Essential Math for Data Science: The Poisson Distribution

<= Previous post
Next post =>

Tags: Data Science, Distribution, Mathematics, Poisson Distribution

The Poisson distribution, named after the French mathematician Denis Simon Poisson, is a discrete distribution function describing the probability that an event will occur a certain number of times in a fixed time (or space) interval.

SAS Talent Development

comments

By **Hadrien Jean**, **Machine Learning Scientist**

FREE SAMPLE
HALIOTIS PUBLISHING
Essential Math for Data Science
CALCULUS, STATISTICS, PROBABILITY THEORY AND LINEAR ALGEBRA
Hadrien Jean

The *Poisson distribution*, named after the French mathematician Denis Simon Poisson, is a discrete distribution function describing the probability that an event will occur a certain number of times in a fixed time (or space) interval. It is used to model count-based data, like the number of emails arriving in your mailbox in one hour or the number of customers walking into a shop in one day, for instance.

Mathematical Definition

Let's start with an example, Figure 1 shows the number of emails received by Sarah in intervals of one hour.

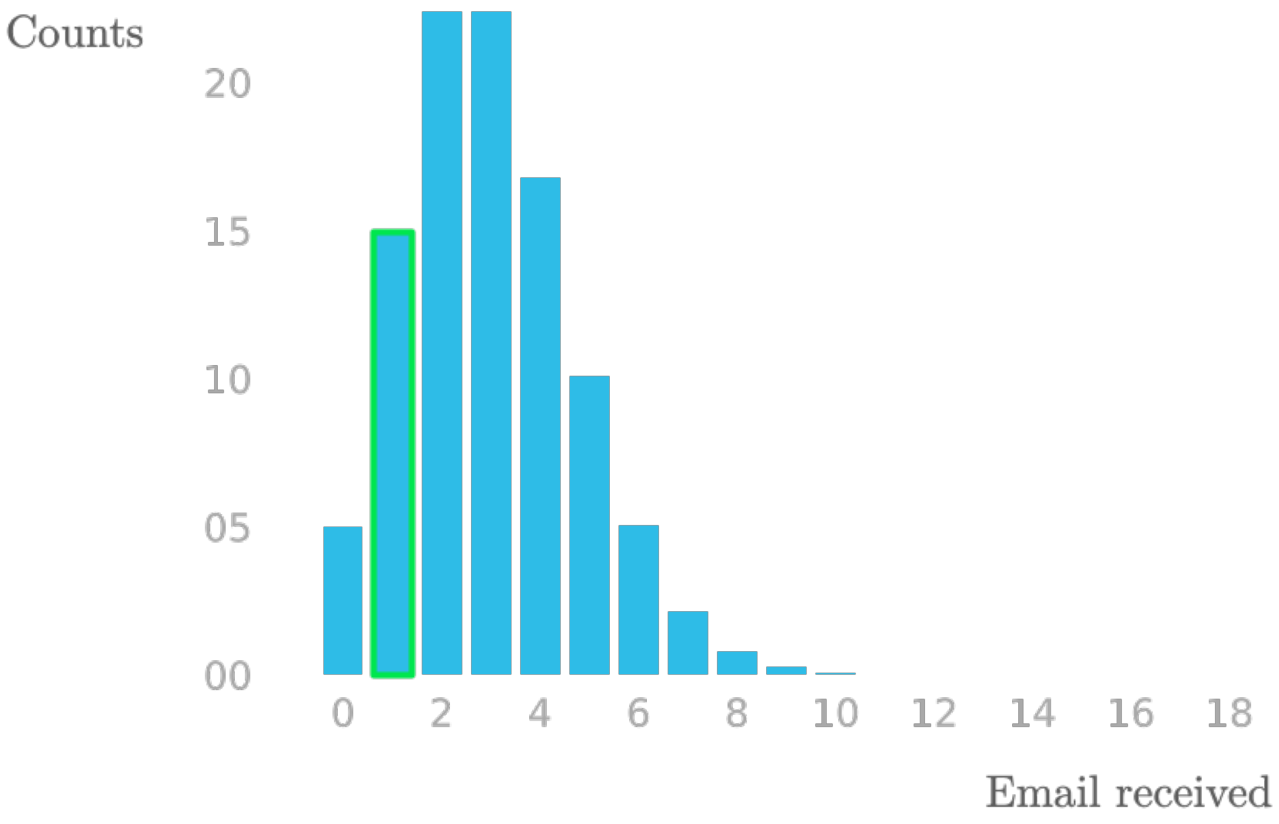


Figure 1: Emails received by Sarah in one-hour intervals for the last 100 hours.

