

How to Calculate Multinomial Distribution Probabilities in R Using `dmultinom()`

Authored by
stats writer

December 2, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Calculate Multinomial Distribution Probabilities in R Using `dmultinom()`*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=103573>

The Multinomial Distribution is an essential probability distribution in statistics, used when analyzing the outcomes of multiple independent trials where each trial can result in one of several categories. It generalizes the Binomial Distribution, moving beyond simple success/failure to handle k distinct outcomes. In the statistical programming environment R, calculating probabilities associated with the Multinomial Distribution is streamlined through the built-in function, `dmultinom()`. This powerful function allows data scientists and analysts to quickly determine the likelihood of obtaining a specific set of counts across various categories.

The primary use of `dmultinom()` is to calculate the probability mass function (PMF) for a given result. It requires inputs detailing the counts observed for each outcome (often provided as a vector), the total number of trials conducted, and the inherent probability associated with each category. Furthermore, while `dmultinom()` provides the exact probability of a specific outcome combination, it can theoretically be combined with other statistical functions in R, such as `pnorm()` or `qnorm()`, for solving more intricate and cumulative probability scenarios when approximating the distribution through related techniques.

Understanding the Multinomial Distribution

The Multinomial Distribution rigorously describes the likelihood of obtaining a specific number of counts, denoted x_1, x_2, \dots, x_k , for k different, mutually exclusive outcomes. This is predicated on the assumption that the probability p_i of each outcome is fixed and remains constant across all independent trials. Unlike the Binomial case which only tracks successes versus failures, the Multinomial handles scenarios where there are three or more discrete results in a fixed number of total trials.

This distribution is foundational in fields like genetics, survey analysis, and quality control whenever categorized data is involved. It is essential for determining the statistical significance of observed frequencies compared to expected theoretical frequencies, provided the total number of trials (n) is fixed and finite. Before calculating any results in R, it is vital to confirm that the observed data adheres to the Multinomial prerequisites: fixed number of trials, independent trials, mutually exclusive outcomes, and fixed probabilities for each category.

The Mathematical Foundation: Multinomial Formula

If a random experiment X follows a multinomial distribution--meaning n independent trials are conducted, and each trial results in one of k possible outcomes--then the probability that outcome 1 occurs exactly x_1 times, outcome 2 occurs exactly x_2 times, and so forth, up to outcome k occurring x_k times, is calculated using the following probability mass function (PMF). This formula combines the probability of the specific sequence occurring with the number of ways that sequence can be ordered:

$$\text{Probability} = n! * (p_1^{x_1} * p_2^{x_2} * \dots * p_k^{x_k}) / (x_1! * x_2! * \dots * x_k!)$$

Applying the Formula: Key Parameters

Understanding the components of the formula is critical for accurate application, particularly when translating real-world problems into input for statistical software like R. The numerator handles the combinatorial aspect (the number of ways to arrange the counts), and the denominator ensures the probabilities of the specific counts are accounted for.

The key variables used in this calculation are defined as follows:

n: Represents the **total number of events** or trials conducted in the experiment. It must always be true that $n = x_1 + x_2 + \dots + x_k$.

x_i : Denotes the specific **number of times outcome i occurs** (the count we are interested in finding the probability for).

p_i : Indicates the **probability** that outcome i occurs in any single, given trial. Crucially, the sum of all probabilities must equal one: $\sum_{i=1}^k p_i = 1$.

Calculating Probability in R using `dmultinom()`

To efficiently solve complex multinomial probability problems without manual calculation of factorials, the statistical software R provides the dedicated function `dmultinom()`. This function acts as the probability mass function for the Multinomial Distribution, calculating $P(X_1=x_1, \dots, X_k=x_k)$.

The general syntax structure for calling this function is straightforward, requiring two aligned vectors of data:

`dmultinom(x=c(1, 6, 8), prob=c(.4, .5, .1))`

In this specific example, we are calculating the probability of observing 1 instance of Outcome 1 (which has a 40% chance), 6 instances of Outcome 2 (50% chance), and 8 instances of Outcome 3 (10% chance), across a total of $n=15$ trials. This clearly illustrates the one-to-one mapping between the counts vector and the probability vector.

The required parameters for `dmultinom()` are as follows:

x: This is a vector representing the frequency or count of each specific outcome (x_1, x_2, \dots, x_k). The total number of trials (n) is implicitly derived by summing the elements in this vector.

prob: This is a vector representing the fixed probability of each corresponding outcome (p_1, p_2, \dots, p_k). It is an absolute requirement that the sum of these probabilities must equal 1.

The following practical examples demonstrate how to utilize this function effectively across various statistical scenarios, moving from electoral analysis to sampling problems.

