INFPUMP	**34**	**107**					
$\hat{m}$			107	107	107	122	120
95%CI			107 – 108	106 – 108	106 – 108	110 – 136	112 – 143

* n is the number of participants in the study
** j is the number of problems discovered after analyses by n participants

In these five datasets, the number of participants ranged from 15 to 76. Previous studies of these datasets [8, 19, 23, 25] demonstrated that the probability of problem detection was heterogeneous. As suggested by the results of the simulation study, the methods not taking account of heterogeneity considered that the discovery process was complete or very close to being complete for all datasets (except MACERR: see below). Thus, we compared the results of the methods that do account for heterogeneity. It is noteworthy that the estimates of μ and σ^2 by both the LNBzt and the matrix-based methods fell within the range observed in our simulation study for all datasets other than MACERR.

All five methods considered that the SAVINGS and MANTEL datasets were complete after 34 and 76 participants had been included, respectively. However, the confidence intervals produced by the matrix-based and the LNBzt methods suggest that few problems had yet to be discovered.

The matrix-based and the LNBzt methods estimated similar number of problems for EDU3D ($\widehat{m}_{\text{matrix-based}} = 152$ and $\widehat{m}_{LNBzt} = 155$). The 95% confidence interval was broader for the LNBzt method (132 to 195) than for the matrix-based method (135 to 167).

The infusion pumps in the INFPUMP study were in early-stage development, and an additional re-design phase (for fixing the usability problems discovered) was planned; this explains why n= 107 unique problems were detected by the 34 participants in the usability testing. The LNBzt and matrix-based methods gave similar estimates and confidence intervals: $\widehat{m}_{\text{LNBzt}} = 122$ (i.e. 15 undiscovered problems), with a 95% confidence interval from 115 to 131, whereas $\widehat{m}_{\text{matrix-based}} = 120$, with a 95% confidence interval from 112 to 143. The parameters computed by the matrix-based method predicted an average probability of detection $\hat{\mu}_{\text{matrix-based}} = logit(0.136)$ and a dispersion

543 of $\hat{\sigma}_{\text{matrix-based}} = 1.52$. For the LNBzt method, the probability $\hat{\mu}_{\text{LNBzt}} = logit(0.136)$ was
544 the same, and the dispersion was slightly higher ($\hat{\sigma}_{\text{LNBzt}} = 1.50$). The confidence interval
545 (from 110 to 136) was narrower. The true number of problems with the pump was not
546 known because it was redesigned after 34 participants had tested the device. However, if
547 we accept the parameters $\hat{\mu}$ and σ as true and apply the results of our simulation study,
548 the INFPUMP data suggest that the LNBzt and matrix-based methods are both reliable.
549 Nevertheless, the breadth of the respective confidence intervals emphasizes the
550 remaining uncertainty for these two methods.

551 Using the MACERR data, the LNBzt predicted a very low average probability of detection
552 ($\hat{\mu}_{\text{LNBzt}} = logit(0.014)$) and a high level of heterogeneity ($\hat{\sigma}_{\text{LNBzt}} = 1.90$). These values
553 were out of the range of the settings tested in the simulation study, and suggested that the
554 number of "rare" problems was high. This might explain the high number of problems
555 predicted by the LNBzt method ($\hat{m}_{\text{LNBzt}} = 449$), and the very large 95% confidence
556 interval (from 256 to 1301). The matrix-based method's estimate was lower
557 ($\hat{m}_{Matrix-based} = 382$), and the 95% confidence interval was narrower (346 to 440).
558 However, the number of participants included in MACERR was low (n=15); a larger
559 number of participants would have been necessary to discover new problems and
560 improve the estimates.

561 On average, computation of the estimate and its confidence interval took less than ten
562 minutes for the matrix-based method, less than one minute for the LNBzt method, and
563 only a few seconds for the three other methods.

564 IV. Discussion

565 We decided to model the full discovery matrix (including unobserved columns) and not
566 just a summary of the data (e.g. the margins). The estimation problem was considered

simultaneously in terms of the (heterogeneous) probability of problem detection and the number of problems. Although the experimental conditions in real-life usability testing are unknown, the matrix-based method outperformed the other methods and appeared to be the most reliable in a broad range of settings.

Most of the currently available methods assume that the probability of detection is the same for all problems. This assumption is likely to be wrong, since real data show that the probability of detection varies [19, 23]. Furthermore, ignoring heterogeneity is known to strongly bias the results [24, 39]. We therefore developed a method that accounted for heterogeneity in the probability of problem discovery p; we used a logit-normal distribution as a plugin to model this uncertainty. The choice of this distribution was convenient in that it allowed us to compare our method with the only published model that accounts for heterogeneity. However, there are no data for confirming the validity of this choice. Nevertheless, this limitation could be easily overcome by replacing the logit-normal by another distribution (such as beta or gamma) if it proves to be more appropriate. This choice could be made using model choice criteria (e.g. the Akaike information criterion or the Bayesian information criterion). However, it should be borne in mind that for a small sample size, fitting for both incompleteness and heterogeneity is complex and inevitably leads to a high degree of uncertainty.

Here, we sampled μ and σ for fixed values of m. This turned out to be a rather time-consuming strategy because we had to run as many chains as there were values of m. We chose not to sample directly from the joint distribution $P(\mu, \sigma, m|\mathbb{d})$ because the dimension of the latent parameters $p_1, p_2, \dots, p_m$ varied as a function of m - making it impossible to use a standard MCMC algorithm. In this particular situation, use of the

reversible jump algorithm [40] might be a solution but would considerably complicate our algorithm.

There are two key moments in medical device development for assessing the best method. Early in the development cycle, the device is not mature; usability testing is referred to as "formative" because many usability problems are being discovered and corrected in an iterative design improvement process. Just before market access, usability testing is referred to as "validation" testing; they are performed on the final version of the device to ensure that no critical usability problems remain [1, 2].

The number of participants in the validation testing is an important parameter for both the regulatory authorities and the device manufacturer. Indeed, a sufficient sample size will (i) guarantee the medical device's compliance with the safety standards required for market authorization, and (ii) avoid a "black swan" effect that would strongly affect the manufacturer's credibility and profitability [41]. The validation testing focuses on the detection of infrequent usability problems. The US Food and Drug Administration requires a minimum of 15 participants [1]. This minimum is based on a naïve estimate, which has been proven to dramatically underestimate the true number of usability problems for this number of participants [12]. Indeed, the average coverage probability observed in our simulation study for $n = 20$ was as low as 12% and did not exceed 51%. Furthermore, this threshold does not consider heterogeneity in the probability of problem detection. Our findings suggest that to produce a relevant estimate with the matrix-based method, at least 20 participants are required in the validation step. In fact, the matrix-based method displayed good statistical properties with as few as 20 participants.

Since the validation testing only concerned problems that are probably less frequent, one could question the need to use methods that account for a heterogeneous probability of problem detection. In fact, problems are expected to be "homogeneously rare". To the best of our knowledge, however, the assumption of homogeneity for rare problems has no theoretical or experimental basis. Furthermore, human factor engineers will define the usability testing's experimental conditions according to the risk analysis, in order to facilitate the detection of problems previously described in the literature. If an engineer suspects the existence of problem removing the cap from an adrenaline pen, he/she might choose to evaluate the device in a more realistic test environment (e.g. with an actor pretending to go into anaphylactic shock); the problem is more likely to occur there than in a quiet, low-fidelity environment. By making some problems more detectable, the human factor engineer might introduce a degree of heterogeneity into the discovery process.

The choice of method was even more obvious for "formative" testing. In our simulations, the "formative" testing corresponds to a setting in which usability problems are frequent and numerous. Schmettow's usability testing of a medical infusion pump is also an example of a formative assessment because it was followed by a redesign. Here, we proved that matrix-based methods are more reliable and have low bias and high consistency. As in the case of the infusion pump, a reliable estimate from a small number of participants is an economic advantage for the manufacturer, who can shorten redesign cycles, accelerate device development, and hasten market access. The matrix-based method met this requirement because it required the fewest participants to guarantee good statistical properties. Another strength of the matrix-based method is its ability to embed previous knowledge through the prior parameters. Indeed, we used weakly informative priors for μ and σ to avoid introducing information that we did not have

about the medical device in question. However, one could take advantage of prior knowledge from earlier stages in device development or from a formative usability assessment to increase the accuracy of the estimate, especially when the sample size is small (i.e. an early control strategy). This approach is actually encouraged by regulatory bodies for medical device clinical trials [42] and helps to reduce the overall sample size.

Although we have suggested a threshold of 20 participants as the minimum sample size for obtaining a reliable estimate with the matrix-based method, we do not consider this to be the final threshold or a "magic number". Indeed, as suggested by various researchers, the estimation models should be run iteratively as the sample size increases [4]. Thus, estimation models constitute a means of controlling and ensuring quality in formative testing and should not solely be considered as a checkpoint for validation testing. Although the matrix-based method was more reliable, the LNBzt method could be used to double check the estimates - especially when high dispersion and/or the presence of very rare problems is suspected. Indeed, the LNBzt method's coverage probability is high, and the overestimation bias makes it a conservative method that could usefully prevent the usability testing from being stopped too early.

V. Conclusions

Estimation models (and particularly matrix-based models) are of value in estimating and monitoring the detection process during usability testing. Matrix-based models have a solid mathematical grounding and, with a view to facilitating the decision-making process for both regulators and device manufacturers, should be incorporated into current standards. To this end, the step-by-step tutorial provided here should facilitate the practical use of the matrix-based method in the evaluation of medical devices.

VI. Step-by-step instructions for applying the matrix-based method

A. Install R

R is a programming language and free software environment for statistical computing supported by the R Foundation for Statistical Computing. It can be installed via the CRAN http://cran.R-project.org.

B. Install the rstan and bridgesampling packages

Run the following code:

> *install.packages("rstan")*

> *install.packages("bridgesampling")*

C. Load the discovery matrix

Consider the d.csv file, a discovery matrix of 0 and 1 (saved in .csv format). It can be imported into R via the following command line instruction:

> *d <- read.csv("d.csv")*

D. Load the stan model

Load the rstan library and compile the stan model (the file *draw_mu_s2.stan* is provided in Additional file 3):

> *library(rstan)*

> *model <- stan_model("draw_mu_s2.stan")*

E. Run the matrix-based method

The matrix-based approach is implemented in the file *functions.R* (provided in Additional file 3). This file can be sourced using the following command line:

```
> source("functions.R")
```

The *heterogenous_bayes()* function runs the matrix-based method on the discovery matrix and returns the estimated m (M is the maximum number of problems on the m grid):

```
> res <- heterogeneous_bayes(d, M = 50, full_output = FALSE)
> print(res)
```

In order to obtain more details (e.g. the posterior distribution of m and sampled values of μ and σ^2 given m), the full_output argument must be set to TRUE:

```
> res_full <- heterogeneous_bayes(d, M = 50, full_output = TRUE)
> plot(res_full$posterior_m,type = "h", main = "Posterior distribution p(m|d)")
> head(res_full$simu_mu_s2)
```

A more detailed version of these instructions is provided as supplementary material (see Additional file 2).

VII. Abbreviations

LNBzt: logit normal binomial zero truncated

GT: Good-Turing

VIII. Declarations

Ethics approval and consent to participate: Not applicable.

Consent for publication: Not applicable.

Availability of data and materials: The data generated and analyzed during the current study are included in this published article and its supplementary information files.

Competing interests: The authors declare that they have no competing interests.

Funding: This research was funded by the Swiss National Science Foundation (grant number: SNSF-164279) and the French Agence Nationale de la Recherche (grant number: ANR-15-CE36-0007). The funding bodies had no role in the design of the study and collections, analysis, and interpretation of data, and in writing the manuscript.

Authors' contributions: AC, BD and VV conceptualized and designed the study. AC, CD, RP and VV carried out the analysis. AC, AD, BD, CD, RP, SP and VV contributed to the interpretation of the results. AC and VV drafted the initial manuscript. AD, BD, CD, RP and SP critically reviewed and revised the manuscript. All authors approved the final manuscript as submitted and have agreed both to be personally accountable for the author's own contributions and to ensure that questions related to the accuracy or integrity of any part of the work, even ones in which the author was not personally involved, are appropriately investigated, resolved, and the resolution documented in the literature.

Acknowledgements: We thank Cedric Bach, James Lewis, and Jakob Nielsen for making their datasets available. We would like to express our particular gratitude to Martin Schmettow for providing us with the full dataset from the infusion pump usability study. We also thank the HPC Computing Mésocentre of the University of Lille which provided us with the computing grid. Lastly, we thank Simone Borsci, James Lewis, and Martin Schmettow for their valuable comments on an early version of this manuscript, which helped us to significantly improve the content.

IX. References

1. US-FDA: **Applying human factors and usability engineering to medical devices: Guidance for industry and Food and Drug Administration staff**. *Washington, DC: FDA* 2016.

2. UK-MHRA: **Human Factors and Usability Engineering – Guidance for Medical Devices Including Drug-Device Combination Products**. In. Edited by Agency MHpR; 2017.
3. US-FDA: **Medical device recall report FY2003 to FY2012**. *Center for Devices and Radiological Health* 2012.
4. Borsci S, Macredie RD, Barnett J, Martin J, Kuljis J, Young T: **Reviewing and extending the five-user assumption: a grounded procedure for interaction evaluation**. *ACM Transactions on Computer-Human Interaction (TOCHI)* 2013, **20**(5):1-23.
5. Borsci S, Macredie RD, Martin JL, Young T: **How many testers are needed to assure the usability of medical devices?** *Expert Rev Med Devices* 2014, **11**(5):513-525.
6. Lewis JR: **Sample sizes for usability studies: Additional considerations**. *Human factors* 1994, **36**(2):368-378.
7. Kanis H: **Estimating the number of usability problems**. *Appl Ergon* 2011, **42**(2):337-347.
8. Lewis JR: **Evaluation of Procedures for Adjusting Problem-Discovery Rates Estimated From Small Samples**. *International Journal of Human-Computer Interaction* 2001, **13**(4):445-479.
9. Hertzum M, Jacobsen NE: **The Evaluator Effect: A Chilling Fact About Usability Evaluation Methods**. *International Journal of Human-Computer Interaction* 2003, **15**(1):183-204.
10. Schmettow M: **Sample size in usability studies**. *Communications of the ACM* 2012, **55**(4):64-70.
11. Borsci S, Londei A, Federici S: **The Bootstrap Discovery Behaviour (BDB): a new outlook on usability evaluation**. *Cogn Process* 2011, **12**(1):23-31.
12. Faulkner L: **Beyond the five-user assumption: Benefits of increased sample sizes in usability testing**. *Behavior Research Methods, Instruments, & Computers* 2003, **35**(3):379-383.
13. Lewis JR: **Using discounting methods to reduce overestimation of p in problem discovery usability studies**. In.: Citeseer; 2000.
14. Sauro J, Lewis JR: **Quantifying the user experience: Practical statistics for user research**: Morgan Kaufmann; 2016.
15. Thomas DG, Gart JJ: **Small sample performance of some estimators of the truncated binomial distribution**. *Journal of the American Statistical Association* 1971, **66**(333):169-177.
16. Virzi RA: **Refining the test phase of usability evaluation: How many subjects is enough?** *Human factors* 1992, **34**(4):457-468.
17. Nielsen J, Landauer TK: **A mathematical model of the finding of usability problems**. In: *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems: 1993*; 1993: 206-213.
18. Good IJ: **The population frequencies of species and the estimation of population parameters**. *Biometrika* 1953, **40**(3-4):237-264.
19. Schmettow M: **Controlling the usability evaluation process under varying defect visibility**. In: *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology: 2009*: British Computer Society; 2009: 188-197.
20. Finney D: **The truncated binomial distribution**. *Annals of Eugenics* 1947, **14**(1):319-328.