SHARES

The bar heights show the number of one-hour intervals in which Sarah observed the corresponding number of emails. For instance, the highlighted bar shows that there were around 15 one-hour slots where she received a single email.

The Poisson distribution is parametrized by the expected number of events λ (pronounced “lambda”) in a time or space window. The distribution is a function that takes the number of occurrences of the event as input (the integer called k in the next formula) and outputs the corresponding probability (the probability that there are k events occurring).

The Poisson distribution, denoted as *Poi* is expressed as follows:

$$\mathrm{Poi}(\mathbf{k};\boldsymbol{\lambda}) = \frac{\boldsymbol{\lambda}^{\mathbf{k}} \mathbf{e}^{-\boldsymbol{\lambda}}}{\mathbf{k}!}$$

for $k = 0, 1, 2, \ldots$

The formula of $Poi(k; \lambda)$ returns the probability of observing k events given the parameter λ which corresponds to the expected number of occurrences in that time slot.

> **Discrete Distributions**
>
> Note that both the binomial and the Poisson distributions are discrete: they give probabilities of discrete outcomes: the number of times an event occurs for the Poisson distribution and the number of successes for the binomial distribution. However, while the binomial calculates this discrete number for a discrete number of trials (like a number of coin toss), the Poisson considers an infinite number of trials (each trial corresponds to a very small portion of time) leading to a very small probability associated with each event.

You can refer to the section below to see how the Poisson distribution is derived from the binomial distribution.

Example

Priya is recording birds in a national park, using a microphone placed in a tree. She is counting the number of times a bird is recorded singing and wants to model the number of birds singing in a minute. For this task, she’ll assume independence of the detected birds.

Looking at the data of the last few hours, Priya observes that in average, there are two birds detected in an interval of one minute. So the value 2 could be a good candidate for the parameter of the distribution λ. Her goal is to know the probability that a specific number of birds will sing in the next minute.

Let’s implement the Poisson distribution function from the formula you saw above:

```
def poisson_distribution(k, lambd):
    return (lambd ** k * np.exp(-lambd)) / np.math.factorial(k)
```

Remember that λ is the expected number of times a bird sings in a one-minute interval, so in this example, you have $\lambda=2$. The function `poisson_distribution(k, lambd)` takes the value of k and λ and returns the probability to observe k occurrences (that is, to record k birds singing).

For instance, the probability of Priya observing 5 birds in the next minute would be:

```
poisson_distribution(k=5, lambd=2)
```

```
0.03608940886309672
```

The probability that 5 birds will sing in the next minute is around 0.036 (3.6%).

As with the binomial function, this will overflow for larger values of k. For this reason, you might want to use `poisson` from the module `scipy.stats`, as follows:

```
from scipy.stats import poisson

poisson.pmf(5, 2)
```

SHARES

```
0.03608940886309672
```

Let’s plot the distribution for various values of k:

```
lambd=2

k_axis = np.arange(0, 25)
distribution = np.zeros(k_axis.shape[0])
for i in range(k_axis.shape[0]):
    distribution[i] = poisson.pmf(i, lambd)

plt.bar(k_axis, distribution)
# [...] Add axes, labels...
```