Example 1: Election Outcome Analysis

Consider a local mayoral election involving three candidates: A, B, and C. Based on prior polls and historical data, the established probability of a voter choosing each candidate is known: Candidate A receives 10% (0.10) of the votes, Candidate B receives 40% (0.40), and Candidate C receives 50% (0.50). This defines our probability vector `prob=c(.1, .4, .5)`.

If we decide to select a relatively small, random sample of 10 voters ($n=10$), we want to determine the precise likelihood that the sample composition aligns with a specific breakdown: exactly 2 voters chose candidate A, 4 voters chose candidate B, and 4 voters chose candidate C. This defines our count vector `x=c(2, 4, 4)`. Since the sample size is fixed and the choice probabilities are independent, this is a perfect application of the Multinomial Distribution.

We use the following code in R to answer this specific probability question using the `dmultinom()` function:

```
# Calculate the multinomial probability for the specific voter breakdown  
dmultinom(x=c(2, 4, 4), prob=c(.1, .4, .5))
```

```
0.0504
```

The resulting calculation shows that the probability of selecting a random group of 10 voters such that exactly 2 voted for A, 4 voted for B, and 4 voted for C is precisely **0.0504**, or 5.04%. This demonstrates the function's ability to handle fixed sample sizes and predetermined outcome probabilities efficiently.

Example 2: Sampling with Replacement from an Urn

In this hypothetical sampling scenario, we consider an urn where the underlying probability of drawing various colors is known. Suppose the probabilities are Yellow: 0.60, Red: 0.20, and Blue: 0.20. Our probability vector is set as `prob=c(.6, .2, .2)`. The sampling is conducted "with replacement," guaranteeing that the probabilities remain constant for every draw--a critical condition for the Multinomial Distribution.

We perform 4 independent draws from the urn ($n=4$). The specific outcome we are interested in is the probability that all 4 selected balls are Yellow. This outcome corresponds to 4 Yellow balls, 0 Red balls, and 0 Blue balls. Therefore, our count vector is `x=c(4, 0, 0)`.

We utilize the `dmultinom()` function in R to compute this specific outcome:

```
# Calculate multinomial probability for 4 Yellow, 0 Red, 0 Blue balls  
dmultinom(x=c(4, 0, 0), prob=c(.6, .2, .2))
```

0.1296

The output confirms that the probability of drawing 4 yellow balls consecutively, given the underlying probabilities, is exactly **0.1296**. If this problem were treated as a Binomial distribution (Yellow vs. Not Yellow), the result would be identical, highlighting how the Multinomial simplifies to the Binomial when only one outcome category has a non-zero count.

Example 3: Analyzing Sports Match Results

Suppose two students play chess against each other. The outcomes of any single game are categorized into three possibilities: Student A wins (probability $p_A=0.5$), Student B wins ($p_B=0.3$), or they tie ($p_T=0.2$). We verify that $0.5 + 0.3 + 0.2 = 1.0$. This probability breakdown is reflected in the vector `prob=c(.5, .3, .2)`.

If they play 10 games in total ($n=10$), we want to find the probability of a specific outcome distribution: Player A wins 4 times, Player B wins 5 times, and they tie 1 time. The count vector defining this specific result is `x=c(4, 5, 1)`.

We use the `dmultinom()` function in R to determine this specific joint probability:

```
# Calculate the joint multinomial probability for chess outcomes  
dmultinom(x=c(4, 5, 1), prob=c(.5, .3, .2))
```

0.0382725

The resulting calculation shows that the probability of observing exactly 4 wins for Player A, 5 wins for Player B, and 1 tie across 10 games is approximately **0.038**. This example clearly demonstrates the utility of the Multinomial Distribution in modeling repeated events with multiple distinct, predefined outcomes.

Choosing Between Multinomial and Binomial Distributions

A frequent point of confusion is when to use the Multinomial Distribution versus the simpler Binomial Distribution. The key distinction lies in the number of possible outcomes per trial. The Binomial Distribution is appropriate only if the outcome of a trial can be categorized into exactly two groups (e.g., success/failure). If, however, the trial can result in three or more distinct, non-

overlapping categories, the Multinomial Distribution must be employed.

For example, in the chess scenario (Example 3), if we were only tracking Student A's wins (success) versus non-wins (failure), we could use the Binomial model with $n=10$ and $p=0.5$. But since we track three separate categories (A wins, B wins, Tie), the Multinomial framework is necessary to calculate joint probabilities for all outcomes simultaneously based on their respective probabilities.

Additional Resources for Data Analysis in R

The Multinomial Distribution, implemented via `dmultinom()` in R, is a powerful tool for analyzing categorical count data. For those looking to delve deeper into related statistical concepts and applications within R, the following tutorials provide additional information about this distribution and broader statistical methods for handling discrete data: