SESUG 2020 Paper 141

Custom SAS® Studio Tasks for All Occasions

John Stephen Taylor, StatistiCode, LLC

ABSTRACT

SAS is an analytics leader that empowers its customers to turn data into decisions, and SAS Studio Tasks have made it easier than ever for users to run complex statistical analyses using a simple point-and-click interface. The suite of tasks that comes preloaded with SAS Studio is impressive, but customers will always have reporting and analytics needs that cannot be met by these initial tasks. Fortunately, it is possible for SAS users to develop their own custom tasks.

In this paper, I discuss a custom SAS Studio Task I developed involving multinomial probability. The multinomial distribution is a generalization of the binomial distribution and models probabilities for *m* possible outcomes, each with a fixed probability of success, in a predetermined number of *n* trials. This task has many industry applications, such as in manufacturing engineering and quality assurance. However, its applications are theoretically limitless and not restricted to industry. For example, this custom SAS Studio Task may be used to determine sample sizes for experiments with multinomial responses in the manufacture of medical devices and to collect LEGO minifigures. As a statistician and SAS programmer, I believe that, with the right amounts of practicality and creativity, custom SAS Studio Tasks can be developed for all occasions!

INTRODUCTION

There is a certain joy that comes from helping others, especially when that help involves more than simply providing an answer. Giving someone a tool that empowers them to learn and discover a solution on their own is a very gratifying experience. A custom SAS Studio Task is an example of such a tool.

SAS Studio is a web-browser based interface for SAS, and it includes many preloaded tasks. SAS Studio Tasks employ an intuitive interface that allows users to provide values to options, which automatically generate predefined SAS code. In essence, tasks are SAS macros that do not require the user to ever view or modify the actual code. This is a huge benefit to many users, even those familiar with SAS.

Experienced SAS users can develop their own custom tasks by writing the underlying XML, Apache Velocity Template Language (VTL), and SAS programming code. This paper describes a custom task developed to solve a sample size problem involving multinomial probability.

MULTINOMIAL DISTRIBUTION

A classic binomial distribution problem involves flipping a coin *n* times, observing how many times the coin lands on heads (or tails), and calculating the probability of observing that count. In this example, each of the independent *n* trials has exactly 2 outcomes (i.e., heads or tails), and each outcome has a fixed probability of success. For example, in the case of a fair coin, there is a 50% probability of observing heads and a 50% probability of observing tails.

If, instead of flipping a coin, we were rolling a 6-sided die *n* times and observing how many times the die landed on the number 3, this would be an example of the multinomial distribution. The multinomial distribution is a generalization of the binomial distribution, where there are *m* outcomes instead of 2. The *n* trials must still be independent, and the probability of each of the *m* outcomes must also be fixed.

Regarding the probabilities of success, in both the binomial and multinomial distributions, the sum of the probabilities must be 100% since exactly one outcome must always occur per trial, and all outcomes are mutually exclusive.

The probability mass function for the multinomial distribution is given by:

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_m = x_m) = \frac{n!}{x_1!\, x_2! \cdots x_m!} p_1^{x^1} \times p_2^{x^2} \times \cdots \times p_m^{x^m}$$

where x_i = non-negative count of outcome i, p_i = probability of success of outcome i, and the sum of the x_i's = n.

While this formula is straightforward, the number of distinct combinations of the x_i's quickly becomes very large as n and m increase, and directly calculating multinomial probability distributions becomes infeasible for even moderate values of n and m. For example, when n = 10 and m = 5, there are 1,001 distinct combinations of x_i's possible.

INDUSTRY PROBLEM – QUALITY ASSURANCE

Suppose a manufacturing plant produces a large batch of a product each shift. During the first step of the process, injection molding produces an essential part for the final finished product. During each shift, there are 4 different molds concurrently used, and each mold is expected to yield a similar final product that meets manufacturing specifications (within certain tolerances).