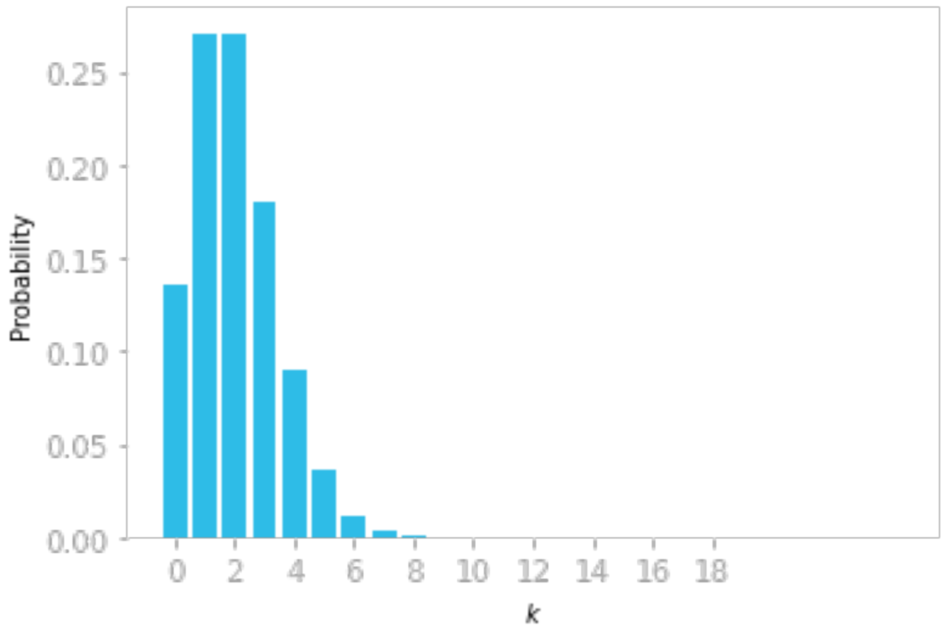


Figure 2: Poisson distribution for λ=2.

The probabilities corresponding to the values of k are summarized in the probability mass function shown in Figure

1. You can see that it is most probable that Priya will hear one or two birds singing in the next minute.

Finally, you can plot the function for different values of $\lambda$:

```
f, axes = plt.subplots(6, figsize=(6, 8), sharex=True)

for lambd in range(1, 7):

    k_axis = np.arange(0, 20)
    distribution = np.zeros(k_axis.shape[0])
    for i in range(k_axis.shape[0]):
        distribution[i] = poisson.pmf(i, lambd)

    axes[lambd-1].bar(k_axis, distribution)
    axes[lambd-1].set_xticks(np.arange(0, 20, 2))
    axes[lambd-1].set_title(f"$\lambda$: {lambd}")

# Add axes labels etc.
```

