ROYAL SOCIETY
OPEN SCIENCE

rsos.royalsocietypublishing.org

Research

Article submitted to journal

Subject Areas:
Statistics, Computational Methods

Keywords:
Multinomial Counts, Exact Probability, Urns

Author for correspondence:
Pasquale Cirillo
e-mail: P.Cirillo@tudelft.nl

Computing the Exact Distributions of Some Functions of the Ordered Multinomial Counts: Maximum, Minimum, Range and Sums of Order Statistics

Marco Bonetti[1], Pasquale Cirillo[2] and Anton Ogay[3]

[1]Unit of Population Dynamics and Health - Dondena Research Center, Institute for Data Science and Analytics, Department of Social and Political Sciences, Bocconi University, Via G. Roentgen 1, 20136 Milan (MI), Italy.
[2]Applied Probability Group - Delft University of Technology, Van Mourik Broekmanweg 6, 2628 XE Delft, The Netherlands.

Starting from seminal neglected work by Rappeport (1968), we revisit and expand on the exact algorithms to compute the distribution of the maximum, the minimum, the range, and the sum of the J largest order statistics of a multinomial random vector under the hypothesis of equiprobability. Our exact results can be useful in all those situations in which the multinomial distribution plays an important role, from goodness-of-fit tests to the study of Poisson processes, with applications spanning from biostatistics to finance. We describe the algorithms, motivate their use in statistical testing and illustrate two applications. We also provide the codes and ready-to-use tables of critical values.

1. Introduction

The multinomial random vector arises naturally in several statistical problems, from queuing theory to software reliability models, from clinical trials to financial mathematics, from goodness-of-fit tests to transportation problems [1,2]. The multinomial experiment is a common way of representing the multinomial random

THE ROYAL SOCIETY
PUBLISHING

vector as the result of throwing n independent balls into m independent urns, each with a given probability of attraction p_i, and of counting the number n_i of balls that fall into urn i, for $i = 1, ..., m$ [3]. The probability mass function (pmf) of the resulting vector of counts $(N_1, \ldots, N_m)^T$ is

$$P(N_1 = n_1, \ldots, N_m = n_m; \mathbf{p}, m, n) = \frac{n!}{n_1! \cdots n_m!} \prod_{i=1}^{m} p_i^{n_i},$$

where $\mathbf{p} = (p_1, \ldots, p_m)^T$ and $\sum_{k=1}^{m} n_k = n$.

A common statistical hypothesis of interest is whether the underlying multinomial distribution is equiprobable, so that the probability for a ball of falling in any of the urns is always the same, i.e. $H_0 : P = P_0$ vs. $H_1 : P \neq P_0$, where P_0 is the equiprobable pmf

$$P_0(N_1 = n_1, \ldots, N_m = n_m; \mathbf{p} = (p, \ldots, p)^T, m, n) = \frac{n!}{n_1! n_2! \ldots n_m!} p^n, \tag{1.1}$$

with $p = \frac{1}{m}$. From now on we indicate the pmf in Equation (1.1) as $\mathrm{Mult}\,(n, \mathbf{p_0})$, where $\mathbf{p_0} = (m^{-1}, \ldots, m^{-1})^T$.

Many procedures have been proposed to test the equiprobability hypothesis, a good review being [2]. The classical way is to use the χ^2 goodness-of-fit test, first introduced by Pearson [4] and based on the statistic

$$X^2 = \sum_{i=1}^{m} \frac{(N_i - np)^2}{np}.$$

Other approaches replace X^2 with the Neyman modified X^2 [5], with the log-likelihood ratio statistic

$$G^2 = 2 \sum_{i=1}^{m} N_i \ln\left(\frac{N_i}{np}\right),$$

or with the Freeman-Tukey statistic [6].

In 1962, Young [7] revisited this problem and proposed two alternatives based on the rescaled range

$$W_m = \left(\max_{1 \leq k \leq m} \frac{N_k}{n} - \min_{1 \leq l \leq m} \frac{N_l}{n} \right) \tag{1.2}$$

and the rescaled mean

$$M_m = (mn)^{-\frac{1}{2}} \sum_{i=1}^{m} \left| N_i - \frac{n}{m} \right|.$$

These new statistics revealed some power advantages under certain alternatives [7].

In testing the equiprobability hypothesis, all the statistics above rely on approximations (like the Normal, the χ^2, the Beta, the Dirichlet, or the Gumbel), being their exact distributions not known. This requires that the original data satisfy some often heuristic conditions: for instance, the χ^2 approximation for the Pearson statistic is typically recommended when $n \geq 5m$ [2,6]. In several applications, especially when dealing with small samples, these conditions are rarely satisfied and as a consequence the tests may be unreliable.

More recently, Corrado [8] has offered a solution, based on what he calls *stochastic matrices*, for computing the exact probabilities of the multinomial maximum, minimum, and range. Corrado's approach clearly solves the problem of using potentially inaccurate approximations but–as we shall see–it represents an unsatisfactory solution, since it requires ad-hoc computations for each combination of n and m. Our aim is thus to propose a more general and flexible approach.

Our investigation originates from an old PhD thesis by Rappeport [9], in which two algorithms for the exact computation of the distributions of the multinomial maximum and of the sum of the 3 largest multinomial order statistics were proposed. We first describe Rappeport's results, and then we present novel general algorithms for computing the exact distributions of the multinomial minimum, of the range and of the sum of the J largest order statistics. This means that, for example, the distribution of the test statistic W_n in Equation (1.2) can now be obtained exactly.

The article develops as follows. Section 2 provides a quick overview of the existing distributional approximations for the multinomial range and other order statistics, whereas Section 3 contains the exact results of Corrado [8]. Section 4 is devoted to the original algorithms by Rappeport, while Sections 5 and 6 introduce our results for the minimum and the range respectively. Section 7 discusses some non-trivial accuracy issues of the commonly used approximations. In Section 8 we provide some additional motivation for the use of the (sums of) the largest multinomial counts in hypothesis testing. In Section 9 we shortly describe some of the many possible applications that involve these exact results. We close with a discussion in Section 10. The Appendix contains tables of critical values for the multinomial maximum, minimum, range and sums, as well as codes for all algorithms.

2. Approximations

Starting from some results of Pearson & Hartley [10], a first approximation for the multinomial distribution under equiprobability was introduced by Johnson & Young [11]. Young [7] used this approximation to derive a limiting distribution for the range of the multinomial sample.

The limiting distribution of the maximum, using a Gumbel approximation, was initially introduced by Kolchin in 1978 [12], with some errors that were later corrected in DasGupta [13]. No general result appears to be available for the multinomial minimum or other order statistics. Some specific cases, easily computable by hand, are described in [2].

All the approximations presented below are derived under the hypothesis of equiprobability.

(a) Approximation of the Distribution of the Range

It is well-known that, marginally, for $i = 1, \dots, m$ one has $N_i \sim \mathrm{Binom}(n, p_i)$, so that under the null hypothesis $E(N_i) = np_i = np$ and $\mathrm{Var}(N_i) = np(1-p)$ with $p = 1/m$. Using the multidimensional central limit theorem, the joint distribution of the standardized multinomial vector $(\omega_1, \dots, \omega_m)^T$, with

$$\omega_i = \frac{N_i - np}{\sqrt{np(1-p)}} = \frac{mN_i - n}{\sqrt{n(m-1)}},$$

converges in distribution, as $n \to \infty$, to a multivariate normal distribution with zero mean vector, unit variances, and covariance between ω_i and ω_j equal to $(1-m)^{-1}$ for $i \neq j$. Note that the limiting distribution is actually degenerate, i.e. its support is $(m-1)$-dimensional due to the n-sum constraint that applies to the N_i terms (or, equivalently, the zero-sum constraint on the ω_i terms). One has a non-degenerate limiting distribution for any choice of a set of $m-1$ terms from the set $(\omega_1, \dots, \omega_m)$.

The distribution of the range of m i.i.d standard normal variables $X_1, X_2, \dots, X_m$ is a known quantity, and it can be computed as

$$P\left(\max_{1\le i\le m} X_i - \min_{1\le j\le m} X_j \le r\right) = m\int_{-\infty}^{\infty} \phi(x)\left(\int_x^{x+r} \phi(u)\mathrm{d}u\right)^{m-1} \mathrm{d}x,$$

where $\phi(x)$ is the probability density function of a standard normal random variable. Using this, Young [7] shows that the distribution of the scaled multinomial range can be approximated as

$$P\left(\max_{1\le i\le m} \frac{N_i}{n} - \min_{1\le j\le m} \frac{N_j}{n} \le r\right) \approx P\left(\max_{1\le i\le m} X_i - \min_{1\le j\le m} X_j \le (r+\delta_m)\sqrt{nm}\right),$$

where δ_m is a continuity factor such that

$$\delta_m = \frac{1}{n} \text{ for } m = 2, \text{ and } \delta_m = \frac{1}{2n} \text{ for } m > 2.$$

The approximation works best for large values of the ratio n/m (our simulations suggest $n > 5m$). We refer to [2,7] for additional details.

(b) Approximation of the Distribution of the Maximum

The approximating distribution for the maximum of a multinomial sample was proposed by Kolchin [12] and improved by DasGupta [13], to which we refer for all technical details.

Set

$$\mu = \frac{n}{m} \text{ and } \kappa = \frac{\log m - \frac{1}{2}\log\log m}{\mu},$$

and let ϵ be the unique positive root of the equation

$$(1+\epsilon)\log(1+\epsilon) - \epsilon = \kappa.$$

Then the law of the maximum of a multinomial sample converges in distribution to a Gumbel random variable, i.e.

$$P\left(\frac{\max\limits_{1\le i\le m} N_i - \mu(1+\epsilon)}{\sqrt{\frac{n}{2m\log m}}} + \frac{\log(4\pi)}{2} \le z\right) \xrightarrow{d} e^{-e^{-z}},$$

for all real z, as $n \to \infty$. As observed in [2,8], the approximation by Kolchin and Dasgupta is the best one available for the multinomial maximum in the literature so far.

We now turn to the exact results.

3. Exact Results by Corrado (2011)

Corrado's approach [8] is based on a matrix representation for the construction of the transition probabilities for the number of balls in the different urns. The main advantage of Corrado's method is that it does not require equiprobability.

Let N_k be the random number of balls in urn k. The sequence $S_k = S_{k-1} + N_k$ describes the cumulative ball count from $S_0 = 0$ to $S_m = n$, where m is the total number of urns. The transition probability from S_{k-1} to S_k is equal to

$$P(S_k = s_k \mid S_{k-1} = s_{k-1}; p_k^*) = \\ \begin{cases} \binom{n-s_k}{s_k - s_{k-1}} (p_k^*)^{s_k - s_{k-1}} (1-p_k^*)^{n-s_k} & s_k \ge s_k - 1 \\ 0 & \text{otherwise} \end{cases}, \tag{3.1}$$

where $p_k^* = \frac{p_k}{\sum_{j=k}^m p_j}$ (so that in particular $p_1^* = p_1$). For brevity, in what follows we write $P(s_k \mid s_{k-1}; p_k^*)$.

From Equation (3.1), for $k = 1, \dots, m$, we can determine upper-triangular matrices of the form

$$Q_k = \begin{bmatrix} P(0 \mid 0; p_k^*) & P(1 \mid 0; p_k^*) & \dots & P(n \mid 0; p_k^*) \\ 0 & P(1 \mid 1; p_k^*) & \dots & P(n \mid 1; p_k^*) \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}.$$

These matrices provide a straightforward way to calculate the desired exact probabilities of the multinomial order statistics. The first transition from $s_0 = 0$ to s_1 travels across a starting row vector Q_1^1 defined as

$$Q_1^1 = [P(0 \mid 0; p_1) \;\; P(1 \mid 0; p_1) \;\; \dots \;\; P(n \mid 0; p_1)].$$

Q_1^1 indicates the first row of matrix Q_1. The product $Q_1^1 \times Q_2$ is a row vector whose elements give the convolution distribution of $S_2 = N_1 + N_2$, while $Q_1^1 \times Q_2 \times \cdots Q_k$ represents the convolution distribution of the random sum S_k. Naturally $Q_1^1 \times Q_2 \times \cdots Q_m = 1$ concentrates all the mass on $S_m = n$, so that Q_m is equal to a column vector of ones.

In order to calculate the exact probability for the maximum amount of balls in the urns not to exceed a given r, i.e. $P\left(\max\limits_{1\le i\le m} N_i \le r\right)$, Corrado suggests the following procedure:

(i) In all the matrices Q_k, $k = 1, ...m$, set to 0 all the transition probabilities $P(s_k \mid s_{k-1}; p_k^*)$ for which $s_k - s_{k-1} > r$. This defines a new series of sub-matrices, $Q_1^*, ..., Q_m^*$, called *culled*.

(ii) The product of the sub-matrices $Q_1^*, ..., Q_m^*$ gives the exact probability of $P\left(\max_{1\leq i\leq m} N_i \leq r\right)$.

A simple example taken from [8] will clarify the method. Imagine to throw 3 balls across 3 urns, so that $n = m = 3$. Then we can easily verify that $Q_1^1 = [0.296\ 0.444\ 0.222\ 0.037]$, $Q_3 = [1\ 1\ 1\ 1]$ and

$$Q_2 = \begin{bmatrix} 0.125 & 0.375 & 0.375 & 0.125 \\ 0 & 0.25 & 0.5 & 0.25 \\ 0 & 0 & 0.5 & 0.25 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Now, suppose that one is interested in computing the probability that the multinomial maximum is smaller than or equal to 2 . The matrices above need to be modified as $Q_1^{*1} = [0.296\ 0.444\ 0.222\ 0]$, $Q_3^* = [0\ 1\ 1\ 1]$ and

$$Q_2^* = \begin{bmatrix} 0.125 & 0.375 & 0.375 & 0 \\ 0 & 0.25 & 0.5 & 0.25 \\ 0 & 0 & 0.5 & 0.25 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Then $P\left(\max_{1\leq i\leq m} N_i \leq 2\right) = Q_1^{*1} \times Q_2^* \times Q_3^* = 0.889.$

The distribution of the minimum can be obtained similarly, if one modifies each matrix Q_k, by setting $P(s_k|s_{k-1}; p_k^*)$ equal to 0 for $s_k - s_{k-1} < r$ (please notice the change in the inequality sign). Additional details are available in [8].

Interestingly, the distribution of the multinomial range can also be computed using the matrix representation. Set $Q_k(a_k, b_k)$ to be the culled matrix for urn k, where $P(s_k \mid s_{k-1}; p_k^*) = 0$ for all $N_k > a_k$ or $N_k < b_k$. Introducing the set of all possible allocations of n balls across m urns as $\cap_{k=1}^m \{a_k \geq N_k \geq b_k\}$, one can express the joint probability of the maximum and the minimum ball counts as

$$P\left(\cap_{k=1}^m a_k \geq N_k \geq b_k\right) = Q_1^{*1} \times \prod_{k=2}^{m-1} Q_k(a_k, b_k) \times Q_m^*. \tag{3.2}$$

Note that the set of allocations described above have intersecting intervals, so that to compute the exact probabilities for the range the intersection probabilities should be subtracted:

$$P\left(\max_{1\leq k\leq m} N_k - \min_{1\leq l\leq m} N_l < r\right) =$$
$$= \sum_{h=0}^{n-r+1} Q_1^{*1} \times \prod_{k=2}^{m} Q_k(h+r-1, h) \times Q_m^* - \sum_{h=0}^{n-r} Q_1^{*1} \times \prod_{k=1}^{m} Q_k(h+r-1, h+1) \times Q_m^*.$$

Corrado's method works nicely and allows for the exact probability computations of the multinomial maximum, minimum and range. However, it has a strong limitation: for every new composition, and for every value of r, the culled matrices have to be redesigned and recalculated, something not very efficient.

4. Rappeport's Algorithms

In 1968, Rappeport [9] proposed two iterative algorithms for the distribution of the multinomial maximum, and for the sum of the 3 largest multinomial order statistics. That work remained unpublished, and Rapperport's idea has been mostly ignored. These algorithms are based on the

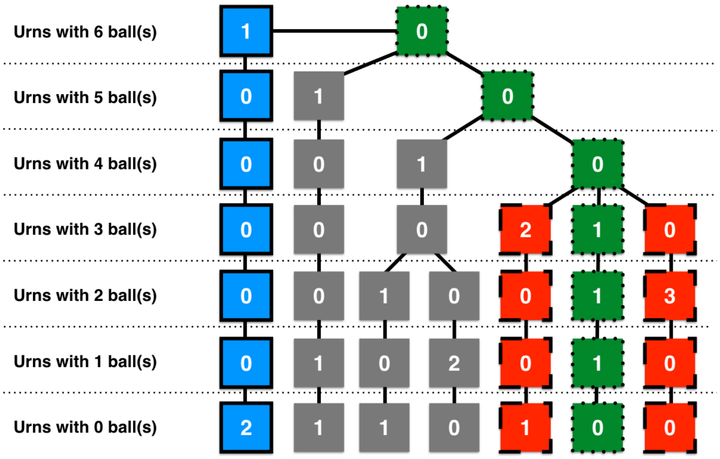

Figure 1: Tree representation of the possible outcomes of the multinomial experiment of throwing 6 balls into three equiprobable urns.

representation of all the possible outcomes of the multinomial experiment in the form of a tree. The desired probabilities are computed by moving across the branches of this tree according to certain rules. A relevant feature of Rappeport's approach is the possibility of deriving a general algorithm, which does not require adjustments that depend on the specific parameters and values of interest.

Consider 6 balls thrown across 3 equiprobable urns. The possible outcomes of this multinomial experiment can be represented with the tree in Figure 1, where each branch corresponds to a particular partition (up to rearrangement) of the balls into the urns. For instance, the blue path on the left, characterized by the squares with bold edges, represents the situation in which one of the urns contains all the 6 balls, while the other two are empty. Conversely, the green path with the dotted squares represents the configuration where one urn contains three balls, one urn contains two balls, and the remaining urn contains only one ball. And so on for all the other branches.

Rappeport also discusses the case of urns with different attraction probabilities, under the assumption that they can be collected into two or three same-probability groups. In these non-equiprobable situations, Rappeport's approach loses its generality, and it necessarily requires case-specific adjustments that do depend on the characteristics of the groups of urns. For this reason, and since the main null hypothesis in multinomial statistical tests is that of equiprobability, we do not discuss the details of the two- and three-group cases.

(a) Distribution of the Maximum

Let $N_{<1>}, ..., N_{<m>}$ be the order statistics of the multinomial counts $N_1, ..., N_m$ in descending order, so that $N_{<1>}$ is the maximum, $N_{<2>}$ is the second largest order statistics, and so on, up to the minimum $N_{<m>}$. Note that, just like the counts, the ordered counts clearly need not to be all different.

In the tree representation, to compute the exact probabilities $P(N_{<1>} \leq r; n, m)$ of the multinomial maximum under equiprobability, one sums the probabilities of all the paths, whose nodes are characterized by zeros for all the levels from $r+1$ to n. For example, suppose that one wants to compute the probability that no urn contains more than $r = 3$ balls, when $n = 6$ and $m = 3$. Then she needs to sum the probabilities of the green dotted and the red dashed paths in Figure 1, because in all the other situations (gray and blue) $N_{<1>} \geq 4$.

To perform such computations, Rappeport develops an efficient iterative procedure, in which the probability $P(N_{<1>} \leq r; n, m)$ is explicitly obtained from $P(N_{<1>} \leq r-1; n, m)$ and some initial value.

Let us compute the probability that the maximum number of balls in the m urns is exactly equal to r, and let us assume that such a maximum is unique, i.e. all the others urns have at most r-1 balls. Using Equation (1.1), and introducing the operator W_2, which is the sum over all possible values $n_{<2>}, ..., n_{<m>}$ such that $n_{<1>} > n_{<2>}$, we can write

$$P\left(N_{<1>} = r \mid N_{<1>} > N_{<2>}; n, m\right) = \frac{1}{r!} W_2 \left(\frac{n!}{m^n \prod_{i=2}^{m} n_{<i>}!} \frac{m!}{\prod_{k=0}^{r-1}((\#n_i = k)!)} \right).$$

Forcing notation, here $(\#n_i = k)$ denotes the number of n_i's equal to k, that is to say the number of urns containing k balls. The fraction $\frac{m!}{\prod_{k=0}^{r-1}(\#n_i=k)!}$ arises from a simple combinatorial argument: since the urns are equiprobable, the probability of a given composition of $n_1, ..., n_m$ should be multiplied by the total number of its unique permutations.

Let us now relax the assumption of uniqueness for the maximum, to obtain

$$\begin{aligned} &P\left(N_{<1>} = r \mid N_{<1>} = N_{<q>} > N_{<q+1>}; n, m\right) = \\ &= \frac{1}{(r!)^q \, q!} W_{q+1} \left(\frac{n!}{m^n \prod_{i=q+1}^{m} n_{<i>}!} \frac{m!}{\prod_{k=0}^{r-1}((\#n_i = k)!)} \right). \end{aligned}$$

Let $q = 0$ indicate the case of $n_{<1>} < r$. Summing over q then yields

$$\begin{aligned} P\left(N_{<1>} \leq r; n, m\right) &= \sum_q \frac{1}{(r!)^q \, q!} \frac{n! \, m!}{m^n} \\ &\times \quad W_{q+1} \left(\frac{1}{\prod_{i=q+1}^{m} (n_{<i>}!) \prod_{k=0}^{r-1}((\#n_i = k)!)} \right). \end{aligned} \tag{4.1}$$

The number of urns containing exactly r balls cannot be greater than $\lfloor \frac{n}{r} \rfloor$, which therefore defines the upper limit of the summation. Moreover, since all the other $n - rq$ balls should be placed in the remaining $m - q$ urns, with maximum not exceeding $r - 1$, one must have $(m - q)(r - 1) \geq (n - rq)$, therefore the lower limit for q is $\max(0, n - rm + m)$. The range of the summation in Equation (4.1) thus becomes

$$\max(0, n - rm + m) \leq q \leq \lfloor \frac{n}{r} \rfloor. \tag{4.2}$$

Noting that

$$W_{q+1} \left(\frac{1}{\prod_{i=q+1}^{m} (n_{<i>}!) \prod_{k=0}^{r-1}((\#n_i = k)!)} \frac{(m-q)! \, (n-rq)!}{(m-q)^{n-rq}} \right) = P\left(N_{<1>} \leq r - 1; n - rq, m - q\right),$$

the following iterative formula for the probability of the multinomial maximum holds:

$$P(N_{<1>} \leq r; n, m) = \sum_q A_q P(N_{<1>} \leq r - 1; n - rq, m - q), \tag{4.3}$$

where

$$A_q = \frac{n! \, m!}{m^n} \frac{1}{(r!)^q \, q!} \frac{(m-q)^{n-rq}}{(m-q)! \, (n-rq)!}.$$

The starting point of the iteration is represented by the probability that the maximum is smaller than or equal to 1 (where the former clearly cannot occur for $n, m > 0$), i.e.

$$P(N_{<1>} \leq 1; n, m) = P(N_{<1>} = 1; n, m) \quad = \begin{cases} \frac{m!}{m^n (m-n)!} & \text{if } m \geq n \\ 0 & m < n \end{cases}.$$

This quantity can be easily derived from Equation (1.1), since the only possible configurations corresponding to $\{N_{<1>} = 1\}$ are those with n frequencies equal to 1 and $n - m$ frequencies equal to 0, and there are $\binom{m}{n}$ such sequences.

Table 3 contains critical values for the maximum as obtained with this algorithm, for different combinations of n and m. As expected, these exact numbers coincide with those obtainable using Corrado's stochastic matrices [8] (and the same will be true for the minimum and the range).

(b) Distribution of the Sum of the J Largest Order Statistics

The algorithm for the maximum can also be used for the calculation of the exact distribution of the sum of the J largest order statistics. Rappeport discusses explicitly the cases $J = 2, 3$, and only gives some hints about the general case $3 < J < m$.

Consider the case $J = 3$ as in the original work by Rappeport[1]. To compute $P(N_{<1>} + N_{<2>} + N_{<3>} \leq r; n, m)$ one may partition this probability into different terms, corresponding to the different possible ranges of $N_{<1>}$ and $N_{<2>}$. Indeed, one can distinguish among three disjoint cases:

(i) $N_{<1>} \leq \frac{r}{3}$. In this case

$$P\left(\sum_{i=1}^{3} N_{<i>} \leq r; n, m\right) = P\left(N_{<1>} \leq \frac{r}{3}; n, m\right).$$

(ii) $N_{<1>} > \frac{r}{3}$ and $N_{<2>} \leq \frac{r - N_{<1>}}{2}$. Here one can fix a value of $N_{<1>} = t_1$, thus forcing one urn to contain exactly t_1 balls. The remaining urns define a smaller sample with $n^* = n - t_1$ and $m^* = m - 1$. If the maximum of this new sample is smaller or equal than $\frac{r-t_1}{2}$, the original inequality for the sum of the three largest order statistics will automatically hold. The total probability in this case is equal to the sum over all possible values of t_1, i.e.

$$P\left(\sum_{i=1}^{3} N_{<i>} \leq r; n, m\right) = \sum_{t_1 = \lfloor \frac{r}{3} + 1 \rfloor}^{r} A_{t_1} P\left(N_{<1>} \leq \frac{r - t_1}{2}; n - t_1, m - 1\right),$$

with

$$A_{t_1} = \frac{n!\, m!}{m^n} \frac{1}{t_1!} \frac{(m-1)^{n-t_1}}{(m-1)!\,(n-t_1)!}.$$

(iii) $N_{<1>} > \frac{r}{3}$ and $N_{<2>} > \frac{r - N_{<1>}}{2}$. One may proceed as in the previous case, but now both values of $N_{<1>}$ and $N_{<2>}$ must be fixed, so that

$$P\left(\sum_{i=1}^{3} N_{<i>} \leq r; n, m\right) = \sum_{t_1 = \lfloor \frac{r}{3} + 1 \rfloor}^{r-1} \sum_{t_2 = \lfloor \frac{r-t_1}{2} + 1 \rfloor}^{\min(t_1, r - t_1)} A_{t_1,t_2} B_{t_1,t_2} \times P(N_{<1>} \leq r - t_1 - t_2; n - t_1 - t_2, m - 2),$$

where

$$A_{t_1,t_2} = \frac{n!\, m!}{m^n} \frac{(m-2)^{n-t_1-t_2}}{(m-2)!\,(n-t_1-t_2)!} \frac{1}{t_1!\, t_2!},$$

and $B_{t_1,t_2} = 0.5$ if $t_1 = t_2$, and it is equal to 1 otherwise. The term B_{t_1,t_2} accounts for the possibility of t_1 and t_2 being equal.

Collecting the probabilities from points 1, 2 and 3 above yields the desired probability.

The more general distribution of the sum of the $J < m$ largest order statistics can be computed similarly to the $J = 3$ case, by splitting the probability $P\left(\sum_{i=1}^{J} N_{<i>} \leq r; N, m\right)$ into J terms, each one dealing with some combinations of values for the first $J - 1$ order statistics.

[1] The original formulation of Rappeport contained a few errors, here corrected.

The general explicit formula for the distribution of the sum of the J largest order statistics is

$$P\left(\sum_{i=1}^{J} N_{<i>} \leq r; n, m\right) = \tag{4.4}$$

$$P\left(N_{<1>} \leq \frac{r}{J}; n, m\right) + \sum_{t_1 = \lfloor \frac{r}{J}+1 \rfloor}^{r} A_{t_1} P\left(N_{<1>} \leq \frac{r - t_1}{J-1}; n - t_1, m-1\right) +$$

$$\cdots + \sum_{t_1} \cdots \sum_{t_{J-1}} A_{t_1,\ldots,t_{J-1}} B_{t_1,\ldots,t_{J-1}} \times P\left(N_{<1>} \leq r - \sum_{i=1}^{J-1} t_i; n - \sum_{i=1}^{J-1} t_i, m - J + 1\right).$$

If we denote with I the total number of summations for a particular range of values, then the relative summation limits are defined as

$$\begin{cases} \lfloor \frac{r}{J} + 1 \rfloor \leq t_1 \leq r - I + 1 & \text{if } i = 1 \\ \lfloor \frac{r - \sum_{i=1}^{I-1} t_i}{J - I + 1} + 1 \rfloor \leq t_i \leq \min\left(t_{I-1}, r - \sum_{i=1}^{I-1} t_i\right) & \text{if } 2 \leq i \leq I \end{cases}.$$

The coefficients A and B are calculated according to the formulas

$$A_{t_1,\ldots,t_I} = \frac{n!\,m!}{m^n} \frac{1}{\prod_{i=1}^{I}(n_{<i>})!} \frac{(m-I)^{(n - \sum_{i=1}^{I} t_i)}}{(m-I)!(n - \sum_{i=1}^{I} t_i)!}, \tag{4.5}$$

and

$$B_{t_1,\ldots,t_I} = \frac{1}{\prod_{k=t_1}^{t_I} (\#t_i = k)!},$$

where, again, $(\#t_i = k)$ denotes the number of t_i's equal to k.

In the Appendix we report MATLAB code to compute the distribution of the sum of the $J < m$ largest order statistics. In Tables 4 and 5 we provide critical values for $J = 2$ and $J = 3$. Importantly, note that the algorithm for the sum can also be used immediately to compute the exact probabilities of the second, the third and all the other order statistics.

5. The exact distribution of the multinomial minimum

The distribution of the smallest order statistic $N_{<m>}$ can be easily derived, by using the probability of the sum of the $m-1$ largest order statistics, given that $P(N_{<m>} \geq r; n, m) = P\left(\sum_{i=1}^{m-1} N_{<i>} \leq n - r; n, m\right)$. However, it is not difficult to see that this approach turns out to be very computationally inefficient already for quite small values of n and m.

A new, efficient algorithm for the multinomial minimum can be constructed starting from that of the multinomial maximum. One needs to slightly modify the way in which we move across the branches of the tree. We start by assigning probability 0 to all the branches of the tree that contain urns with less than r balls. By iterating through all the possible values of the maximum, from r to n, we then compute

$$P(N_{<m>} \geq r) = \sum_{t=r}^{n-1} P(N_{<1>} \leq t; n, m), \tag{5.1}$$

with the conditions $P(N_{<1>} \leq r - 1; n, m) = 1$ if $n = m = 0$, and $P(N_{<1>} \leq r - 1; n, m) = 0$ if $n \neq 0$ or $m \neq 0$. Then, by computing $1 - P(N_{<m>} \geq r + 1)$, one can easily obtain the probability mass function of the multinomial minimum.

The MATLAB code is once again available in the Appendix, and Table 6 provides some useful critical values.

6. The Exact Distribution of the Multinomial Range

We now introduce a new iterative algorithm for computing the exact distribution of the multinomial range. Unlike the solution proposed by Corrado [8], our approach does not require any modification of the algorithm for every new urn composition.

Back to our earlier example with $n = 6$ balls and $m = 3$ urns, consider the probability that the range is smaller than or equal to 3. Such a probability can be split into two terms corresponding to the different ranges of variation of the maximum. That is,

$$P(N_{<1>} - N_{<3>} \leq 3) = P(N_{<1>} - N_{<3>} \leq 3 \,|\, N_{<1>} \leq 3) + P(N_{<1>} - N_{<3>} \leq 3 \,|\, N_{<1>} > 3).$$

The first term can be computed using Equation (4.3), since $P(N_{<1>} - N_{<3>} \leq 3 \,|\, N_{<1>} \leq 3) = P(N_{<1>} \leq 3)$.

To compute the second term one may use a procedure similar to the one for the minimum: to iterate through all the possible values for the maximum, while assigning zero probability to all the branches that have urns with less balls than the current maximum, i.e. 3 balls in our example. For instance, we compute $P(N_{<1>} \leq 4 \,|\, N_{<1>} > 3)$ with the additional conditions that $P(N_{<1>} < 1; n, m) = 1$, if $n = m = 0$, and $P(N_{<1>} < 1; n, m) = 0$, when $n \neq 0$ or $m \neq 0$, so to avoid meaningless paths.

The general algorithm is therefore as follows:

$$\begin{aligned} P(N_{<1>} - N_{<m>} \leq r; n, m) &= P(N_{<1>} \leq r; n, m) \\ &+ P(N_{<1>} - N_{<m>} \leq r \,|\, N_{<1>} > r; n, m). \end{aligned} \tag{6.1}$$

The first term is easily computed using the algorithm for the maximum, so we focus our attention on the second term. Assume, for the time being, that $N_{<1>} = N_{<q>} = r + 1 > N_{<q+1>}$. Then, one can rewrite the second term on the right-hand side of Equation (6.1) as

$$\begin{aligned} & P(N_{<1>} - N_{<m>} \leq r | N_{<1>} = N_{<q>} = r + 1 > N_{<q+1>}; n, m) = \\ = & \frac{F_{q+1,1}\left(\frac{n!\,m!}{m^n \prod_{i=q+1}^{m}(n_{<i>}!) \prod_{k=t-r}^{r}((\#n_i = m)!)}\right)}{((r+1)!)^q \, q!}, \end{aligned}$$

where $F_{q+1,1}$ is the operator that sums over all possible values of $n_{<q+1>}, \ldots, n_{<m>}$ such that the minimum $n_{<m>} \geq 1$, and $n_{<1>} = n_{<q>} > n_{<q+1>}$. Summing over q thus gives the total probability

$$\begin{aligned} & P(N_{<1>} - N_{<m>} \leq r | N_{<1>} = r + 1; n, m) = \\ & = \sum_q \frac{n!\,m!}{m^n((r+1)!)^q q!} \times F_{q+1,1}\left(\frac{1}{\prod_{i=q+1}^{m}(n_{<i>}!) \prod_{k=t-r}^{r}((\#n_i = m)!)}\right). \end{aligned} \tag{6.2}$$

In this case, $N_{<1>} > r$, and the term corresponding to $q = 0$ should be excluded, so that

$$\max\,(1, n - (r+1)m + m) \leq q \leq \lfloor \frac{n}{r+1} \rfloor.$$

Now, multiplying and dividing Equation (6.2) by $[(m - q)^{(n-(r+1)q)}][(m - q)!\,(Nn(r + 1)q)!]^{-1}$ yields

$$\begin{aligned} P(N_{<1>} - N_{<m>} \leq r \,|\, N_{<1>} = r + 1; n, m) = \sum_q \frac{n!\,m!}{m^n((r+1)!)^q q!} \frac{(m-q)^{(n-(r+1)q)}}{(m-q)!\,(N-(r+1)q)!} \times \\ \times F_{q+1,1}\left(\frac{(m-q)!(n-(r+1)q)!}{(m-q)^{(n-(r+1)q)}} \frac{1}{\prod_{i=q+1}^{m} n_{<i>}! \prod_{k=t-r}^{r}(\#n_i = k)!}\right). \end{aligned} \tag{6.3}$$

Note that the term

$$F_{q+1,1}\left(\frac{1}{\prod_{i=q+1}^{m} n_{<i>}!\prod_{k=t-r}^{r}(\#n_i=k)!}\frac{(m-q)!(n-(r+1)q)!}{(m-q)^{(n-(r+1)q)}}\right)$$

is an alternative way of writing

$$P\left(N_{<m>}\geq 1\,|\,N_{<1>}\leq r; n-(r+1)q, m-q\right).$$

The condition $N_{<1>}\leq r$ can be imposed to the algorithm by changing the threshold for the maximum. In other words,

$$P\left(N_{<m>}\geq 1|N_{<1>}\leq r; n-(r+1)q, m-q\right)=P(N_{<1>}\leq r; n, m),$$

with, similarly to what we have seen before, $P(N_{<1>}\leq 0; n, m)=1$, when $n=m=0$, and $P(N_{<1>}\leq 0; n, m)=0$, if $n\neq 0$ or $m\neq 0$.

Plugging this into Equation (6.3) produces

$$P(N_{<1>}-N_{<m>}\leq r|N_{<1>}=r+1; n, m)=$$

$$\sum_{q}\frac{1}{((r+1)!)^q q!}\frac{n!\,m!}{m^n}\quad=\frac{(m-q)^{(n-(r+1)q)}}{(m-q)!(n-(r+1)q)!}\times P(N_{<1>}\leq r; n-(r+1)q, m-q),$$

with the same conditions on $P(N_{<1>}\leq 0; n, m)$.

Recalling Equation (4.3) for the distribution of the maximum, we can finally calculate the second term on the rhs of Equation (6.1), by summing over all the possible values of $N_{<1>}$. Thus

$$P(N_{<1>}-N_{<m>}\leq r|N_{<1>}>r; n, m)=\sum_{t=r+1}^{n}P(N_{<1>}\leq t\,|\,N_{<1>}>t-1; n, m),$$

where $P(N_{<1>}\leq t-r-1; n, m)=1$, if $n=m=0$, and $P(N_{<1>}\leq t-r-1; n, m)=0$, if $n\neq 0$ or $m\neq 0$.

The condition $N_{<1>}>t-1$ is introduced at every summation step in order to avoid multiple calculations for the same branches of the outcome tree. This results in different ranges of summation in the recursion, i.e.

$$\max\left(1, n-tm+m\right)\leq q\leq\lfloor\frac{n}{t}\rfloor \text{ if } N_{<1>}>r$$

and

$$\max\left(0, n-tm+m\right)\leq q\leq\lfloor\frac{n}{t}\rfloor \text{ if } N_{<1>}\leq r.$$

Summarizing, the distribution of the range can be evaluated via the iteration step

$$P(N_{<1>}-N_{<m>}\leq r; n, m)=P(N_{<1>}\leq r; n, m)+\sum_{t=r+1}^{n}P(N_{<1>}\leq t\,|\,N_{<1>}>t-1; n, m),\quad(6.4)$$

with the same conditions as above for $P(N_{<1>}\leq t-r-1; n, m)$, when $N_{<1>}>r$.

As for the other exact algorithms, in Table 7 we provide a selection of critical values, and the related code is in the Appendix.

7. Approximations vs. Exact Results

One may want to compare the exact probabilities computed with the new algorithms with the approximations seen in Section 2. More generally, the operating characteristics of the exact vs. the approximate tests can be explored. Here we mainly focus our attention on the multinomial range, given that its use has been proposed in the literature for several statistical tests [2].

In goodness-of-fit tests, under the name "urn tests" [3], i.i.d. data generated from a statistical distribution and properly binned can be described as the result of a multinomial experiment.

Using the approximations given in Section 2, Young showed that a goodness-of-fit test based on the multinomial range has power advantages with respect to classical alternatives such the

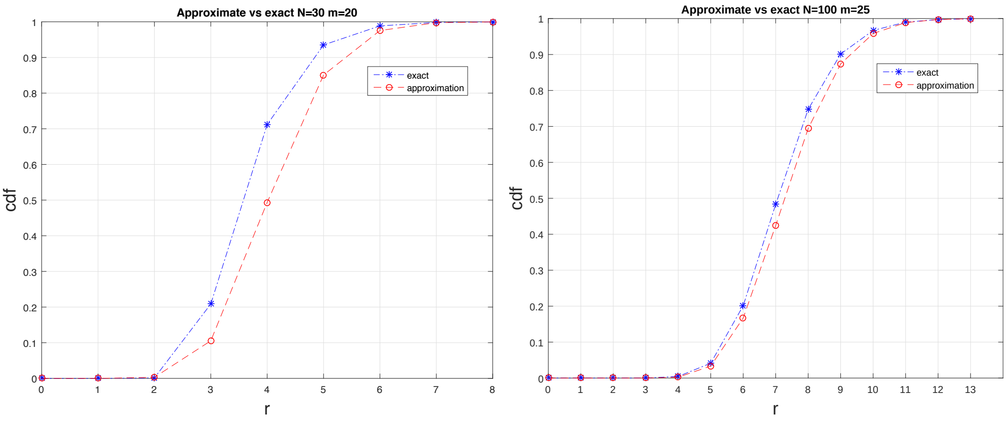

Figure 2: Comparison between the exact probabilities for the range and the approximation by Young [7] for $n = 30$ and $m = 20$ (left), and for $n = 100$ and $m = 25$ (right).

χ^2 [7]. For a more general comparison between tests based on the multinomial order statistics and the classical χ^2 goodness-of-fit test statistic, we refer to [14] and [9].

Consider a sample of i.i.d. observations $X_1, \dots, X_n$ from a distribution F on the real line, with interest in testing some null hypothesis $\tilde{H}_0 : F = F_0$.

The support of the distribution hypothetical F_0 can be partitioned into m *equiprobable* non-overlapping sub-intervals $B_1, \dots, B_m$ (for the optimal choice of the number of intervals we refer to [7,15]). We then define the variables $N_1, \dots, N_m$ as the absolute frequencies of the actual observations in the sample that fall in the intervals $B_1, \dots, B_m$, that is $N_i = \sum_{j=1}^{n} I(X_j \in B_i)$ for $1 \leq i \leq m$.

By construction of the bins, under $\tilde{H}_0$, the random vector of counts $(N_1, \dots, N_m)^T$ follows the $Mult(n, \mathbf{p})$ distribution with $\boldsymbol{p} = \boldsymbol{p_0} = \left(\frac{1}{m}, \dots, \frac{1}{m}\right)^T$. Note that $\tilde{H}_0$ is then transformed into the multinomial null hypothesis H_0. Hence we can assess the goodness-of-fit to F_0 by considering the transformed hypothesis testing problem: $H_0 : \boldsymbol{p} = \boldsymbol{p_0}$ vs. $H_1 : \boldsymbol{p} \neq \boldsymbol{p_0}$.

To test the multinomial equiprobability hypothesis, we can use the multinomial range as test statistic, as suggested in [7]. In what follows we compare the performances of our exact results, with those of the approximation discussed in Section 2.

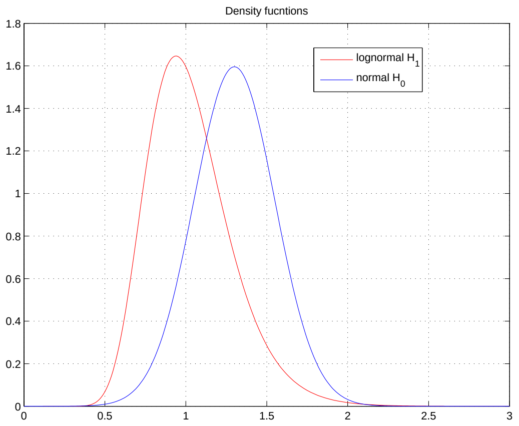

Figure 3: Densities of a Normal(1.3,0.25) and of a Lognormal(0,0.25).

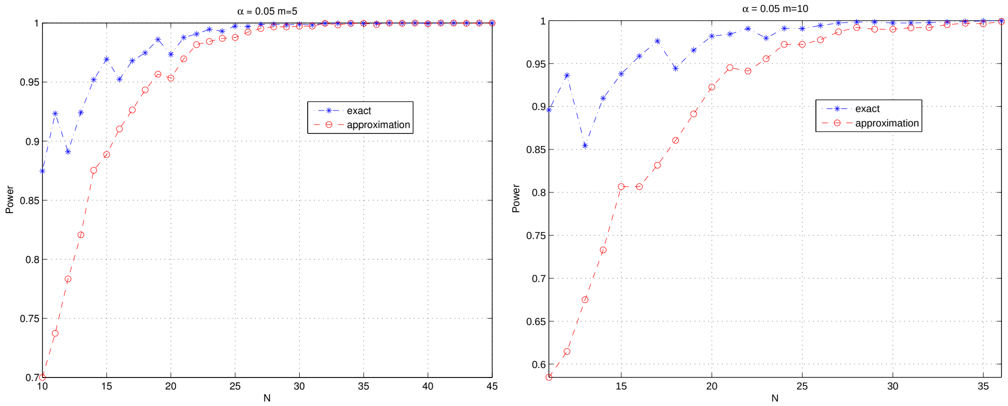

Figure 4: Power comparison of the goodness-of-fit tests when using 5 equiprobable bins (left) or 10 equiprobable bins (right).

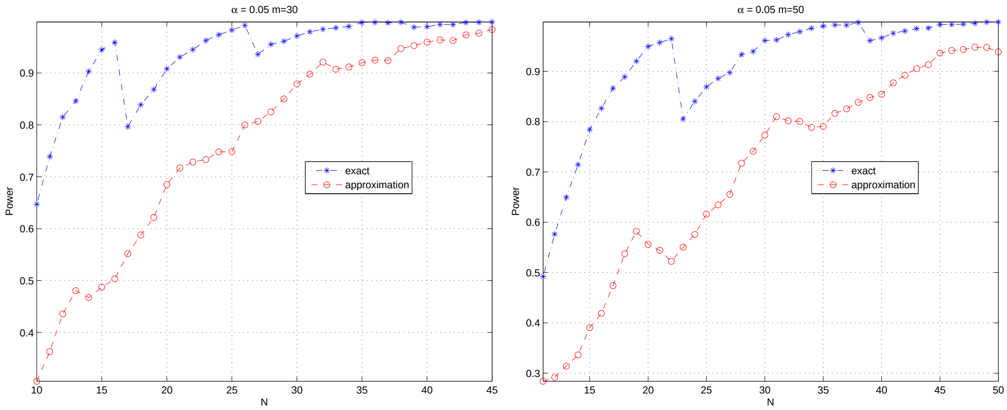

Figure 5: Power comparison of the goodness-of-fit tests when using 30 equiprobable bins (left) or 50 equiprobable bins (right).

Figure 2 shows two comparisons between the exact and the approximate cumulative distribution functions of the multinomial range, in the multinomial experiment, for $n = 30$ and $m = 20$, and for $n = 100$ and $m = 25$. As expected, when the number of balls is small with respect to the number of urns, the approximation is rather poor. Good results are only obtainable for $n \geq 5m$. This is in line with the findings in [7].

In our experiments, given the discrete nature of the test statistic, we constructed randomized testing procedures to ensure that the desired level of significance α (0.05) could indeed be achieved exactly in all cases [16].

As the null distribution F_0 we used a normal distribution $N(\mu, \sigma^2)$ with $\mu = 1.3$ and $\sigma = 0.25$, while as the alternative distribution F_1 we considered a lognormal distribution $LN(\mu, \sigma)$ with $\mu = 0$ and $\sigma_{LN} = 0.25$. The two densities are shown in Figure 3.

The power of the exact and of the approximate test procedures were computed via Monte Carlo estimation: for each combination of number of bins (m), and number of observations (n), we generated 3000 i.i.d. samples.

Figure 4 shows the power advantage of the test based on the exact distribution, for all test sample sizes, and for the alternative hypothesis we have considered here. The figure tells that, when using $m = 5$ or 10 urns, the exact test outperforms the approximate one for sample sizes smaller than 25, and it is at least as powerful as the approximate test in the other cases.

Figure 5 compares the powers of the tests for $m = 30$ and $m = 50$, respectively. The figure shows that the exact-distribution-based test is more powerful than the approximate one, for all values of n up until about 50, when all powers approach one.

While this small simulation exercise clearly has no ambition of being exhaustive, on the basis of the findings above, we can say that–as one would expect–the exact test appears to perform better than the approximate one, over a wide range of sample sizes, and for different numbers of urns. Moreover, the improvement in power seems to increase with the number of urns, and it can be very large for the smaller sample sizes.

8. Uniformly Most Powerful Tests and Sums of Multinomial Counts

We now discuss two constructions of testing procedures for the multinomial probabilities that motivate the use of the largest and of the sum of the J largest multinomial counts.

We consider *one* sample $\mathbf{N} = (N_1, \dots, N_m)^T = (n_1, \dots, n_m)^T = \mathbf{n}$, where $\sum_{j=1}^{m} n_j = n$, from the Multinomial$(n, \mathbf{p})$ distribution with $\mathbf{p} = (p_1, \dots, p_m)^T$. We focus on the null hypothesis $H_0 : p_1 = p_2 = \dots = p_m = 1/m$, or $H_0 : \mathbf{p} = \mathbf{p_0}$ in the notation of Section 1.

In this section we use the notation $L(\mathbf{p}; \mathbf{n}, m, n)$ for the likelihood function corresponding to a sample of size one from the multinomial distribution (i.e. its pmf).

Similar arguments to those that follow, with the appropriate changes, can be exploited to motivate the use of the minimum (or the sum of the J smallest) counts.

(a) Uniformly Most Powerful Test for an increase in one probability

First, note that if we restrict all the p_j for $j \neq i$ to be equal (and still $\sum_{j=1}^{m} p_j = 1$), then for a given value of p_i one necessarily has $p_j = (1 - p_i)/(m - 1)$, for all $j \neq i$. Now, consider testing the null hypothesis H_0 of equiprobabilty against the alternative hypothesis that corresponds to an increased probability of attraction for the i-th urn, with all other probabilities being equal:

$$H_{1i} : p_i = p_+ > \frac{1}{m}, \; p_j = \frac{1 - p_+}{m - 1} \; \forall j \neq i,$$

where $p_+ \in (1/m, 1)$. The Most Powerful (MP) level α test ($\alpha \in (0, 1)$) for the problem H_0 vs. H_{1i} can be easily obtained by direct application of the Neyman-Pearson's lemma (see, e.g., [17]). The rejection region of the MP test is defined by

$$\begin{aligned} \frac{\mathrm{L}\left(\mathbf{p} = \left(\frac{1}{m}, \dots, \frac{1}{m}\right); \mathbf{n}, m, n\right)}{\mathrm{L}\left(p_+; \mathbf{n}, m, n\right)} &= \frac{\frac{n!}{n_1! \cdots n_m!} \left(\frac{1}{m}\right)^n}{\frac{n!}{n_1! \cdots n_m!} p_+^{n_i} \left(\frac{1 - p_+}{m - 1}\right)^{n - n_i}} \\ &= \left[\frac{m - 1}{m(1 - p_+)}\right]^n \left[\frac{1 - p_+}{p_+(m - 1)}\right]^{n_i} \end{aligned}$$

being less than or equal to some constant such that the probability of rejection under the null be equal to α (note the slightly inconsistent – but convenient – notation for $L\left(p_+; \mathbf{n}, m, n\right)$ vs. $L\left(\mathbf{p}; \mathbf{n}, m, n\right)$). Since $p_+ > 1/m$, it follows immediately that $(1 - p_+) < (m - 1)p_+$, so that the rejection region can be written equivalently as $N_i \geq k_\alpha$. Given the discrete nature of the multinomial random vector, the rejection region must be augmented by construction of a randomization procedure to allow the test to have size α.

Since this rejection region does not depend on the choice of p_+, the MP test is actually *Uniformly* Most Powerful (UMP) for the wider testing problem

$$H_0 : p_1 = \dots = p_m = \frac{1}{m} \text{ vs. } H_{1i} : p_i > \frac{1}{m}, \; p_j = \frac{1 - p_i}{m - 1} \; \forall j \neq i.$$

Notice that this same rejection region emerges as the UMP test if one considers just the i-th marginal count N_i, which is distributed as a Binomial(n, p_i) random variable, and the alternative testing problem $H^*_{0i} : p_i = \frac{1}{m}$ vs. $H^*_{1i} : p_i > \frac{1}{m}$.

Let us now combine the collection of such tests obtained by letting $i = 1, \ldots, m$. Within the parameter space $\Theta = \{p_i \in (0, 1), i = 1, \ldots, m : \sum_{i=1}^{m} p_i = 1\}$, the null hypothesis H_0 can be written equivalently as $H_0 : \mathbf{p} \in \Theta_0 = \{1/m, \ldots, 1/m\}$. Trivially, $\Theta_0 = \cap_{i=1}^{m} \Theta_0$, so we may construct a global union-intersection test that rejects H_0 whenever at least one of the m tests rejects (see, e.g., [18]). Easily, such test would then have a rejection region of the form

$$\cup_{i=1}^{m} \{N_i \geq k^*\} = \{\max(N_1, \ldots, N_m) \geq k^*\}, \tag{8.1}$$

where k^* is common to the m tests due to the symmetric nature of the individual tests over $i = 1, \ldots, m$ under the null. To control the overall type I error probability α, the test should therefore reject if and only if $\max(N_1, \ldots, N_m) \geq k^*$, where the constant k^* should satisfy the size requirement that $\mathrm{P}(\max(N_1, \ldots, N_m) \geq k^*; H_0) = \alpha$, and it should therefore be obtained from the distribution of the test statistic $N_{<1>} = \max(N_1, \ldots, N_m)$ under H_0. Please observe that, in practice, the m individual tests (and the global test) are randomized, so that 8.1 holds only approximately. Nevertheless, the construction provides quite a strong motivation for the use of the test based on the largest observed count.

(b) Uniformly Most Powerful Test for an equal increase in two probabilities

Let us now consider the case of an equal increase of *two* of the $m \geq 3$ attraction probabilities, corresponding to the two urns i and j, $i \neq j$. The level α UMP test for the problem

$$H_0 \text{ vs. } H_{1ij} : p_i = p_j = p_+ > 1/m, \; p_h = \frac{1 - 2p_+}{m - 2} \; \forall h \neq i, j$$

can also be obtained from the Neyman-Pearson's lemma. Note that $p_+ < 0.5$ must hold.

The likelihood function[2] corresponding to the two probabilities p_i and p_j being the same, with all the others being equal, is

$$L(p_+; \mathbf{n}, m, n) \quad = \quad \frac{n!}{n_1! \cdots n_m!} p_+^{n_i} p_+^{n_j} \left(\frac{1 - 2p_+}{m - 2} \right)^{n - (n_i + n_j)} .$$

And the MP test rejects if and only if the likelihood ratio

$$\frac{L\left(\frac{1}{m}; \mathbf{n}, m, n)\right)}{L\left(p_+; \mathbf{n}, m, n)\right)} \quad = \quad \frac{\frac{n!}{n_1! \cdots n_m!} \left(\frac{1}{m}\right)^n}{\frac{n!}{n_1! \cdots n_m!} p_+^{n_i + n_j} \left(\frac{1 - 2p_+}{m - 2}\right)^{n - (n_i + n_j)}}$$

is less than or equal to some constant. It is easy to see that since $p_+ > (1 - 2p_+)/(m - 2)$, the rejection region can be written equivalently as $N_i + N_j \geq k$ (again with randomization adjustment). Since the test does not depend on the value of p_+ as long as it is greater than $1/m$ (and smaller than 0.5), the same test is UMP level α for the testing problem H_0 vs. $H_{ij} : p_i = p_j > 1/m, \; p_h = (1 - 2p_i)/(m - 2) \; \forall h \neq i, j$.

For this case one should note that the test that we have derived would *not* be UMP if we allowed p_i and p_j to take different values (both greater than $1/m$, with sum less than one) under the alternative hypothesis $\tilde{H}_{ij}$. Indeed, simple calculations show that, in such a case, the rejection region of the MP test for H_0 vs. $\tilde{H}_{ij} : p_i = \tilde{p}_i, p_j = \tilde{p}_j, \; p_h = (1 - \tilde{p}_i - \tilde{p}_j)/(m - 2) \; \forall h \neq i, j$ would look like $\gamma_i^{N_i} \gamma_j^{N_j} > k$, where $\gamma_i = \tilde{p}_i(m - 2)/(1 - \tilde{p}_i - \tilde{p}_j) > 1$, and similarly for γ_j. The (randomized) MP level α test would therefore require the rejection threshold k to be obtained from the distribution, under H_0, of the test statistic $\gamma_i^{N_i} + \gamma_j^{N_j}$, or equivalently of the test statistic $N_i \log(\gamma_i) + N_j \log(\gamma_j)$. As a consequence, the rejection threshold would depend on the specific values $\tilde{p}_i$ and $\tilde{p}_j$, and the resulting test could therefore *not* be UMP for the wider testing problem H_0 vs. $H^*_{ij} : p_i > 1/m, p_j > 1/m, \; p_h = (1 - p_i - p_j)/(m - 2) \; \forall h \neq i, j$.

[2] For ease of notation, we still call it $L(p_+; \mathbf{n}, m, n)$.

On the other hand, if we restrict ourselves to the case $p_i = p_j$, then we can again generalize this UMP level α test to reject against the alternative hypothesis that, for *some* pair (i, j) of probabilities, there has been an equal increase from $1/m$. Here, too, the $\binom{m}{2}$ UMP level α tests can be combined–up to the mentioned approximation due to the randomized nature of the tests–by rejecting H_0 whenever at least one among them does. The resulting rejection region is then

$$\cup_{i \neq j}\{N_i + N_j \geq k\} = \{N_{<1>} + N_{<2>} \geq k\},$$

where the equality of the two rejection regions can be easily verified. Again, the constant k that would ensure the overall type I error probability α should be obtained from the distribution of the sum of the two largest observed multinomial counts $N_{<1>} + N_{<2>}$.

The discussion in this subsection can be easily extended to the case of an equal increase in more than two of the m probabilities.

9. Two Illustrations

(a) Test for the Homogeneity of a Poisson Process

Suppose $M(t)$ is a homogeneous Poisson process on the line. If we partition its time domain into m non-overlapping equal-length sub-intervals $B_1, \ldots, B_m$, then, conditionally on the total number of events observed n, the numbers of events N_i in the intervals B_i, $i = 1, ..., m$, follow an equiprobable multinomial distribution [2]. Thus, a test for the homogeneity of the Poisson process is readily constructed from the absolute frequencies.

We simulate a non-homogeneous Poisson process (NHPP) and check whether the multinomial range test is able to identify the non-homogeneity. We use the time-scale transformation of a homogeneous Poisson process (HPP) with (constant) rate equal to one to generate the desired NHPP.

Intensity function	Parameters	Exact test	Approximate test
$\lambda(t) = 2 + 0.01 * t$	$m = 30$ and $T = 20$	0.184	0.037
$\lambda(t) = 0.3 * t$	$m = 20$ and $T = 200$	0.427	0.371
$\lambda(t) = 0.05 * t$	$m = 15$ and $T = 20$	0.133	0.059
$\lambda(t) = 2 + \sin(2\pi t)$	$m = 30$ and $T = 200$	0.213	0.084

Table 1: Power of the exact and approximate test in case of Poisson processes with intensity function $\lambda(t)$.

The interarrival times T of a HPP with rate one are known to be exponentially distributed with intensity one, i.e. $P(T \geq t) = \exp(-t)$. The interarrival times T' for a NHPP are such that $P(T' \geq t) = \exp(-\Lambda(t))$, where $\Lambda(x)$ is the integrated rate function of the process, i.e. the expected number of points in the interval $(0, x]$, with $\Lambda(0) = 0$ (this is also called the cumulative hazard function). It is immediate to see that if $T_1, \ldots, T_n$ are a sample of interarrival times generated from the HPP with rate one, then the transformed times $T'_i = \Lambda^{-1}(T_i)$, $i = 1, \ldots, n$, are a sample from a NHPP with integrated rate function $\Lambda(t)$. Hence we can simulate a NHPP by simply sampling exponential variables with parameter equal to one, and by taking the inverse Λ^{-1} of the generated interarrival times.

Four examples of power comparison (based on 1000 simulated samples each) between the exact and the asymptotic test are shown in Table 1, where three linear assumptions and a sin-shaped one for the NHPP intensity function are considered, for different numbers of bins (m) and maximum overall time length (T). Type I error probability was set equal to 0.05. One can easily appreciate the evident advantage of the exact test over the asymptotic one.

Let us focus a bit more on the the effect of choosing different numbers of disjoint intervals, when testing for the homogeneity of a Poisson Process. Assume that the true intensity function is described by the harmonic function $\lambda(t) = 2 + \sin(5\pi t)$. Figure 6 shows the power of the two tests when the time domain ($T = 200$) is split into 10 or 20 disjoint intervals. The blue and the red lines represent the case with 10 intervals, while the black and the green the one with 20. We can observe once again the better power performances of the exact test. The huge power gain between the two cases with 10 and 20 bins can be explained by the following fact: if we split the time domain into 20 non-overlapping, equal-length intervals, each of them covers the period of the intensity function, averaging out the effect of the harmonic function.

Therefore, the exact range-based test shows better performances with respect to the approximate one, when dealing with Poisson processes, but the specific performance may be quite dependent on the width (i.e. the number) of the bins.

(b) An Application to Disease Clustering

We now discuss a simple application of the multinomial range test to the problem of disease clusters' detection, something frequently of interest to epidemiologists and biostatisticians [19,20].

Often, disease clustering is initially approached as a hypothesis testing problem. The main goal is to test a null hypothesis of no clustering, i.e. a common rate of disease across the study region, against an alternative hypothesis of presence of clusters, or more generally of deviations from the null spatial distribution.