21. Rider PR: **Truncated binomial and negative binomial distributions**. *Journal of the American Statistical Association* 1955, **50**(271):877-883.
22. Shah S: **The asymptotic variances of method of moments estimates of the parameters of the truncated binomial and negative binomial distributions**. *Journal of the American Statistical Association* 1961, **56**(296):990-994.
23. Schmettow M: **Heterogeneity in the usability evaluation process**. *People and Computers XXII Culture, Creativity, Interaction 22* 2008:89-98.
24. Caulton DA: **Relaxing the homogeneity assumption in usability testing**. *Behaviour & Information Technology* 2001, **20**(1):1-7.
25. Schmettow M, Vos W, Schraagen JM: **With how many users should you test a medical infusion pump? Sampling strategies for usability tests on high-risk systems**. *J Biomed Inform* 2013, **46**(4):626-641.
26. DasGupta A, Rubin H: **Estimation of binomial parameters when both n, p are unknown**. *Journal of Statistical Planning and Inference* 2005, **130**(1-2):391-404.
27. Fisher RA: **The negative binomial distribution**. *Annals of Eugenics* 1941, **11**(1):182-187.
28. Haldane JB: **The fitting of binomial distributions**. *Annals of Eugenics* 1941, **11**(1):179-181.
29. Carroll RJ, Lombard F: **A note on N estimators for the binomial distribution**. *Journal of the American Statistical Association* 1985, **80**(390):423-426.
30. Olkin I, Petkau AJ, Zidek JV: **A comparison of n estimators for the binomial distribution**. *Journal of the American Statistical Association* 1981, **76**(375):637-642.
31. Hall P: **On the erratic behavior of estimators of N in the binomial N, p distribution**. *Journal of the American Statistical Association* 1994, **89**(425):344-352.
32. Robert C: **The Bayesian choice: from decision-theoretic foundations to computational implementation**: Springer Science & Business Media; 2007.
33. Meng X-L, Wong WH: **Simulating ratios of normalizing constants via a simple identity: a theoretical exploration**. *Statistica Sinica* 1996:831-860.
34. Bach C, Scapin DL: **Comparing inspections and user testing for the evaluation of virtual environments**. *Intl Journal of human–computer interaction* 2010, **26**(8):786-824.
35. Lewis JR, Henry SC, Mack RL: **Integrated office software benchmarks: A case study**. In: *Interact: 1990*; 1990: 337-343.
36. Nielsen J, Molich R: **Heuristic evaluation of user interfaces**. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: 1990*; 1990: 249-256.
37. Team SD: **RStan: the R Interface to Stan. R package version 2.17. 3**. In.; 2018.
38. Gronau QF, Singmann H, Wagenmakers E-J: **Bridgesampling: An R package for estimating normalizing constants**. *arXiv preprint arXiv:171008162* 2017.
39. Woolrych A, Cockton G: **Why and when five test users aren't enough**. In: *Proceedings of IHM-HCI 2001 conference: 2001*: Eds)(Cépaduès Editions, Toulouse, FR, 2001); 2001: 105-108.
40. Green PJ: **Reversible jump Markov chain Monte Carlo computation and Bayesian model determination**. *Biometrika* 1995, **82**(4):711-732.
41. Bias RG, Mayhew DJ: **Cost-justifying usability: An update for the Internet age**: Elsevier; 2005.
42. US-FDA: **Guidance for the use of Bayesian statistics in medical device clinical trials**. *Maryland: US Food and Drug Administration* 2010.

X. Figures

A. Figure 1

Title: Bias in the prediction of m: the mean error and 95% fluctuation interval (as a percentage of the true m) as a function of the sample size (n).

Legend: The results are presented for various probabilities of problem detection ((μ, σ), columns) and various numbers of usability problems (m, rows). The dashed line represents the true m.

B. Figure 2

Title: Consistency in the prediction of m: the RMSE for the prediction of m (as a percentage of the true m) as a function of the sample size (n).

Legend: The results are presented for various probabilities of problem detection ((μ, σ), columns) and various numbers of usability problems (m, rows). The LNBzt results are not represented for $m < 100$ and m$u = logit(0.1)$, due to a high RMSE.

XI. Supplementary Material

A. Additional file 1: R code

File name: Additional file 1.

File format: .zip

Title of data: reproducible R code with the simulation study performed in this manuscript.

Description of data: Unzip the folder, open the Rproject using Rstudio® (https://rstudio.com/) and execute the R code in "*simulation.R*".

The outputs are available in a .rds format in order to avoid large computation time.

B. Additional file 2: Short tutorial

File name: Additional file 2.

File format: .zip

Title of data: Step by step tutorial for the matrix-based method presented in this manuscript.

Description of data: Unzip the folder, open the file *"tutorial.pdf*" and follow the instructions.

C. Additional file 3

File name: Additional file 3.

File format: .zip

Title of data: R scripts.

Description of data: Unzip the folder to access the following R script with (i) the functions and (ii) the stan model.

D. Tabulated version of the data presented in this manuscript

S-Table 5: Bias in the prediction of m: the mean error (as a percentage of the true m) as a function of the sample size (n).

		μ	logit(0.1)														logit(0.2)													
		σ	0.5					1					2					0.5					1					2				
m	method	n	15	20	30	40	50	15	20	30	40	50	15	20	30	40	50	15	20	30	40	50	15	20	30	40	50	15	20	30	40	50
20	double deflation		21.7	20.7	19.9	19.7	19.8	17.7	17.7	18.0	18.5	18.8	13.8	14.4	15.4	16.2	16.6	20.2	19.8	19.9	19.9	20.0	18.4	18.7	19.2	19.5	19.7	15.8	16.5	17.3	17.8	18.2
	Good-Turing		19.8	19.7	19.5	19.6	19.7	16.9	17.3	17.9	18.4	18.8	13.8	14.4	15.4	16.2	16.6	19.7	19.7	19.9	19.9	20.0	18.2	18.7	19.2	19.5	19.7	15.7	16.5	17.3	17.8	18.2
	LNBzt		22.9	21.5	20.4	20.1	20.3	23.9	22.1	21.5	20.7	20.4	30.9	27.4	24.8	22.7	22.4	20.3	20.1	20.1	20.0	20.0	20.9	20.4	20.2	20.1	20.1	25.0	22.7	21.8	21.1	20.7
	matrix-based		22.9	21.4	20.2	19.8	19.8	20.1	19.7	19.6	19.5	19.5	18.3	18.7	19.0	19.0	19.2	20.2	19.8	19.8	19.9	20.0	19.5	19.5	19.5	19.6	19.7	18.8	19.1	19.2	19.3	19.3
	naive		17.4	18.3	19.1	19.5	19.7	15.6	16.6	17.7	18.3	18.8	13.5	14.2	15.4	16.2	16.6	19.3	19.5	19.9	19.9	20.0	17.9	18.6	19.2	19.5	19.7	15.7	16.4	17.3	17.8	18.2
50	double deflation		53.4	51.1	49.5	49.2	49.3	43.2	43.6	44.6	45.9	47.0	34.0	35.9	38.6	40.4	41.7	50.2	49.5	49.7	49.9	49.9	45.8	46.5	48.0	48.8	49.2	39.3	41.0	43.3	44.6	45.3
	Good-Turing		49.0	48.8	48.6	48.9	49.2	41.6	42.8	44.4	45.9	47.0	33.9	35.9	38.6	40.4	41.7	49.0	49.1	49.6	49.9	49.9	45.4	46.4	48.0	48.8	49.2	39.3	41.0	43.3	44.6	45.3
	LNBzt		52.5	51.9	50.2	50.1	50.0	53.1	52.2	50.9	50.4	50.5	58.7	55.8	53.6	54.0	55.6	50.3	50.1	50.0	50.0	50.0	50.7	50.3	50.1	50.1	50.1	53.2	52.0	51.1	51.1	50.6
	matrix-based		54.1	51.9	50.4	49.9	49.7	49.7	50.0	49.6	49.4	49.5	47.5	48.6	49.1	49.2	49.4	50.4	49.9	49.6	49.9	49.9	49.6	49.5	49.4	49.5	49.5	49.0	49.2	49.2	49.5	49.2
	naive		43.3	45.5	47.6	48.6	49.2	38.7	41.1	43.9	45.7	46.9	33.4	35.7	38.5	40.4	41.7	48.1	48.9	49.6	49.9	49.9	44.7	46.2	48.0	48.8	49.2	39.2	40.9	43.3	44.6	45.3
100	double deflation		105.5	102.2	98.9	98.2	98.6	85.9	86.6	89.0	91.7	93.7	68.0	71.9	77.0	80.5	83.3	100.0	99.0	99.3	99.7	99.9	91.3	93.1	96.0	97.5	98.4	78.6	81.9	86.3	88.9	90.9
	Good-Turing		97.4	97.5	97.2	97.6	98.4	82.8	85.1	88.6	91.6	93.7	67.8	71.8	77.0	80.5	83.3	97.8	98.2	99.3	99.7	99.9	90.4	92.8	96.0	97.5	98.4	78.5	81.9	86.3	88.9	90.9
	LNBzt		101.3	100.4	100.0	99.9	100.0	101.3	100.9	100.5	100.2	100.2	107.7	105.2	101.8	101.5	100.2	99.9	100.0	100.0	100.0	100.0	100.5	100.2	100.1	100.1	100.1	102.8	101.8	101.0	100.6	100.7
	matrix-based		104.2	102.5	100.7	99.9	99.7	98.0	99.2	99.5	99.5	99.5	95.1	97.4	98.0	98.7	98.6	100.4	99.9	99.6	99.7	99.9	99.5	99.4	99.5	99.5	99.5	98.3	98.9	99.2	99.3	99.6
	naive		86.4	91.1	95.3	97.1	98.3	77.2	82.1	87.7	91.4	93.6	66.9	71.5	76.9	80.5	83.3	96.0	97.7	99.2	99.7	99.9	89.3	92.5	96.0	97.5	98.4	78.3	81.9	86.3	88.9	90.9

S-Table 6: Consistency in the prediction of m: the RMSE for the prediction of m as a function of the sample size (n).

		μ	logit(0.1)														logit(0.2)													
		σ	0.5					1					2					0.5					1					2			
m	method	n	15	20	30	40	50	15	20	30	40	50	15	20	30	40	50	15	20	30	40	50	15	20	30	40	50	15	20	30	40	50
20	double deflation		4.3	2.8	1.5	1.0	0.7	3.8	3.3	2.6	2.0	1.6	6.6	6.0	5.0	4.2	3.8	1.4	0.9	0.4	0.3	0.1	2.3	1.8	1.2	0.8	0.6	4.6	3.9	3.1	2.6	2.2
	Good-Turing		3.1	2.3	1.5	1.0	0.7	4.1	3.5	2.6	2.0	1.6	6.6	6.0	5.0	4.2	3.8	1.3	0.9	0.4	0.3	0.1	2.4	1.8	1.2	0.8	0.6	4.6	3.9	3.1	2.6	2.2
	LNBzt		17.9	16.9	6.3	1.2	9.6	22.3	16.5	14.2	8.2	6.0	42.7	33.5	25.9	17.5	14.4	1.9	1.1	0.5	0.3	0.2	8.0	2.6	1.4	0.9	0.7	26.6	13.2	12.2	5.9	4.0
	matrix-based		6.2	3.9	1.8	1.1	0.8	5.8	4.5	3.1	2.2	1.7	7.2	6.6	5.8	4.5	4.4	1.7	1.1	0.4	0.3	0.2	3.1	2.0	1.3	0.9	0.7	5.3	4.4	3.7	3.1	2.4
	naive		3.4	2.5	1.5	1.0	0.7	4.9	3.9	2.8	2.1	1.6	6.9	6.1	5.0	4.2	3.8	1.3	0.9	0.4	0.3	0.1	2.5	1.9	1.2	0.8	0.6	4.7	3.9	3.1	2.6	2.2
50	double deflation		6.8	4.1	2.3	1.7	1.2	8.2	7.3	5.9	4.5	3.5	16.4	14.4	11.8	10.0	8.7	2.1	1.4	0.7	0.4	0.3	4.8	4.0	2.4	1.6	1.2	11.0	9.4	7.2	5.9	5.1
	Good-Turing		4.9	3.7	2.5	1.8	1.3	9.4	7.9	6.1	4.6	3.5	16.5	14.5	11.8	10.0	8.7	2.2	1.5	0.7	0.4	0.3	5.2	4.1	2.5	1.6	1.2	11.1	9.4	7.2	5.9	5.1
	LNBzt		18.3	19.0	5.9	4.5	1.2	19.5	14.6	7.6	4.7	4.9	38.9	28.9	22.7	27.7	37.2	2.6	1.6	0.7	0.4	0.3	4.8	3.2	1.9	1.4	1.1	15.6	11.7	8.0	7.5	6.3
	matrix-based		8.2	5.1	2.7	1.7	1.3	8.9	7.1	4.5	3.3	2.5	11.7	10.3	8.4	7.3	6.6	2.5	1.5	0.8	0.4	0.3	4.4	3.1	1.9	1.4	1.2	8.3	6.9	5.3	4.3	3.8
	naive		7.5	5.2	3.0	2.0	1.3	11.8	9.3	6.5	4.7	3.5	16.9	14.7	11.9	10.0	8.7	2.6	1.6	0.7	0.4	0.3	5.7	4.2	2.5	1.6	1.2	11.2	9.4	7.2	5.9	5.1
100	double deflation		9.8	6.3	3.4	2.7	1.9	15.6	14.3	11.6	8.8	6.8	32.3	28.5	23.4	19.9	17.1	3.0	2.1	1.1	0.6	0.3	9.4	7.4	4.4	2.9	2.0	21.8	18.5	14.1	11.6	9.5
	Good-Turing		7.1	5.7	4.1	3.1	2.1	18.2	15.6	11.9	8.9	6.8	32.6	28.5	23.4	19.9	17.1	3.6	2.5	1.2	0.6	0.3	10.2	7.7	4.5	2.9	2.0	21.9	18.5	14.1	11.6	9.5
	LNBzt		11.6	7.7	4.0	2.5	1.7	14.8	10.3	6.5	4.8	3.8	35.9	24.8	15.2	14.4	8.8	3.7	2.2	1.0	0.6	0.3	6.3	4.2	2.6	1.9	1.5	16.0	13.0	11.0	6.4	5.4
	matrix-based		9.8	7.2	3.8	2.4	1.7	11.5	9.2	6.3	4.8	3.8	14.8	13.2	10.8	9.6	8.3	3.5	2.2	1.2	0.6	0.3	6.0	4.2	2.7	2.0	1.6	11.0	9.2	7.3	6.1	5.3
	naive		14.4	9.8	5.4	3.5	2.2	23.3	18.4	12.8	9.1	6.9	33.4	28.9	23.5	19.9	17.1	4.8	2.9	1.2	0.6	0.3	11.2	8.0	4.5	2.9	2.0	22.1	18.5	14.1	11.6	9.5

S-Table 7: Coverage probability (in % of the 95% confidence interval) of $\hat{m}$ with each combination (m,μ,σ,n).

		μ	logit(0.1)														logit(0.2)													
		σ	0.5					1					2					0.5					1					2			
m	method	n	15	20	30	40	50	15	20	30	40	50	15	20	30	40	50	15	20	30	40	50	15	20	30	40	50	15	20	30	40	50
20	double deflation		85	84	91	85	74	87	80	48	28	20	16	6	1	1	2	90	87	56	36	90	68	42	20	43	69	5	2	4	8	16
	Good-Turing		96	93	84	67	62	69	59	33	20	18	7	3	1	1	2	86	69	39	74	97	44	29	26	54	72	2	1	5	9	16
	LNBzt		97	96	93	91	90	89	91	92	87	92	92	90	93	94	94	93	86	85	87	97	90	91	85	86	87	94	93	90	94	90
	matrix-based		87	91	97	99	100	95	96	96	97	97	97	96	96	96	95	97	100	98	97	91	96	97	98	97	96	96	95	95	95	96
	naive		47	52	50	50	57	12	13	11	11	15	0	0	1	1	2	51	53	33	77	98	13	17	26	57	72	0	2	5	9	16
50	double deflation		52	60	83	85	61	85	69	28	8	4	1	0	0	0	0	71	90	64	6	65	60	24	8	11	43	0	0	0	0	1
	Good-Turing		95	93	82	63	40	51	35	11	3	3	0	0	0	0	0	87	65	55	23	93	22	8	5	21	45	0	0	0	0	1
	LNBzt		97	93	90	92	86	94	92	94	92	96	92	95	95	91	96	94	93	82	86	84	92	94	88	94	86	94	98	95	96	92
	matrix-based		90	94	96	98	98	97	95	96	95	96	97	97	95	95	94	96	98	97	94	93	95	96	96	96	96	95	95	94	95	95
	naive		18	28	36	31	37	0	1	1	1	2	0	0	0	0	0	43	32	49	26	93	2	2	3	24	45	0	0	0	0	1
100	double deflation		21	32	72	83	63	83	59	10	2	0	0	0	0	0	0	46	85	46	16	21	45	10	1	0	18	0	0	0	0	0
	Good-Turing		94	92	81	50	26	31	14	1	0	0	0	0	0	0	0	86	59	43	3	84	7	1	1	2	20	0	0	0	0	0
	LNBzt		97	96	90	88	86	92	94	94	94	93	95	97	95	93	95	93	92	89	79	89	94	92	92	93	93	93	96	96	95	94
	matrix-based		72	81	90	93	96	82	85	88	91	92	89	87	88	88	89	91	94	96	93	91	89	93	94	95	96	85	87	89	91	90
	naive		2	10	22	22	13	0	0	0	0	0	0	0	0	0	0	28	30	43	2	86	0	0	1	5	20	0	0	0	0	0