An Efficient Python Approach for Simulation of Poisson Distribution

[1]D.K. Sharma, [2]Bhopendra Singh, [3]Raja M, [4]R. Regin, [5]S. Suman Rajest,
[1]*Department of Mathematics, Jaypee University of Engineering and Technology, M.P., India.*
[2]*Amity University, Dubai.*
[3]*Department of Multimedia, VIT School of Design, Vellore Institute of Technology, Vellore, India.*
[4]*Department of information Technology, Adhiyamaan college of Engineering, Tamil Nadu, India.*
[5]*Vels Institute of Science, Technology & Advanced Studies, Tamil Nadu, India.*
dilipsharmajiet@gmail.com, bsingh@amityuniversity.ae, raja.m@vit.ac.in,
regin12006@yahoo.co.in, sumanrajest414@gmail.com

2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS) | 978-1-6654-0521-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICACCS51430.2021.9441895

***Abstract*—A basic understanding of probability, of its key mathematical features as well as the characteristics it presents within specific circumstances, is the aim of this paper. The action of probability is related to the characteristics of the phenomena that we can forecast. This relation can be described as a distribution of probability. Identify the nature of phenomena (which can also be described by variables), the distribution of probability is defined. The likelihood can be represented by a binomial or Poisson distribution for categorical (or discrete) variables in the majority of cases. For their potential use, distributions of probability are briefly defined along with some examples. The Poisson distribution is a discrete distribution of probabilities that is mostly utilized within a given time span for a model distribution of count data, such as the number of traffic accidents and the number of phone calls received. Each entry starts with a definition and explanation of the Poisson distribution properties, that is followed by a discussion of how to obtain or estimate the Poisson distribution. Finally, every entry provides a discussion of applications that use python libraries for delivery.**

Keywords—Probability distribution; Poisson Distribution; Probability mass function; Numpy; Mathplotlib

I. INTRODUCTION

There are some useful statistical properties available in the Poisson distribution. It is complementary and full; it possesses enough figures. Its exponential structure allows for "exact" conditional tests to be constructed. In developing useful statistical techniques, this paper explores how well these properties can be used. Part I includes the derivation of Poisson conditional types depending on the variance and the cumulants of the samples. In the case of left-truncated samples, the' variance' test is also expanded. Significance checks for cross-product fractions of Poisson averages are obtained in Section II. In biomedical science, all the approaches are explained with realistic examples [12]. The distribution of Poisson[1] is also selected part of many important models for testing biological appearance. It can be used to analyze the number of events that are supposed to occur spontaneously over time, or can occur as a "law of tiny groups" specifically, as just a binomial distribution limit with such a large sample and a small parameter[3].

There are also useful statistical properties of the Poisson distribution. It is additive; which is to say, a Poisson distribution itself has the number of independent Poisson variates [13]. When collecting from some kind of Poisson distribution, the average is a standard adequate estimate. The knowledge that it's full is important for our purposes here. In that all other cumulants are equal to its average, it is exceptional. In the basic system [2,] the Poisson average can be thought of as an exponential function of the associated inputs. Then there's a minimum collection of relevant statistics, and in some cases, "exact" conditional relevance metrics can be calculated [14]. These sophisticated Poisson properties are used to derive certain contingent methods[4] for evaluating count outcomes. The completeness property is often used to extract skewness and kurtosis related moments from the "variance" test statistics and test statistics. A few of these findings can also be applied to the rear situation. We consider exact and proton decay conditional tests for Poisson mean bridge complexes. Many of the methods are explained using important functional examples [15-16].

For 10 years now, people have also been freelancing. The estimated annual income so far has been around $80,000. It's possible to feel like people are stuck in a rut this year and decide to hit 6 figures. To do that, we want to start by measuring the likelihood of this exciting accomplishment occurring, but we don't understand how to do it. There are also situations where an unpredictable occurrence is known to occur and organisations try to determine the uncertainty of the subset occurring larger or smaller than this rate in the future. For example, on special occasions such as Black Fridays or Cyber Mondays, store owners here who know their average sales will try to estimate how much more they would produce. This will help them store more products and handle their workers accordingly. In this post, we will speak about the theory behind the distribution of Poisson which is used to design situations as above, how to recognize and use its formula, and when to use Python code to simulate it. This paper is split into six components. The second element focuses on the discrete probabilitydistributions. In section 3 probability mass function is explained. Poisson distribution along with python is explained in section 4. It considers a simple Poisson distribution, with the test statistic's allocation dependent on the sample mean's data point [5]. In section 5 simulating the Poisson distribution is done. Section 6 concludes the article.