At the end of the shift, quality assurance is performed by selecting a sample of 8 items from the recently completed batch. Measurements are taken on the sampled items and statistical tests are performed to ensure that the batch meets specifications and can be released. Since it is known that the injection molding process is crucial to the specifications of the final product, at least one item from each of the 4 molds should be included in the sample.

This approach sounds straightforward. However, there is one additional piece of crucial information. When examining the items in the final batch, there is no way to identify which mold was used for which items. With this additional information, how certain can we be that the sample of 8 items includes at least 1 item from each of the 4 molds?

The manufacturing engineers inform us that each mold is expected to produce 25% of the final product in the batch. In other words, each mold is equally likely to have been used. We are selecting a sample of size 8, and each sampled item comes from exactly one of the 4 molds. Furthermore, the 8 sampled items are expected to be independent from one another. In determining how probable it is that each of the 4 molds is represented at least once in the 8 sampled items, this is a multinomial distribution problem.

Thinking in terms of x_i's, a sample of 8 items might include 2 items from each mold ($X_1 = 2$, $X_2 = 2$, $X_3 = 2$, $X_4 = 2$). It is also possible a sample might include $X_1 = 5$, $X_2 = 1$, $X_3 = 1$, $X_4 = 1$ items from the 4 molds. Both examples would meet our requirement of having at least 1 item from each mold. An example that would not meet the requirement is $X_1 = 4$, $X_2 = 2$, $X_3 = 2$, $X_4 = 0$ since no items from mold 4 are included.

The objective is to determine the multinomial probability distribution (i.e., all possible x_i combinations and their associated probabilities), and calculate the total probability for combinations where each $x_i \geq 1$.

CUSTOM SAS STUDIO TASK

The custom SAS task I developed to solve this problem is called the Multinomial Complete Set Sampling task. The task and user interface are built using many of the common SAS task elements:

- **Registration**: contains the metadata for the task and describes the name, description, version
- **Metadata**: identifies the options needed to run the task
- **UI**: determines the layout of the user interface (UI)
- **Dependencies**: hides/shows options in the UI based on values of other options
- **Code Template**: uses a blend of Apache VTL and SAS programming to generate SAS code based on the options provided by the user

Figure 1 illustrates the task UI in its simplest form. The user provides values for the number of independent trials (*n*) and the number of outcomes (*m*). A radio button allows the user to specify whether the probabilities of success (p_i) are all the same or not. When the default value of "Yes" is selected, the probabilities of success are all automatically calculated by the code to be (1 / *m*). The final piece of required information is an output folder location since this task produces a PDF report summarizing the options, the calculation method, and the result.

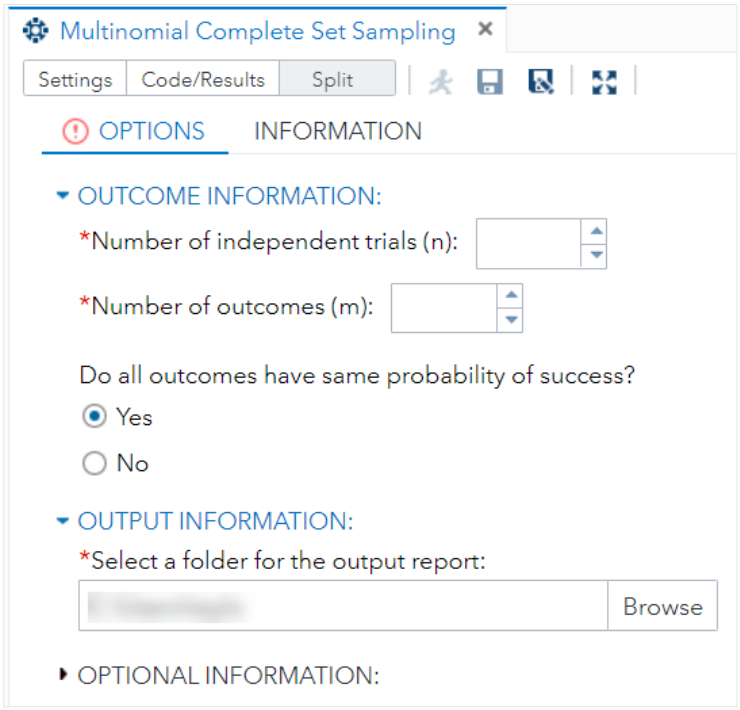


Figure 1. Multinomial Complete Set Sampling task basic interface

SAS task developers may include prompt and error messages to guide users in providing values for options (Figure 2). For number of independent trials (*n*), the user is expected to provide a value between 3 and 500 (inclusive).