SHARES

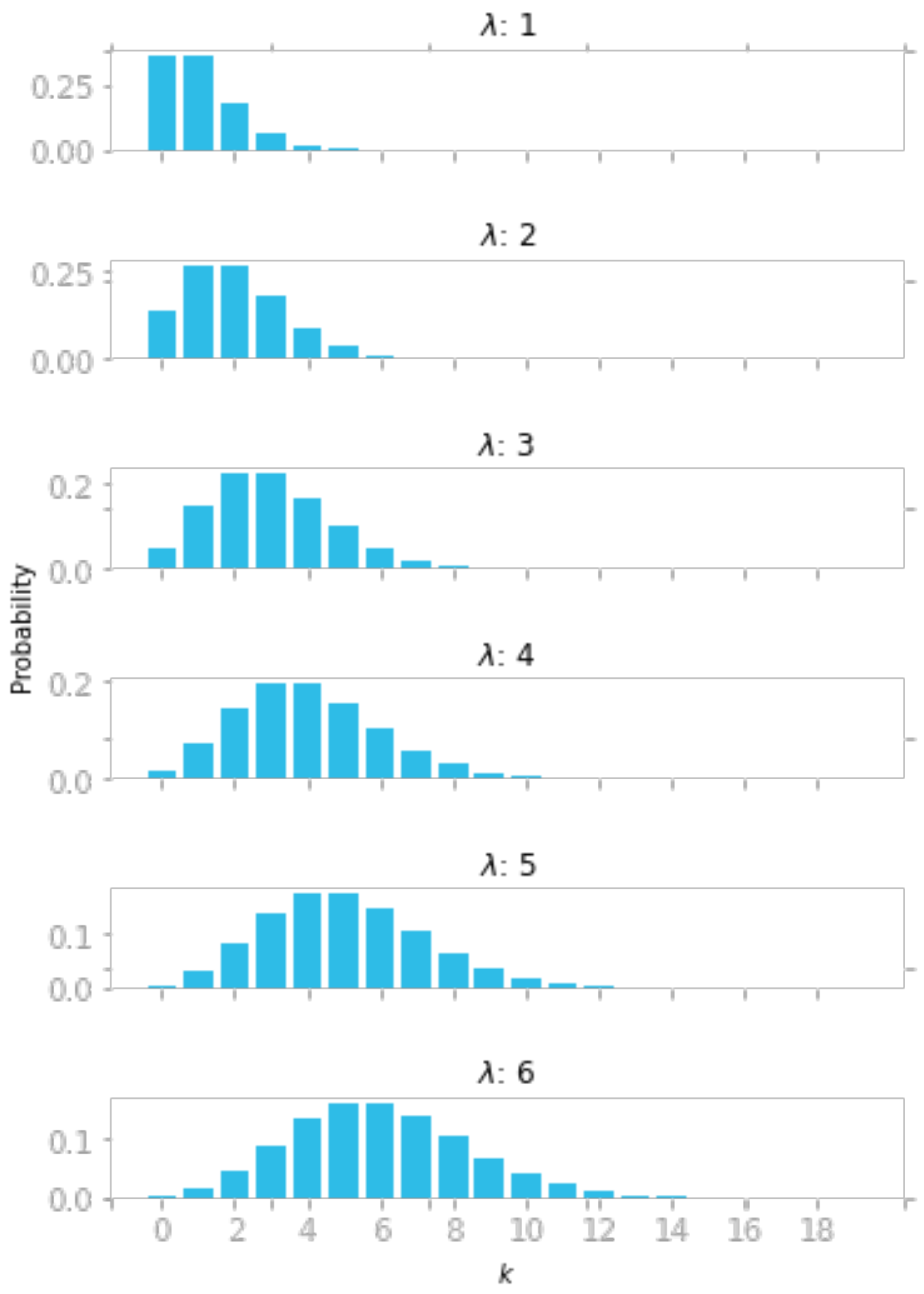


Figure 3: Poisson distribution for various values of λ.

Figure 3 shows the Poisson distribution for various values of λ, which looks a bit like a normal distribution in some cases. However, the Poisson distribution is discrete, not symmetric when the value of λ is low, and bounded to zero.

Bonus: Deriving the Poisson Distribution

Let's see how the Poisson distribution is derived from the Binomial distribution.

You saw in Essential Math for Data Science that if you run a random experiment multiple times, the probability to get mm successes over N trials, with a probability of a success μ at each trial, is calculated through the binomial distribution:

$$\mathrm{Bin}(\mathbf{k}; \mathbf{N}, \boldsymbol{\mu}) = \binom{\mathbf{N}}{\mathbf{k}} \boldsymbol{\mu}^{\mathbf{k}} (1 - \boldsymbol{\mu})^{\mathbf{N}-\mathbf{k}}$$

Problem Statement

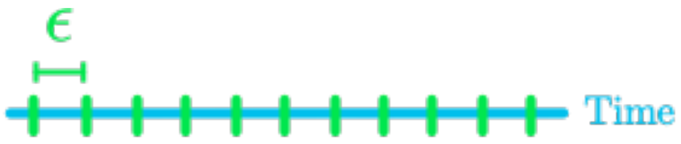
SHARES

How can you use the binomial formula to model the probability to observe an event a certain number of times *in a given time interval* instead of in a certain number of trials? There are a few problems:

1. You don't know NN, since there is no specific number of trials, only a time window.
2. You don't know μμ, but you have the expected number of times the event will occur. For instance, you know that in the past 100 hours, you received an average of 3 emails per hour, and you want to know the probability of receiving 5 emails in the next hour.

Let's handle these issues mathematically.

To address the first point, you can consider time as small discrete chunks. Let's call these chunck ϵϵ (pronounced "epsilon"), as shown in Figure 4. If you consider each chunk as a trial, you have *N* chunks.



>Figure 4: You can split the continuous time in segments of length ϵ.

The estimation of a continuous time scale is more accurate when ϵ is very small. If ϵϵ is small, the number of segments *N* will be large. In addition, since the segments are small, the probability of success in each segment is also small.

To summarize, you want to modify the binomial distribution to be able model a very large number of trials, each with a very small probability of success. The trick is to consider that *N* tends toward infinity (because continuous time is approximated by having a value of ϵ that tends toward zero).

Update the Binomial Formula

Let's find μμ in this case and replace it in the binomial formula. You know the expected number of event in a period of time *t*, which we'll call λ (pronounced "lambda"). Since you split *t* into small intervals of length ϵ, you have the number of trials:

$$\mathbf{N = t \cdot \epsilon}$$

You have λ as the number of successes in the *N* trials. So the probability μ to have a success in one trial is:

$$\mu = \frac{\boldsymbol{\lambda}}{\mathbf{N}}$$

Replacing μμ in the binomial formula, you get:

$$\binom{\mathbf{N}}{\mathbf{k}}\left(\frac{\boldsymbol{\lambda}}{\mathbf{N}}\right)^{\mathbf{k}}\left(1-\frac{\boldsymbol{\lambda}}{\mathbf{N}}\right)^{\mathbf{N-k}}$$