II. DISCRETE PROBABILITY DISTRIBUTIONS

A accurate prediction [6] is a form of process that shows the contribution of different (countable) outcomes, such as 1, 2, 3, and so on. There are two types of statistical allocations: discrete and continuous. In data processing, transform is a discrete term. Statisticians attempting to describe a study's results and probabilities can map observable statistical models from either an information source to generate a likelihood function diagram [7]. A total sample can give rise to a variety of fractal dimension diagram structures. The most popular probability distributions are normal, uniform, binomial, geometric, Poisson, chi-squared, gamma, and beta exponential.

Binomial, Poisson, Bernoulli, and multinomial are by far the most common discrete probability distributions. Inventory management is one field where discrete distribution can be helpful to companies. Analyzing the calculated frequency using a limited amount of available inventory and a limited quality of physical supply will provide a confidence interval that will guide the appropriate inventory assignment to make the best the use lot size.

In the Monte Carlo simulation, discrete distributions [8] can also occur. Simulation of Monte Carlo is a simulation method that, through programmed technology, guess the probabilities of different results. It is used mainly to assist in predicting situations and identifying threats. Outcomes with discrete values can generate discrete allocations for exploration in the Monte Carlo simulation. These distributions were used to quantify risk and trade-offs between different items within consideration. We're going to build some understanding of discrete probability distributions before we get to the paper's actual ingredients.

III. Probability mass function

Let's describe, first, how we mean by discreet. In descriptive statistics, any data that is reported or obtained by numbering is discrete data, i. e. Integers. Integers. Examples are test scores, the number of cars in a car park, that number of hospital births, etc.

There's many random tests, then, that have discrete effects. A coin flip, for instance, has two results: heads and tails (1 and 0), rolling a die has 6 discrete results, and so on. It will have a discrete distribution of probability if a random variable Y is being used to store potential results of a discrete experiment. The distribution of probabilities reports all the potential consequences of a random experiment. Let's construct the distribution of a single coin flip as a trivial example:

$$P(Y = heads) = 0.5, P(Y = tails) = 0.5$$

That was easy. If we want to record the distribution programmatically, it would be in the form of a Python list or Numpy array:

$$Y = [0.5,0.5]$$

We may imagine, though, that constructing the distribution and finding the probabilities in this way is impractical for massive experiments that have several potential outcomes. Thankfully, each probability distribution that has ever been invented arrives with its own formula to measure the probability of any result. These functions are called Probability Mass Functions to discrete probability distributions (PMF).

IV. Poisson distribution with python

With the aid of a case study, we can begin to comprehend the Stochastic process. Let's just say we get a kick out of seeing new-born babies in a hospital. According to our findings and records, the hospital averages 6 new-born babies [9] per hour.

Fig.1 New-born baby

We know we should go on a business trip tomorrow, so we want to visit the hospital for the last time before we leave for the airport. We want to see as many new children as possible because we're going to be gone for months, so we're curious about the possibility of seeing 10 or more babies an hour before our flight. The results will adopt a classic Poisson distribution if we contemplate studying new-born babies as a arbitrary assessment. The explanation is that it holds all the requirements appropriate for a Poisson distribution:

- There is a known rate of events: 6 new babies every hour on average
- Events occur independently: 1 baby being born does not affect the timing of the next
- The known rate is constant over time: the average number of babies per hour does not change over time
- Two events do not happen at exactly the same instant
- (Reminder: each outcome is discrete)

There are several major market consequences of Poisson distribution. Companies also use it to predict the amount of transactions or customers that occur on a specific day, provided that they know the average daily rate. Making such predictions allows companies to make higher performance, scheduling, or personnel decisions. Overstocking, for instance, suggests losses due to low sales operation or not providing enough products means losing business opportunity. In short, the distribution of Poisson allows to discover in a set time interval the uncertainty of an occurrence occurring larger or smaller than the already reported outlay (often noted as λ (lambda)).

This formula provides its Probability Mass Function:

$$g(k, \lambda) = P(Y = k) = \frac{\lambda^k e^{-\lambda}}{k!} \qquad (1)$$

where
k is the number of successes (the number of times a desired even happening)
λ is the given rate

e is Euler's number: e = 2.71828…
k! is the factorial of *k*

Using this formula, we can find the probability of seeing 10 new-born babies knowing that the average rate is 6. Hence from (1) we have

$$g(k = 10, \lambda = 6) = P(Y = 10) = \frac{6^{10}e^{-6}}{10!} = 0.0413$$

Unfortunately, there is only ~4% percent chance of seeing 10 babies.

There are still some points we have to keep in mind. Even though there is a known rate, it is just an average, so the timing of the events can be completely random. For example, it can be observed 2 babies born back-to-back or we may end up waiting for half an hour for the next. Also, in practice, the rate λ may not always be constant. This can even be true for our new-born babies experiment. Although this aspect disappears, we can indeed allow the approximation to be Poisson since it is similar by to the actions of the case.

V. SIMULATING THE POISSON DISTRIBUTION

Simulating or drawing samples from Poisson distribution is very easy using numpy [10]. We first import it and use its random module for simulation:

```
import numpy as np
```

To draw samples from a Poisson distribution, we only need the rate parameter *λ*. We will plug it into np.random.poisson function and specify the number of samples:

```
poisson = np.random.poisson(lam=10, size=10000)
```

Here, we are simulating a distribution with a rate of 10 and has 10k data points. To see this distribution, we will plot the results of its PMF. Though we could do it by hand, there is already a very good library called empiricaldist, written by Allen B. Downey - author of well-known books such as *ThinkPython* and *ThinkStats*. We will install and import its Pmffunction into our environment:

```
from empiricaldist import Pmf # pip install empiricaldist
```

Pmf has a function called from_seq which takes any distribution and computes the PMF:

```
poisson = np.random.poisson(lam=10, size=10000)pmf_poisson = Pmf.from_seq(poisson)
pmf_poisson
```

TABLE I. PROBABILITES OF OUTCOMES

Outcomes	Probability(Probs)
1	0.0009
2	0.0027
3	0.0077
4	0.0203
5	0.0424
6	0.0609
7	0.0902
8	0.1086
9	0.1274
10	0.1305
11	0.1092
12	0.0979
13	0.0719
14	0.0457
15	0.0353
16	0.0224
17	0.0124
18	0.0072
19	0.0039
20	0.0015
21	0.0003
22	0.0005
23	0.0002

Recall that PMF shows the probabilities of each unique outcome, so in the above result, the outcomes are given as index and probabilities under probs. Let's plot it using matplotlib [11].

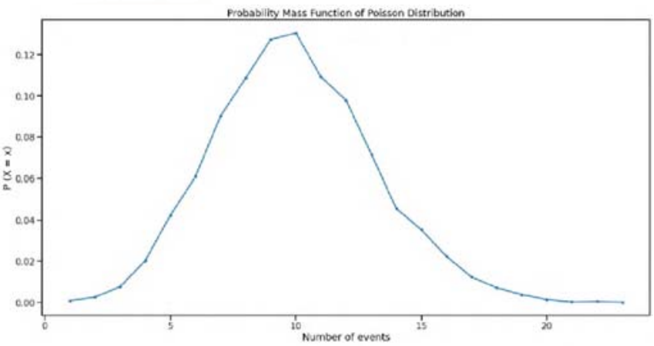

Fig-2 PMF of Poisson Distribution

As expected, the highest probability is for the mean (rate parameter, λ).

Now, let's say we have forgotten the Poisson distribution formula for the PMF. How will we find the chance of seeing 10 new babies with a rate of 6 if we were doing our experiment of observing new-born babies?

Well, to start off, using ours given rate as a parameter, lets simulate the perfect Poisson distribution. Even, for better precision, we make sure to draw a lot of samples:

With a rate of 6 and a duration of 1 million, we sample a distribution. Next, we'll see how many of them have ten babies:

So, in 4,1114 trials, we detected 10 children (each hour can be considered to have one trial). We divide this sum, then, by the total number of samples:

```
>>> births_10 / 1e60.041114
```

If we recall, using the PMF formula, the result was 0.0413 and we can see that our hand-coded solution is a pretty close match.

VI. CONCLUSION

In this paper, the feature of probability mass was obtained from observations and the time of high probability was considered. In 75 percent of the cases, the Poisson distribution, including its probability mass function, gives the best fit. The effectiveness of new distributions shows that the effect on the hosting ability of the distribution function used is limited but still comparable to the influence. Therefore, any distribution of probabilities can be used with the aid of python libraries. Until counterexamples are found, the basic libraries such as numpy and mathplotlib are common enough to be able to generalise this finding. On Poisson distribution, there is always plenty to be said. In the corporate world, we discussed the fundamental use and its consequences. Sections of the Poisson distribution are also important, such as how it applies to both the distribution of Binomial.

References

[1] C.Y. Luo, J.J. Xiong and R.A. Shenoi, "A reliability-based approach to assess fatigue behavior based on small incomplete data sets", *Advanced materials research*, vol. 44–46, pp. 871-978, 2008.

[2] Hasting, N. A. J., & Peacock, J. B. (1974). Statistical distributions. New York: Wiley & Sons.

[3] Hines, W. W., Montgomery, L. D. C., Goldsman, D. M., & Burror, C. M. (2003). Probability and statistics for engineers. New York: Wiley & Sons.

[4] H. An, W. G. An, Y. L. Zhao and Z. J. Song, "A method of fuzzy probability under small sample", *Advanced materials research*, vol. 44–46, pp. 879-884, 2008.

[5] K. Fujiyama, "Application of the unified statistical material database for design and life/risk assessment of high temperature components", *Fifth China-Japan bilateral symposium on high temperature strength of materials*, pp. 297-304, 2004.

[6] Law, A. M., & Kelton, W. D. (2000). Simulation modeling and analysis. Boston: McGraw Hill.

[7] H. Anandakumar and K. Umamaheswari, "Supervised machine learning techniques in cognitive radio networks during cooperative spectrum handovers," Cluster Computing, vol. 20, no. 2, pp. 1505–1515, Mar. 2017.

[8] H. Anandakumar and K. Umamaheswari, "A bio-inspired swarm intelligence technique for social aware cognitive radio handovers," Computers & Electrical Engineering, vol. 71, pp. 925–937, Oct. 2018. doi:10.1016/j.compeleceng.2017.09.016

[9] https://pharmeasy.in/blog/newborn-care-week-how-to-care-for-a-newborn-baby/

[10] https://plot.ly/python/

[11] https://matplotlib.org/

[12] Avijit Mondal, Radha Tamal Goswami, "Enhanced Honeypot cryptographic scheme and privacy preservation for an effective prediction in cloud security, Microprocessors and Microsystems, Volume 81, 2021,103719, https://doi.org/10.1016/j.micpro.2020.103719.

[13] D Datta, S Mishra, SS Rajest, (2020) "Quantification of tolerance limits of engineering system using uncertainty modeling for sustainable energy" International Journal of Intelligent Networks, Vol.1, 2020, pp.1-8, https://doi.org/10.1016/j.ijin.2020.05.006

[14] Leo Willyanto Santoso, Bhopendra Singh, S. Suman Rajest, R. Regin, Karrar Hameed Kadhim (2020), "A Genetic Programming Approach to Binary Classification Problem" EAI Endorsed Transactions on Energy, Vol.8, no. 31, pp. 1-8. DOI: 10.4108/eai.13-7-2018.165523

[15] Bhopendra Singh, S. Suman Rajest, K. Praghash, Uppalapati Srilakshmi and R. Regin (2020) Nuclear structure of some even and odd nuclei using shell model calculations. Proceedings of the 2020 2nd International Conference on Sustainable Manufacturing, Materials and Technologies. AIP Conference Proceedings, 2020, https://aip.scitation.org/doi/abs/10.1063/5.0030932

[16] Avijit Mondal, Arnab Kumar Das, Sayan Nath, Radha Tamal Goswami,"Review Study on Different Attack Strategies of Worm in a Network", Webology, Volume 17, Number 2, December, 2020.