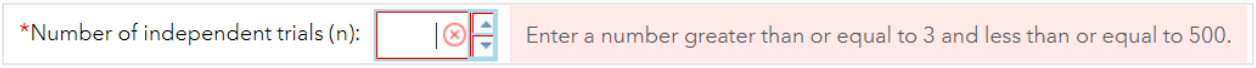


Figure 2. Number of independent trials prompt message

The number of outcomes (*m*) is expected to be between 3 and 20 (inclusive) (Figure 3). The upper limits for *n* and *m* were chosen mostly to minimize processing time. However, both limits can be increased, and tests have been performed for much larger values.

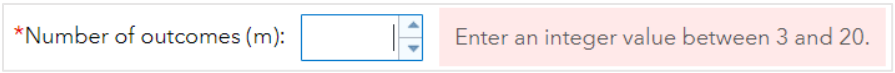


Figure 3. Number of outcomes prompt message

The user selects a radio button value to indicate whether the probability of success for each of the *m* outcomes is the same (“Yes”) or different (“No”). When the user selects “No”, they must provide probability values for each of the *m* outcomes individually (Figure 4). These probabilities are expected to sum to 100% (±.01% due to rounding). Since the upper limit for *m* is 20, there are 20 spaces provided. However, the user is expected to provide only as many probabilities as are needed. Since probabilities of either 0% or 100% are not acceptable, a range has been imposed.

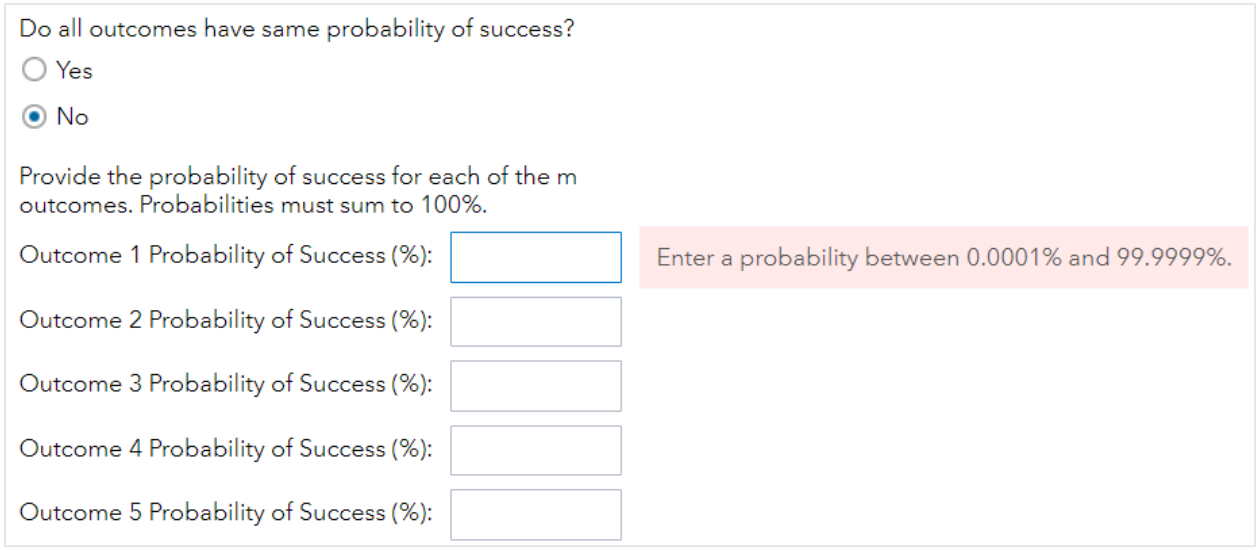


Figure 4. Unequal probability of success (5 of 20 spaces shown)

The final requirement is to provide a location for the output report (Figure 5). This can be done by either copying and pasting a path directly into the space provided, or by clicking the “Browse” button and navigating to the desired location.