Numerous ways to construct such tests were proposed over the years, and one of the possible approaches is based on the multinomial distribution. In this case, the study region is divided into (roughly) equal population subregions. Then, under the null hypothesis of a common rate of disease across the wider region, and conditionally on the total number of cases observed in the region, the number of events (cases) observed in the different subregions follows an equiprobable multinomial distribution.

We briefly illustrate this approach by using a well-known epidemiological data set of diagnosed leukaemia cases over 8 counties in upstate New York [20]. This data originated from the New York State Cancer Registry, and were gathered during the 5-year period 1978-1982, with a total of 584 individuals diagnosed with leukaemia over a population of approximately 1 million people. The original data contain spatial information about registered events split into 790 census tracts, which however have different population sizes. Their spatial distribution is shown in Figure 7 (left map).

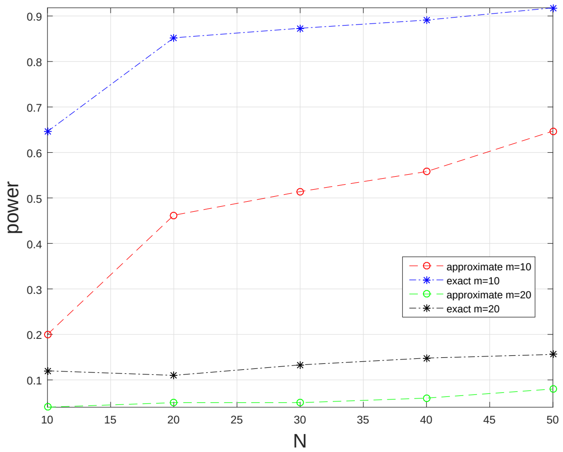

Figure 6: Power of the tests in case of harmonic intensity function $\lambda(t) = 2 + \sin(5\pi t)$ for the NHPP.

In order to perform the multinomial tests based on the range, we have to group the data points into subregions of approximately equal population. Ideally, the population of these subregions should be exactly the same, but that is impossible due to the original grouping in census tracts. To our knowledge, there is no existing unique algorithm to create spatial partitions such that the elements of the partition have equal population. We propose the following (admitteldly ad-hoc) procedure:

(i) Define the number of the new subregions to be constructed.
(ii) Use that number to compute a desired value for the population of each subregion.
(iii) Create subregions, roughly satisfying the desired population size, around locations with high initial population. Specifically, use the k-means algorithm to create subregions for the rest of the observations, initializing the subregions with locations that have the highest populations.
(iv) Trade observations between subregions based on the population and the distance, until the population constraints are satisfied.

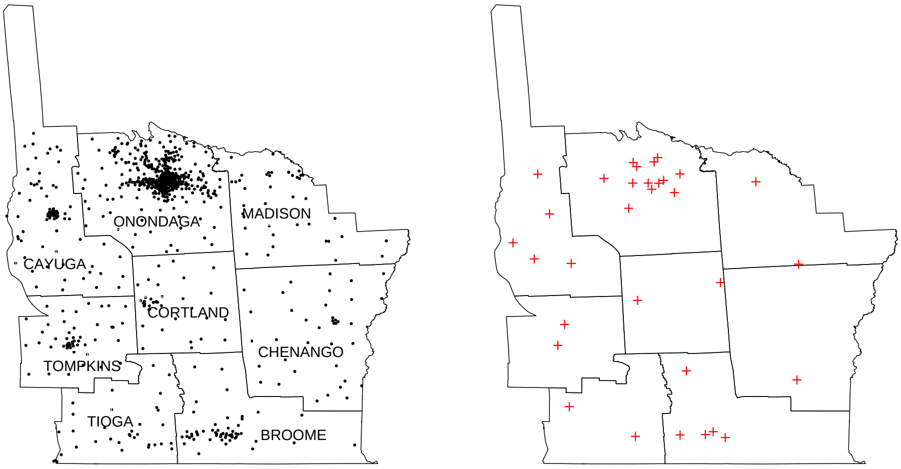

Figure 7: Distribution of leukaemia cases over the 8 counties of the state of New York (left map), and positions of the centroids of the 32 new subregions (right map).

Using this approach, we were able to create new subregions with approximately equal population (some of the points were reassigned to a different subregion in the post-processing stage). In Figure 7 (right), 32 subregion centroids are shown. For each subregion, the population varies from 36036 to 39528 (roughly 3.48 % of the whole population). Under the null hypothesis, these new subregions generate cases with (roughly) the same intensity.

In this grouping, the maximum number of individuals diagnosed with leukaemia within one subregion is 39 versus a minimum of 3 cases, so that the multinomial range statistic is equal to 33. The total number of cases registered in each subregion is presented in Table 2.

Using the exact test based on the range of the sample, the null hypothesis of equiprobable multinomial distribution is rejected at the 0.01 level, since the 99th percentile of the range's exact distribution in the case of 584 events and 32 bins is 25. The null hypothesis is also rejected when one uses 25 subregions (results not shown), which supports previous research about leukaemia cases for this data set [20].

Importantly, in this illustration there is no difference in our conclusions between using the test based on the exact or the approximate distribution of the range, since for $n = 584$ and $m = 32$ the

Subr	Nr	Subr	Nr	Subr	Nr	Subr	Nr	Subr	Nr	Subr	Nr	Subr	Nr	Subr	Nr
1	34	5	23	9	18	13	20	17	25	21	24	25	14	29	5
2	28	6	20	10	14	14	17	18	20	22	12	26	12	30	4
3	13	7	18	11	17	15	14	19	9	23	31	27	24	31	3
4	23	8	27	12	39	16	13	20	21	24	24	28	17	32	3

Table 2: Number of subjects (Nr) diagnosed with leukaemia in each of 32 subregions (Subr).

approximate distribution is close to the exact one. For smaller data sets the use of the exact test would however be quite preferable.

10. Discussion

The use of the order statistics computed from the (clearly dependent) multinomial counts opens the possibility of revising several approaches, which have been followed for many decades to assess deviations from the equal probability null hypothesis. One should expect some of the new test statistics to perform particularly well against some alternatives, for example when one or a few cells have very high (or very low) probabilities associated with them.

From our small simulation exercise, these test statistics appear to benefit greatly from the use of their exact distributions, avoiding the problems due to the use of approximations. Note that more extensive explorations can be pursued using the code that we are providing in this article.

The exact distributions could also be used to develop additional test statistics, best suited for specific problems. For example, in the disease clustering illustration one could recognize that disease cluster studies often originate from a perceived high risk of disease in some subregion(s). Such investigations then run the risk of falling into what is known as the Texas sharpshooter fallacy, i.e. finding an apparently statistically significant difference in risk, when comparing the disease rate in that subregion to the rest of the region. The fact that one is looking at that comparison conditionally on having observed a high rate, if not taken into account properly, can easily produce false positive errors. Note that in such setting, one is actually comparing exactly the largest multinomial order statistic (or one of the largest ones, or their sum) to the counts observed in the other subregions. Given the conditioning on the total number of cases observed, this is the same as constructing the test on just the largest (or the sum of the few largest) order statistics form the multinomial vector. Such statistics could then naturally be used to test the null hypothesis of homogeneous distribution of the disease risk. Our algorithms therefore allow one to take into account the selection process of the subregion under investigation, within the appropriate statistical testing framework.

Other interesting applications of the exact tests are in the field of risk management. In particular, in credit risk modeling of the number of defaults of counterparties that share the same creditworthiness in terms of rating, but belong to different industrial sectors [21]. In these settings, given the small number of default events and the large number of sectors for the exposures in a granular portfolio, exact multinomial tests may represent an advantage with respect to the more commonly used χ^2 approximation. A similar reasoning is even more relevant if one restricts her attention on low default portfolios (LPDs) [22].

Data accessibility statement

The data used for the epidemiological application are available in [20]. All the other data are based on simulations, for which all the necessary information for reproduction is given.

Research ethics statement

This statement does not apply.

Permission to carry out fieldwork statement

This statement does not apply.

Funding statement

PC acknowledges the support of the Marie Curie CIG Multivariate Shocks (PCIG13-GA-2013-618794) from the European Union.

Competing interests statement

No competing interest present.

Authors' contributions statement

MB found the original work by Rappeport [9], proposed the idea of the paper, developed the part about UMP tests and helped in drafting and correcting the manuscript. PC performed the literature review, co-developed the algorithms, worked on the graphical part and drafted the manuscript. AO co-developed the algorithms, worked on the two illustrations and wrote the Matlab codes. All authors gave final approval for publication.

References

1. Andersen EB, Jensen N, Kousgaard N. 1987 *Statistics for Economics, Business Administration, and the Social Sciences*. NY: Springer.
2. Johnson NL, Kotz S, Balakrishnan N. 1997 *Discrete Multivariate Distributions*. NJ: Wiley & Sons.
3. Johnson NL, Kotz S. 1977 *Urn Models and Their Applications*. NJ: Wiley & Sons.
4. Pearson K. 1900. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. Reprinted in *Philosophical Magazine Series 5* **50**:157–175, 2009.
5. Cochran WG. 1954. Some methods for strengthening the common χ^2 tests. *Biometrics* **10**:417–451.
6. Read TRC, Cressie N. 1984. Multinomial goodness-of-fit tests. *Journal of the Royal Statistical Society Series B* **46**:440–464.
7. Young DH. 1962. Two alternatives to the standard χ^2-test of the hypothesis of equal cell frequencies. *Biometrika* **49**:107–116.
8. Corrado CJ. 2011. The exact distribution of the maximum, minimum and the range of Multinomial/Dirichlet and Multivariate Hypergeometric frequencies. *Statistics and Computing* **21**:349–359.
9. Rappeport MA. 1968 *Algorithms and computational procedures for the application of order statistics to queing problems*. PhD Thesis, New York University.
10. Pearson ES, Hartley HO. 1958 *Biometrika Tables for Statisticians 1*. UK: Cambridge University Press.
11. Young DH, Johnson NL. 1960. Some applications of two approximations to the multinomial distribution. *Biometrika* **47**:463–469.
12. Kolchin V, Sevast'yanov B, Chistykov V. 1978 *Random allocations*. Washington DC: V.H Winston & Sons.
13. Dasgupta A. 2017. Exact tail probabilities and percentiles of the multinomial maximum. *Technical Report*, Purdue University, Indiana. See Online PDF.
14. Compiani G. 2012 *Some considerations on the goodness-of-fit problem*. Master Thesis, Bocconi University, Italy.
15. Read TRC, Cressie N. 1988 *Goodness-of-Fit Statistics for Discrete Multivariate Data, Volume 1*. NY: Springer.
16. Lehmann EL, Romano JP. 2005 *Testing statistical hypotheses*. NY: Springer.
17. Mukhopadhyay N. 2000 *Probability and Statistical Inference*. NY: Marcel Dekker.
18. Casella G, Berger RL. 2002 *Statistical Inference*. CA: Duxbury Press.

19. Bonetti M, Pagano M. 2005. The interpoint distance distribution as a descriptor of point patterns, with an application to spatial disease clustering. *Statistics in Medicine* **24**:753–773.
20. Lange N, Ryan L, Billard L, Brillinger D, Conquest L, Greenhouse J (eds). 1994 *Case Studies in Biometry*. NJ: Wiley & Sons.
21. Sironi A, Resti A. 2007 *Risk Management and Shareholders' Value in Banking*. NJ: Wiley Finance.
22. Basel Committee on Banking Supervision. 2005. Validation of low-default portfolios in the Basel II Framework. *Basel Committee Newsletter* **6**. See BCBS Newsletter.

Appendix: Codes and Tables

In this appendix we collect the MATLAB code that implement all the exact algorithms discussed in this article, as well as some tables of critical values.

Algorithm for the Maximum

```
function [P] = max_order_statistic(t,N,I)
%% Script calculates probability of highest order statistic being <=t
% Under equiprobable multinomial
% Input : N — number of balls
%   I — number of urns(cells)
%  t — argument of cdf
P = 0;
if t == 0 && N~=0
    P = 0;
    return
end
if t==0 && N==0
    P=1;
    return
end
if t>=N
    P = 1;
    return
end
if N==0 && I~=0
    P=1;
    return
end
if N==0 && I==0
    P=1;
    return
end
common_term = gammaln(N+1)+gammaln(I+1)—N*log(I);
switch t
    case 1
        if I>=N
            P = exp(gammaln(I+1)—gammaln(I—N+1)—N*log(I)); % explicit calculation of P(n<1><=1)
              calc = [calc;t,N,I,P];
        else
            P = 0;
            calc = [calc;t,N,I,P];
        end
    otherwise
        % range of summation for q
        LowSum = max(0,N-t*I+I);
        UpSum = floor(N/t);
        for q = LowSum:UpSum
            summ_term = (-q*gammaln(t+1)-gammaln(q+1)-gammaln(I-q+1)-gammaln(N-t
                *q+1));
            if I==q
                summ_term_nominator = 0;
            else
                summ_term_nominator = (N-t*q)*log(I-q);
            end
            coef = exp(common_term+summ_term+summ_term_nominator);
                [temp]  = max_order_statistic(t-1,N-t*q,I-q);

            P = P + coef*temp;

        end
end
end
```

Sum of the J Largest Order Statistics

```
function [ P ] = highest_order_statistics( t,N,I,J )
%% Probability of the sum of the first J largest order statistics being smaller
    than t
%   P(sum(1:J)n<i> <= t : N,I) under H1 - equiprobable multinomial
% Input: N- number of trials
% I - number of cells(urns)
% t - argument of cdf
% J - number of largest order statistics
if J>I
    error('J should be smaller or equal than I')
end
if J==I && (t<N )
    error(' Total sum is every time equal to N')
end

if t == 0 && N~=0
    P = 0;
    return
end
if t==0 && N==0
    P=1;
    return
end
if t>=N
    P = 1;
    return
end

%% first term if n<1> <= t/J
P = max_order_statistic(floor(t/J),N,I);
```

```
%% recursive summation over all possible options
for sum_depth = 1 : J-1
    rangeArg = [];
    cur_depth = 1;
    P = P + recursive_sum(t,N,I,J,sum_depth,cur_depth,rangeArg);
end
end
function S = recursive_sum(t,N,I,J,sum_depth,cur_depth,rangeArg)
%% auxiliary function for calculating sum of nested loops
S = 0;
if cur_depth <= sum_depth % either increment summation depth or calculate the term
    if cur_depth == 1
        cur_range(1) = floor(t/J+1);
        cur_range(2) = t-sum_depth+1;
    else
        cur_range(1) = floor((t - sum(rangeArg(1:cur_depth-1)))/(J- cur_depth+1)+1);
        cur_range(2) = min( rangeArg(cur_depth-1), t - sum(rangeArg(1:cur_depth-1)));
    end
    for r=cur_range(1):cur_range(2)
       rangeArg(cur_depth) = r;
        S = S + recursive_sum(t,N,I,J,sum_depth,cur_depth+1,rangeArg);
    end
else
    prob_arg = floor( (t-sum(rangeArg))/(J-sum_depth));
    temp_p = max_order_statistic(prob_arg, N - sum(rangeArg),I-sum_depth);
    common_term = gammaln(N+1) + gammaln(I+1) - N*log(I);
    coef = (N-sum(rangeArg))*log(I-sum_depth) - gammaln(I-sum_depth+1) - gammaln( N - sum(rangeArg)+1);
    for k=1:numel(rangeArg)
        coef = coef - gammaln(rangeArg(k)+1);
    end
    equal_statistics = unique(rangeArg);
    for k=1:numel(equal_statistics)
        temp = numel(find(rangeArg == equal_statistics(k)));
        coef = coef   - gammaln(temp+1);
    end
    S  = temp_p*exp(common_term+coef);

end
end
```

Algorithm for the Minimum

```
function P = smallest_order_value( t,N,I )
%% Function to calculate the probability of smallest order statistic to be >= than t for equiprobable multinomial
% Input:
% t  - argument of "survival" function
% N - number of balls
% I  - number of cells
```

```

P = 0;
% add for exceptions for "naive" input
if t>floor(N/I)
    P=0;
    return
end
if t==0
    P=1;
    return
end
    aux = max_for_min(N,N,I,calc,t);
    P = P + aux;
end

function [aux,calc] = max_for_min(t_max,N,I,calc,t)
aux = 0;
if t_max<t
    if N==0 && I==0
        aux=1;
        return
    else
        aux = 0;
        return
    end
else
    if N==0 && I == 0
        aux=1;
        return
    end
    if t_max==1
        if I==N
            aux = exp(gammaln(I+1)-gammaln(I-N+1)-N*log(I)); % explicit
                calculation of P(n<1><=1)
            return
        else
            aux= 0;
            return
        end
     end
    if N==0 && I~=0
        aux=0;
        return
    end
    common_term = gammaln(N+1)+gammaln(I+1)-N*log(I);
    LowSum = max(0,N-t_max*I+I);
    UpSum = floor(N/t_max);
    for q = LowSum:UpSum
        summ_term = (-q*gammaln(t_max+1)-gammaln(q+1)-gammaln(I-q+1)-gammaln(N-
            t_max*q+1));
        if I==q
            summ_term_nominator = 0;
        else
            summ_term_nominator = (N−t_max∗q)∗log(I−q);
        end
        coef = exp(common_term+summ_term+summ_term_nominator);
        [temp]  = max_for_min(t_max−1,N−t_max∗q,I−q,t);
        aux = aux + coef∗temp;
    end
end
end
```

Algorithm for the Range

```
function [P] = range_probability( t,N,I )
%% Function to calculate the probability of the range to be < =than t
%for equiprobable multinomial
% Input:
% r  — argument of "cdf" function
% N — number of balls
% I  — number of cells
P = 0;
t= floor(t);
if t>N
    P=1;
    return
end

[P]=max_order_statistic(t,N,I);
prev=[t,N,I];
for t_max = t+1:N
    [aux] = max_for_range(t_max,N,I,prev,t);
    P = P + aux;
    prev = [t_max,N,I];
end
end

function [aux] = max_for_range(t_max,N,I,prev,t)

aux=0;
if [t_max,N,I]==prev
    aux = 0;
    return
end
if prev(1)+1−t_max>t
    if N==0 && I==0
        aux=1;
        return
    else
        aux = 0;
        return
    end
else
    if N==0 && I == 0
        aux=1;
        return
    end
    if t_max==1
        if I==N
            aux = exp(gammaln(I+1)-gammaln(I-N+1)-N*log(I)); % explicit
                calculation of P(n<1><=1)
            return
        else
            aux= 0;
                        return
        end
    end
    if N==0 && I~=0
        aux=0;
                return
    end

    common_term = gammaln(N+1)+gammaln(I+1)-N*log(I);
    LowSum = max(0,N-t_max*I+I);
    UpSum = floor(N/t_max);
    for q = LowSum:UpSum
        summ_term = (-q*gammaln(t_max+1)-gammaln(q+1)-gammaln(I-q+1)-gammaln(N-
            t_max*q+1));
        if I==q
            summ_term_nominator = 0;
        else
            summ_term_nominator = (N-t_max*q)*log(I-q);
        end
        coef = exp(common_term+summ_term+summ_term_nominator);
        [temp]  = max_for_range(t_max-1,N-t_max*q,I-q,prev,t);
        aux = aux + coef*temp;
    end
end
end
```

t		2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
n	m															
5	5	0.710	0.966	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10	5	0.012	0.433	0.836	0.968	0.996	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
15	5	0.000	0.006	0.284	0.700	0.910	0.979	0.996	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000
20	5	0.000	0.000	0.003	0.200	0.584	0.840	0.950	0.987	0.997	0.999	1.000	1.000	1.000	1.000	1.000
25	5	0.000	0.000	0.000	0.002	0.147	0.490	0.769	0.913	0.972	0.992	0.998	1.000	1.000	1.000	1.000
30	5	0.000	0.000	0.000	0.000	0.001	0.113	0.415	0.701	0.872	0.953	0.984	0.995	0.999	1.000	1.000
10	10	0.396	0.873	0.984	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
20	10	0.000	0.127	0.603	0.889	0.976	0.996	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
30	10	0.000	0.000	0.049	0.394	0.753	0.923	0.980	0.995	0.999	1.000	1.000	1.000	1.000	1.000	1.000
40	10	0.000	0.000	0.000	0.022	0.259	0.617	0.848	0.950	0.985	0.996	0.999	1.000	1.000	1.000	1.000
50	10	0.000	0.000	0.000	0.000	0.011	0.173	0.498	0.765	0.908	0.968	0.990	0.997	0.999	1.000	1.000
60	10	0.000	0.000	0.000	0.000	0.000	0.006	0.119	0.401	0.681	0.857	0.944	0.980	0.993	0.998	0.999
15	15	0.219	0.785	0.966	0.996	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
30	15	0.000	0.037	0.432	0.813	0.955	0.991	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
45	15	0.000	0.000	0.009	0.221	0.620	0.867	0.962	0.990	0.998	1.000	1.000	1.000	1.000	1.000	1.000
60	15	0.000	0.000	0.000	0.002	0.114	0.451	0.755	0.911	0.972	0.992	0.998	1.000	1.000	1.000	1.000
75	15	0.000	0.000	0.000	0.000	0.001	0.061	0.321	0.638	0.844	0.942	0.981	0.994	0.998	1.000	1.000
90	15	0.000	0.000	0.000	0.000	0.000	0.000	0.034	0.228	0.530	0.769	0.902	0.963	0.987	0.996	0.999
20	20	0.121	0.706	0.949	0.993	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
40	20	0.000	0.011	0.309	0.742	0.933	0.986	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
60	20	0.000	0.000	0.001	0.123	0.510	0.814	0.944	0.985	0.997	0.999	1.000	1.000	1.000	1.000	1.000
80	20	0.000	0.000	0.000	0.000	0.050	0.329	0.671	0.874	0.958	0.988	0.997	0.999	1.000	1.000	1.000
100	20	0.000	0.000	0.000	0.000	0.000	0.021	0.207	0.532	0.785	0.916	0.971	0.991	0.997	0.999	1.000
120	20	0.000	0.000	0.000	0.000	0.000	0.000	0.010	0.129	0.411	0.689	0.862	0.945	0.980	0.993	0.998
25	25	0.067	0.634	0.931	0.991	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
50	25	0.000	0.003	0.222	0.678	0.912	0.981	0.996	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000
75	25	0.000	0.000	0.000	0.069	0.419	0.764	0.926	0.980	0.995	0.999	1.000	1.000	1.000	1.000	1.000
100	25	0.000	0.000	0.000	0.000	0.022	0.240	0.596	0.837	0.945	0.983	0.995	0.999	1.000	1.000	1.000
125	25	0.000	0.000	0.000	0.000	0.000	0.007	0.133	0.443	0.730	0.891	0.961	0.987	0.996	0.999	1.000
150	25	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.073	0.319	0.617	0.823	0.928	0.973	0.991	0.997
30	30	0.037	0.570	0.914	0.988	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
60	30	0.000	0.001	0.159	0.619	0.891	0.975	0.995	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000
90	30	0.000	0.000	0.000	0.039	0.345	0.717	0.908	0.975	0.994	0.999	1.000	1.000	1.000	1.000	1.000
120	30	0.000	0.000	0.000	0.000	0.010	0.175	0.529	0.802	0.931	0.979	0.994	0.998	1.000	1.000	1.000
150	30	0.000	0.000	0.000	0.000	0.000	0.003	0.086	0.369	0.678	0.866	0.952	0.984	0.995	0.999	1.000
180	30	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.042	0.248	0.553	0.786	0.911	0.967	0.988	0.996
40	40	0.011	0.459	0.880	0.982	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
80	40	0.000	0.000	0.081	0.516	0.850	0.965	0.993	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000
120	40	0.000	0.000	0.000	0.012	0.233	0.631	0.873	0.964	0.991	0.998	1.000	1.000	1.000	1.000	1.000
160	40	0.000	0.000	0.000	0.000	0.002	0.093	0.417	0.737	0.905	0.970	0.991	0.998	0.999	1.000	1.000
200	40	0.000	0.000	0.000	0.000	0.000	0.000	0.035	0.256	0.585	0.819	0.932	0.977	0.993	0.998	0.999
50	50	0.003	0.370	0.848	0.976	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
100	50	0.000	0.000	0.042	0.430	0.811	0.954	0.991	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000
150	50	0.000	0.000	0.000	0.004	0.157	0.556	0.840	0.953	0.988	0.997	0.999	1.000	1.000	1.000	1.000
200	50	0.000	0.000	0.000	0.000	0.000	0.049	0.329	0.677	0.879	0.961	0.989	0.997	0.999	1.000	1.000

Table 3: Distribution of the multinomial maximum. For example, in the case with $n = 10$ and $m = 5$, $P(N_{<1>} \leq 3; n = 10.m = 5) = 0.433$.

t		5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
n	m																		
10	5	0.166	0.588	0.887	0.984	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
15	5	0.000	0.006	0.088	0.392	0.722	0.913	0.981	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
20	5	0.000	0.000	0.000	0.003	0.054	0.277	0.580	0.817	0.939	0.984	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000
25	5	0.000	0.000	0.000	0.000	0.000	0.002	0.037	0.205	0.469	0.720	0.881	0.959	0.988	0.997	0.999	1.000	1.000	1.000
30	5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.027	0.157	0.385	0.631	0.816	0.923	0.972	0.992	0.998	1.000
10	10	0.811	0.967	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
20	10	0.001	0.131	0.415	0.736	0.913	0.978	0.996	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
30	10	0.000	0.000	0.000	0.050	0.206	0.494	0.745	0.898	0.966	0.991	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000
40	10	0.000	0.000	0.000	0.000	0.000	0.023	0.109	0.325	0.573	0.779	0.903	0.963	0.988	0.996	0.999	1.000	1.000	1.000
50	10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.011	0.061	0.216	0.432	0.653	0.816	0.915	0.965	0.987	0.996	0.999
60	10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.006	0.036	0.148	0.325	0.538	0.721	0.852	0.929	0.969
15	15	0.592	0.884	0.978	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
30	15	0.000	0.037	0.172	0.501	0.772	0.922	0.978	0.995	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
45	15	0.000	0.000	0.000	0.009	0.052	0.251	0.511	0.749	0.893	0.962	0.988	0.997	0.999	1.000	1.000	1.000	1.000	1.000
60	15	0.000	0.000	0.000	0.000	0.000	0.002	0.018	0.127	0.316	0.561	0.760	0.888	0.954	0.983	0.994	0.998	1.000	1.000
75	15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.007	0.067	0.193	0.405	0.616	0.786	0.894	0.953	0.981	0.993
90	15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.037	0.119	0.287	0.485	0.674	0.815	0.905
20	20	0.410	0.788	0.948	0.991	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
40	20	0.000	0.011	0.066	0.339	0.635	0.852	0.952	0.987	0.997	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
60	20	0.000	0.000	0.000	0.001	0.012	0.131	0.340	0.610	0.808	0.922	0.972	0.991	0.998	0.999	1.000	1.000	1.000	1.000
80	20	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.052	0.169	0.396	0.622	0.801	0.908	0.963	0.986	0.995	0.999	1.000
100	20	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.022	0.084	0.247	0.451	0.658	0.811	0.907	0.959	0.983
25	25	0.273	0.695	0.912	0.982	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
50	25	0.000	0.003	0.024	0.233	0.515	0.779	0.919	0.976	0.994	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
30	30	0.176	0.612	0.873	0.971	0.995	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
60	30	0.000	0.001	0.008	0.163	0.412	0.708	0.884	0.963	0.990	0.998	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000
40	40	0.070	0.478	0.790	0.945	0.989	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
80	40	0.000	0.000	0.001	0.082	0.257	0.579	0.808	0.932	0.979	0.995	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000
50	50	0.026	0.378	0.708	0.915	0.981	0.997	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
100	50	0.000	0.000	0.000	0.042	0.156	0.471	0.733	0.897	0.967	0.991	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Table 4: Distribution of the sum of the first two order statistics: $N_{<1>} + N_{<2>}$. For example, in the case with $n = 10$ and $m = 10$, $P(N_{<1>} + N_{<2>} \leq 5; n = 10.m = 10) = 0.811$.

t		8	9	10	11	12	13	14	15	16	18	20	22	24	26	28	30	32	34
n	m																		
10	10	0.987	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
20	10	0.020	0.174	0.468	0.747	0.913	0.978	0.996	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
30	10	0.000	0.000	0.000	0.005	0.064	0.230	0.472	0.708	0.869	0.986	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000
40	10	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.028	0.120	0.509	0.856	0.977	0.998	1.000	1.000	1.000	1.000	1.000
50	10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.014	0.181	0.559	0.858	0.972	0.996	1.000	1.000	1.000
60	10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.040	0.254	0.610	0.868	0.970	0.995	0.999
15	15	0.869	0.972	0.996	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
30	15	0.000	0.038	0.174	0.405	0.669	0.856	0.950	0.986	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
45	15	0.000	0.000	0.000	0.000	0.009	0.052	0.160	0.362	0.592	0.898	0.986	0.999	1.000	1.000	1.000	1.000	1.000	1.000
60	15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.018	0.186	0.571	0.866	0.974	0.997	1.000	1.000	1.000	1.000
75	15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.028	0.226	0.578	0.850	0.964	0.994	0.999	1.000
90	15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.053	0.269	0.598	0.846	0.958	0.991
20	20	0.714	0.908	0.979	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
40	20	0.000	0.011	0.066	0.197	0.448	0.697	0.864	0.950	0.985	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
60	20	0.000	0.000	0.000	0.000	0.001	0.012	0.047	0.172	0.373	0.770	0.953	0.994	1.000	1.000	1.000	1.000	1.000	1.000
80	20	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.066	0.351	0.718	0.923	0.986	0.998	1.000	1.000	1.000
100	20	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.004	0.089	0.358	0.695	0.901	0.977	0.996	0.999
25	25	0.558	0.823	0.949	0.989	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
50	25	0.000	0.003	0.024	0.087	0.290	0.550	0.763	0.898	0.963	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
30	30	0.416	0.728	0.909	0.977	0.995	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
60	30	0.000	0.001	0.008	0.036	0.189	0.429	0.662	0.837	0.934	0.993	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
40	40	0.209	0.553	0.816	0.941	0.985	0.997	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
80	40	0.000	0.000	0.001	0.005	0.086	0.260	0.479	0.705	0.862	0.981	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000
50	50	0.096	0.417	0.723	0.896	0.970	0.993	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
100	50	0.000	0.000	0.000	0.001	0.042	0.156	0.332	0.577	0.780	0.962	0.996	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Table 5: Distribution of the sum of the first three largest order statistics: $N_{<1>} + N_{<2>} + N_{<3>}$. For example, in the case with $n = 15$ and $m = 15$, $P\left(\sum_{i=1}^{3} N_{<i>} \leq 8; n = 15, m = 15\right) = 0.869$.

t		0	1	2	3	4	5	6	7
n	m								
5	5	0.962	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10	5	0.477	0.988	1.000	1.000	1.000	1.000	1.000	1.000
15	5	0.171	0.680	0.994	1.000	1.000	1.000	1.000	1.000
20	5	0.057	0.325	0.783	0.997	1.000	1.000	1.000	1.000
25	5	0.019	0.135	0.446	0.844	0.998	1.000	1.000	1.000
30	5	0.006	0.052	0.214	0.539	0.882	0.999	1.000	1.000
40	5	0.001	0.007	0.040	0.140	0.356	0.669	0.926	0.999
30	10	0.371	0.931	0.999	1.000	1.000	1.000	1.000	1.000
40	10	0.142	0.614	0.972	0.999	1.000	1.000	1.000	1.000
50	10	0.051	0.306	0.757	0.986	1.000	1.000	1.000	1.000
60	10	0.018	0.133	0.450	0.841	0.993	1.000	1.000	1.000
80	10	0.003	0.022	0.104	0.319	0.657	0.926	0.997	0.999
90	15	0.03	0.208	0.616	0.945	0.999	1.000	1.000	1.000
100	20	0.113	0.556	0.954	0.999	1.000	1.000	1.000	1.000

Table 6: Distribution of the multinomial minimum. For example, in the case with $n = 25$ and $m = 5$, $P(\min N_i \leq 2; n = 25, m = 5) = 0.446$.

t		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
n	m																
5	5	0.038	0.710	0.966	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10	5	0.012	0.321	0.601	0.867	0.971	0.996	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
15	5	0.006	0.181	0.386	0.659	0.854	0.953	0.988	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
20	5	0.003	0.116	0.265	0.500	0.716	0.868	0.949	0.983	0.995	0.999	1.000	1.000	1.000	1.000	1.000	1.000
25	5	0.002	0.081	0.193	0.388	0.595	0.770	0.887	0.952	0.982	0.994	0.998	1.000	1.000	1.000	1.000	1.000
30	5	0.001	0.059	0.146	0.308	0.497	0.676	0.815	0.906	0.958	0.983	0.994	0.998	0.999	1.000	1.000	1.000
10	10	0.000	0.396	0.873	0.984	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
20	10	0.000	0.060	0.237	0.640	0.896	0.977	0.996	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
30	10	0.000	0.016	0.079	0.313	0.612	0.841	0.949	0.986	0.997	0.999	1.000	1.000	1.000	1.000	1.000	1.000
40	10	0.000	0.006	0.032	0.158	0.380	0.637	0.831	0.935	0.978	0.994	0.998	1.000	1.000	1.000	1.000	1.000
50	10	0.000	0.002	0.015	0.086	0.236	0.458	0.679	0.840	0.932	0.974	0.991	0.997	0.999	1.000	1.000	1.000
60	10	0.000	0.001	0.008	0.050	0.151	0.325	0.535	0.724	0.858	0.935	0.973	0.990	0.997	0.999	1.000	1.000
15	15	0.010	0.061	0.510	0.875	0.978	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
30	15	0.000	0.016	0.079	0.313	0.612	0.841	0.949	0.986	0.997	0.999	1.000	1.000	1.000	1.000	1.000	1.000
45	15	0.000	0.001	0.028	0.111	0.299	0.542	0.756	0.892	0.959	0.986	0.996	0.999	1.000	1.000	1.000	1.000
60	15	0.000	0.001	0.008	0.050	0.151	0.325	0.535	0.724	0.858	0.935	0.973	0.990	0.997	0.999	1.000	1.000
75	15	0.000	0.000	0.005	0.023	0.082	0.198	0.365	0.553	0.719	0.843	0.920	0.963	0.984	0.994	0.998	0.999
90	15	0.000	0.000	0.002	0.013	0.047	0.124	0.250	0.412	0.581	0.729	0.841	0.914	0.957	0.980	0.991	0.996
20	20	0.000	0.121	0.706	0.949	0.993	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
40	20	0.000	0.002	0.029	0.321	0.745	0.934	0.986	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
60	20	0.000	0.000	0.002	0.065	0.274	0.607	0.850	0.954	0.988	0.997	0.999	1.000	1.000	1.000	1.000	1.000
80	20	0.000	0.000	0.000	0.015	0.089	0.301	0.586	0.812	0.929	0.977	0.993	0.998	1.000	1.000	1.000	1.000
100	20	0.000	0.000	0.000	0.004	0.030	0.138	0.349	0.603	0.802	0.917	0.969	0.990	0.997	0.999	1.000	1.000
25	25	0.000	0.067	0.634	0.931	0.991	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
50	25	0.000	0.000	0.010	0.227	0.680	0.912	0.981	0.996	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000
75	25	0.000	0.000	0.000	0.030	0.178	0.505	0.799	0.936	0.983	0.996	0.999	1.000	1.000	1.000	1.000	1.000
100	25	0.000	0.000	0.000	0.005	0.041	0.201	0.483	0.747	0.901	0.967	0.990	0.997	0.999	1.000	1.000	1.000
30	30	0.000	0.037	0.570	0.914	0.988	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
60	30	0.000	0.000	0.003	0.161	0.620	0.891	0.975	0.995	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000
40	40	0.000	0.000	0.201	0.720	0.941	0.990	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
80	40	0.000	0.000	0.001	0.027	0.195	0.587	0.857	0.960	0.990	0.998	1.000	1.000	1.000	1.000	1.000	1.000
50	50	0.000	0.003	0.370	0.848	0.976	0.997	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
100	50	0.000	0.000	0.000	0.042	0.430	0.811	0.954	0.991	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000

Table 7: Distribution of the multinomial range. For example, in the case of $n = 15$ and $m = 15$, $P(\max N_i - \min N_i \leq 3; n = 15, m = 15) = 0.51$.