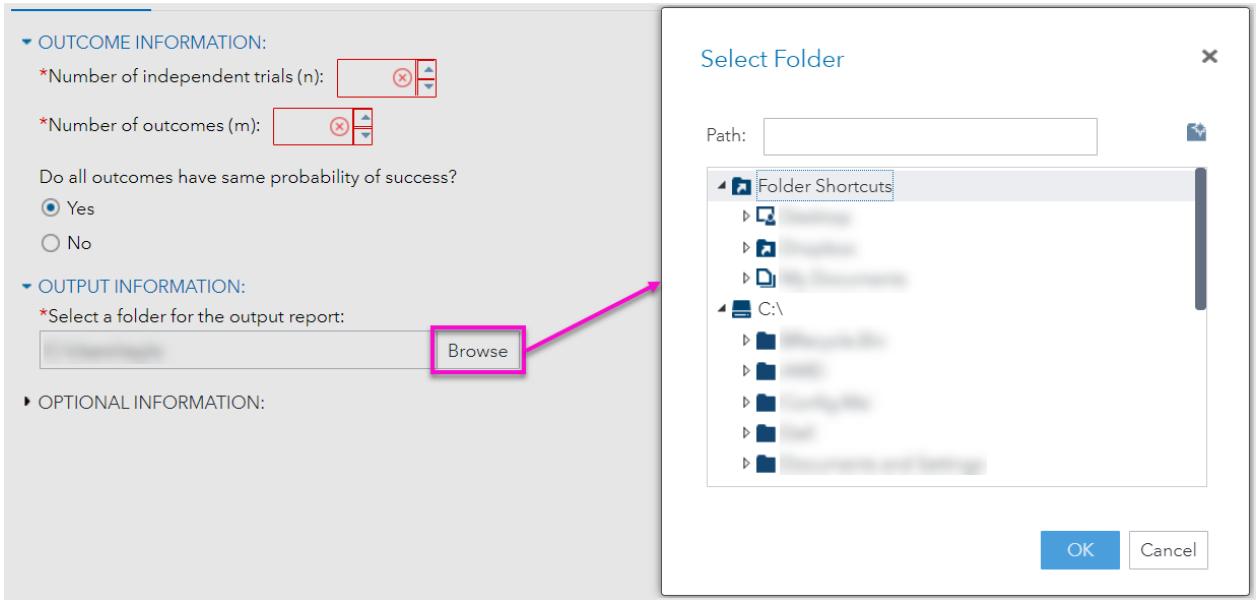


Figure 5. Output folder selection

In the “OPTIONAL INFORMATION” section of the UI, the user may provide text descriptions for each of the *m* outcomes. This feature is used to enhance and customize the final PDF report. If no optional descriptions are provided, default values of “Outcome i” are displayed in the report.

Since this section is optional, it is collapsed or closed by default in the UI. The user can expand or open this section by clicking the arrow to the left of the section header. After expanding the section, the user makes another radio button selection (Figure 6) to prompt the appearance (or disappearance) of the text entry fields (Figure 7). Like probabilities of success, a maximum of 20 spaces are provided for text descriptions for the outcomes.

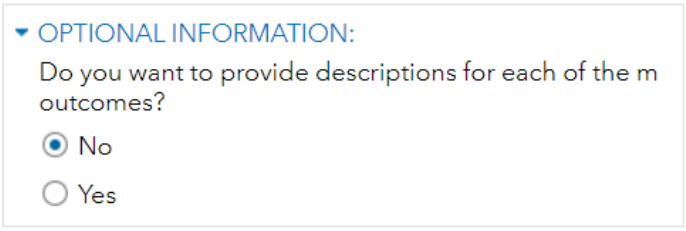


Figure 6. Outcome description radio button selection

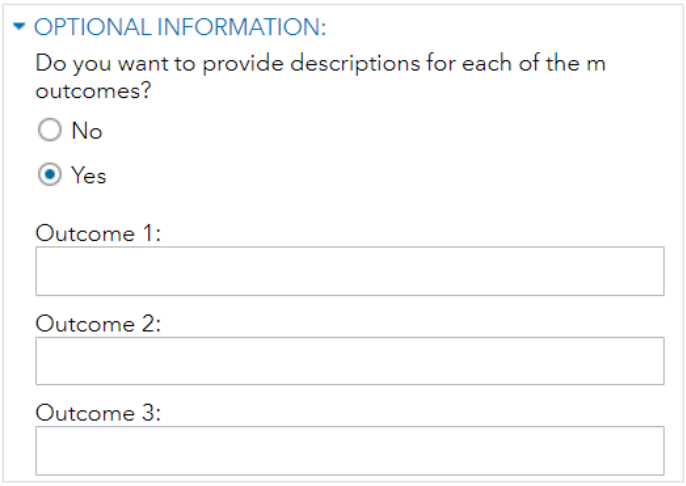


Figure 7. Outcome text descriptions (3 of 20 spaces shown)

QUALITY ASSURANCE SOLUTION

We can run the Multinomial Complete Set Sampling task to determine the probability that a sample of size 8 has at least one representative from each of the 4 equally likely molds used in the manufacturing process (Figure 8).

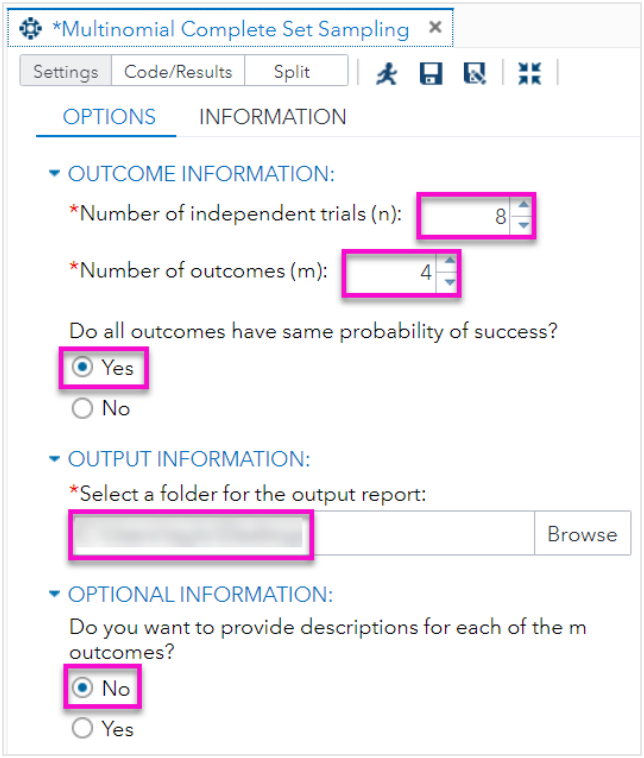
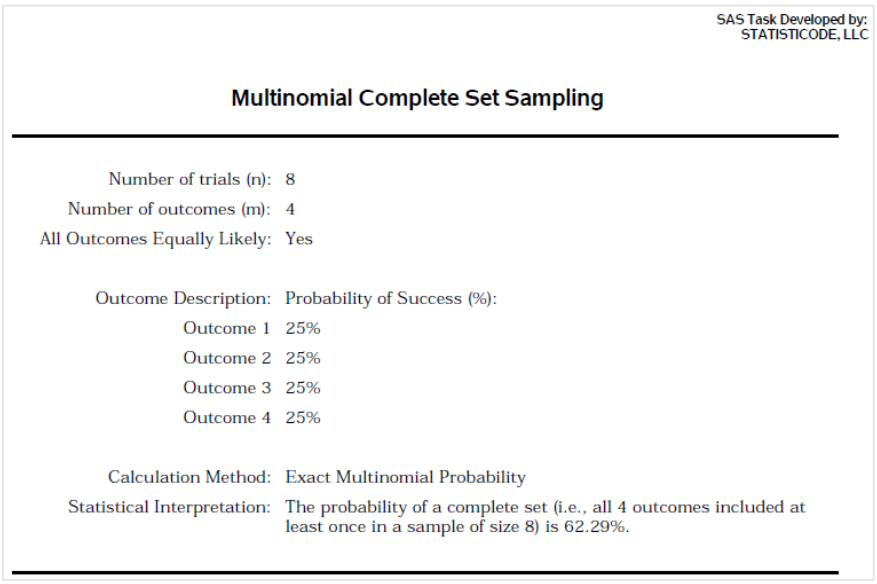


Figure 8. Task with completed options – quality assurance

The question of interest in the quality assurance problem is: how certain can we be that a sample of 8 items includes at least 1 item from each of the 4 molds? The generated PDF report (Figure 9) provides the answer.



SAS Task Developed by:
STATISTICODE, LLC

Multinomial Complete Set Sampling

Number of trials (n):	8
Number of outcomes (m):	4
All Outcomes Equally Likely:	Yes

Outcome Description:	Probability of Success (%):
Outcome 1	25%
Outcome 2	25%
Outcome 3	25%
Outcome 4	25%

Calculation Method:	Exact Multinomial Probability
Statistical Interpretation:	The probability of a complete set (i.e., all 4 outcomes included at least once in a sample of size 8) is 62.29%.

Figure 9. Quality Assurance solution

The first two sections of the report summarize the options that the user provided in the task UI. The third section provides the statistical analysis method and results:

- **Calculation Method**: Exact Multinomial Probability is used when fewer than 10 million calculations are required and processing time is typically less than 1 minute

- **Statistical Interpretation**: an easily understood summary of the results
    - In this example, there is a 62.29% probability that all 4 molds are included at least once in a sample of size 8.

- **Footer information** (not shown):
    - Username of the person running the report
    - Date and time the report was run
    - Page number (X of Y)

Given that there is only a 62% chance of all 4 molds being included in the sample, increasing the sample size to improve the likelihood of a complete set would be recommended. Increasing sample size to 16 gives a 96% chance of all 4 molds being in the sample.

COLLECTING LEGO® MINIFIGURES

To further illustrate the use of the Multinomial Complete Set Sampling task, here is an example of a non-industry problem. LEGO is a company based in Denmark that manufactures plastic construction toys. LEGO bricks and minifigures (or “minifigs”) have been delighting children and adults (myself included) for more than 40 years.

A popular product line sold by LEGO is the collectable minifigure series. These series typically include a complete set of 16-20 minifigures, and they are sold individually in mystery bags (Figure 10). Each mystery bag contains exactly 1 minifigure with no indication of which minifig is included in the package. After the release of a new series, many LEGO enthusiast websites publish estimates of the probability of a package containing a specific minifigure. Very naturally the following question arises: how many mystery bags must a person purchase to have a fair chance of getting a complete set?

Figure 10. LEGO Series 20 mystery bag

This question is very similar to the quality assurance example. In the case of LEGO Series 20, there are 16 possible minifigs (m = 16). Each minifig has a specific probability of being inside the package (p_i). If we select a fixed value for n, we can calculate the probability of obtaining a complete set. We can use the Multinomial Complete Set Sampling task with an n = 32 (i.e., double the number of unique minifigs in the series) and see the result.

Unlike the quality assurance problem, the probabilities of success are no longer the same. We need to use information found on the internet to provide the individual probabilities for each of the 16 outcomes. Furthermore, since each minifig has a specific name associated with it, we can use the optional text descriptions to provide the names (Figure 11).

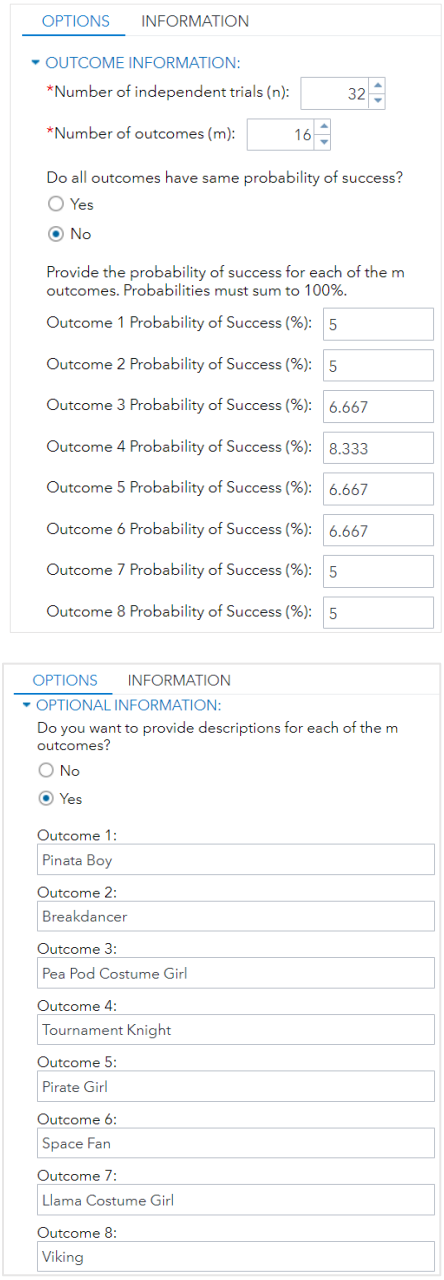
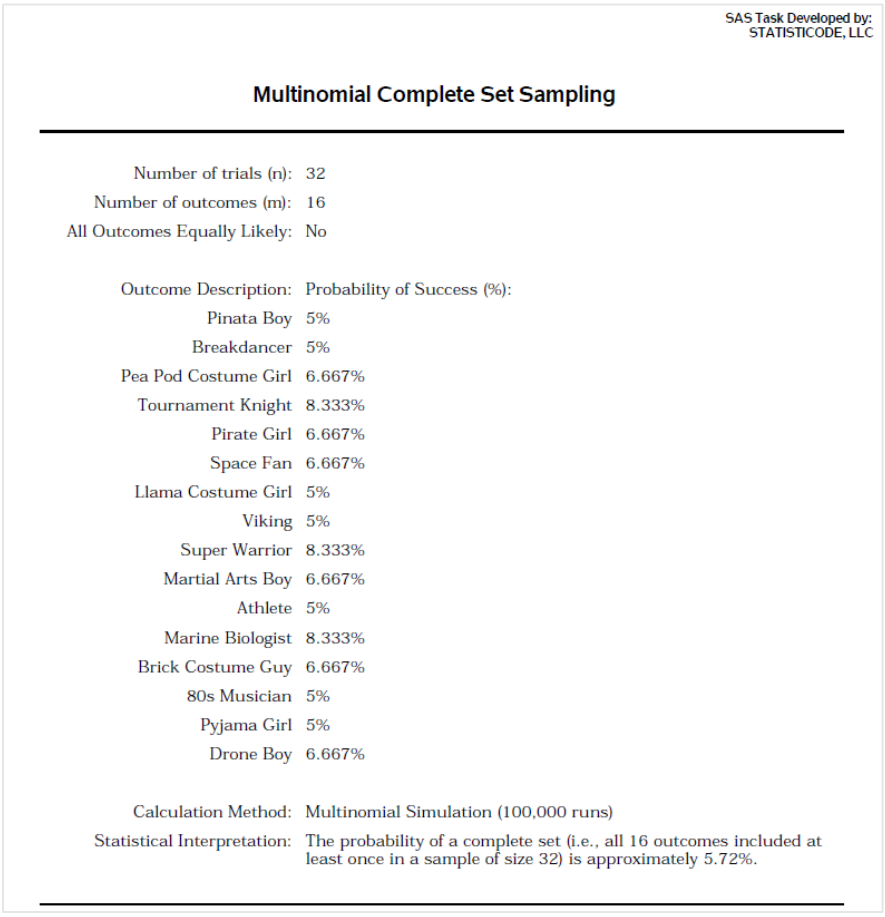


Figure 11. Task with completed options – LEGO (8 of 16 outcomes shown)

Figure 12 displays the PDF report for the LEGO minifigure example. The first two sections of the PDF again summarize the options provided by the user.



SAS Task Developed by:
STATISTICODE, LLC

Multinomial Complete Set Sampling

Number of trials (n):	32
Number of outcomes (m):	16
All Outcomes Equally Likely:	No

Outcome Description:	Probability of Success (%):
Pinata Boy	5%
Breakdancer	5%
Pea Pod Costume Girl	6.667%
Tournament Knight	8.333%
Pirate Girl	6.667%
Space Fan	6.667%
Llama Costume Girl	5%
Viking	5%
Super Warrior	8.333%
Martial Arts Boy	6.667%
Athlete	5%
Marine Biologist	8.333%
Brick Costume Guy	6.667%
80s Musician	5%
Pyjama Girl	5%
Drone Boy	6.667%

Calculation Method:	Multinomial Simulation (100,000 runs)
Statistical Interpretation:	The probability of a complete set (i.e., all 16 outcomes included at least once in a sample of size 32) is approximately 5.72%.

Figure 12. LEGO minifigure solution

The third section displays the statistical analysis method and result. In this example, the calculation method is Multinomial Simulation (100,000 runs). This is because values of $n = 32$ and $m = 16$ result in over 751 billion possible x_i combinations. Needless to say, performing this many exact calculations is not feasible, so simulations are used. A simulated experiment of making 32 selections from each of the 16 outcomes, where each outcome's likelihood of selection is proportional to its specified probability of success, is repeated 100,000 times. After this process completes, the probability of a complete set is approximated, and in this case, there is a 5.72% chance of getting a complete set of 16 minifigs if you purchase 32 mystery bags. This is not very encouraging for collectors. Of course, there are alternative ways of obtaining a complete set, but doing so by trying your luck with mystery bags is so much more fun.

CONCLUSION

Custom SAS Studio Tasks can be very powerful tools, and their functionality and applications are limited only by a developer's imagination and technical skills. Multinomial probability calculations require only a few inputs (i.e., n, m, and p_i), but there are an infinite number of possible results. Whether you are a statistician, programmer, engineer, or AFOL (Adult Fan of LEGO), I hope you see that custom SAS tasks can be developed for all occasions.

REFERENCES

Inman, Elliot, and Wright, Olivia. 2017. “Developing Your Own SAS Studio Custom Tasks for Advanced Analytics.” Proceedings of SAS Global Forum 2017, Orlando, FL. Available at https://support.sas.com/resources/papers/proceedings17/SAS0677-2017.pdf.

SAS Institute Inc. 2016. SAS® Studio 3.6: Developer’s Guide to Writing Custom Tasks. Cary, NC: SAS Institute Inc. Available at https://documentation.sas.com/?docsetId=webeditordg&docsetTarget=titlepage.htm&docsetVersion=3.6&locale=en.

SAS Institute Inc. 2019. SAS® Studio 5.2: Developer’s Guide to Writing Custom Tasks. Cary, NC: SAS Institute Inc. Available at https://documentation.sas.com/?docsetId=webeditordg&docsetTarget=titlepage.htm&docsetVersion=5.2&locale=en.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

John Stephen Taylor

StatistiCode, LLC

(904) 479-1083

jt@statisticode.com

https://www.linkedin.com/in/john-stephen-taylor/