Developing the expression, writing the binomial coefficient as factorials (as you did in Essential Math for Data Science), and using the fact $\mathbf{a^{b-c} = a^b - a^c}$, you have:

$$\frac{\mathbf{N!}}{\mathbf{(N-k)!k!}}\left(\frac{\boldsymbol{\lambda}}{\mathbf{N}}\right)^{\mathbf{k}}\left(1-\frac{\boldsymbol{\lambda}}{\mathbf{N}}\right)^{\mathbf{N}}\left(1-\frac{\boldsymbol{\lambda}}{\mathbf{N}}\right)^{\mathbf{-k}}$$

Let's consider the first element of this expression. If you state that NN tends toward infinity (because ϵ tends toward zero), you have:

$$\lim_{\mathbf{N}\to+\infty}\frac{\mathbf{N!}}{\mathbf{(N-k)!}} = \mathbf{N^k}$$

This is because *k* can be ignored when it is small in comparison to *N*. For instance, you have:

$$\frac{\mathbf{1,000,000!}}{\mathbf{(1,000,000-3)!}} = \mathbf{1,000,000 \cdot 999,999 \cdot 999,998}$$

which approximates $\mathbf{1,000,000 \cdot 1,000,000 \cdot 1,000,000}$

So you the first ratio becomes:

$$\frac{\mathbf{N!}}{\mathbf{(N-k)!k!}} = \mathbf{N^k} \cdot \frac{\mathbf{1}}{\mathbf{k!}}$$

SHARES

Then, using the fact that $\lim_{N\to+\infty}(1+\frac{\lambda}{N})^N = e^\lambda$, you have:

$$\lim_{N\to+\infty}\left(1-\frac{\lambda}{N}\right)^N = e^{-\lambda}$$

Finally, since $\left(1-\frac{\lambda}{N}\right)$ tends toward 1 when *N* tends toward the infinity:

$$\lim_{N\to+\infty}\left(1-\frac{\lambda}{N}\right)^{-k} = 1^{-k} = 1$$

Let’s replace all of this in the formula of the binomial distribution:

$$\lim_{N\to+\infty}\frac{N!}{(N-k)!k!}\left(\frac{\lambda}{N}\right)^k\left(1-\frac{\lambda}{N}\right)^N\left(1-\frac{\lambda}{N}\right)^{-k}$$

$$=\lim_{N\to+\infty}N^k\cdot\frac{1}{k!}\cdot\left(\frac{\lambda}{N}\right)^k\cdot e^{-\lambda}\cdot 1$$

$$=\lim_{N\to+\infty}\frac{1}{k!}\cdot\left(N\frac{\lambda}{N}\right)^k\cdot e^{-\lambda}$$

$$=\frac{1}{k!}\cdot\lambda^k\cdot e^{-\lambda}$$

$$=\frac{\lambda^k e^{-\lambda}}{k!}$$

This is the Poisson distribution, denoted as *Poi*:

$$\mathrm{Poi}(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$$

for *k = 0, 1, 2, ...*

Bio: Hadrien Jean is a machine learning scientist. He owns a Ph.D in cognitive science from the Ecole Normale Superieure, Paris, where he did research on auditory perception using behavioral and electrophysiological data. He previously worked in industry where he built deep learning pipelines for speech processing. At the corner of data science and environment, he works on projects about biodiversity assessement using deep learning applied to audio recordings. He also periodically creates content and teaches at Le Wagon (data science Bootcamp), and writes articles in his blog (hadrienj.github.io).

Original. Reposted with permission.

Related:

- Essential Math for Data Science: Probability Density and Probability Mass Functions
- Essential Math for Data Science: Integrals And Area Under The Curve
- Free Mathematics Courses for Data Science & Machine Learning

SHARES

[<= Previous post]

[Next post =>]

Top Stories Past 30 Days

Most Popular

1. We Don't Need Data Scientists, We Need Data Engineers
2. Are You Still Using Pandas to Process Big Data in 2021? Here are two better options
3. Telling a Great Data Story: A Visualization Decision Tree
4. Top YouTube Channels for Data Science
5. 3 Mathematical Laws Data Scientists Need To Know

Most Shared

1. We Don't Need Data Scientists, We Need Data Engineers
2. Telling a Great Data Story: A Visualization Decision Tree
3. Top YouTube Channels for Data Science
4. Evaluating Deep Learning Models: The Confusion Matrix, Accuracy, Precision, and Recall
5. 3 Mathematical Laws Data Scientists Need To Know

Latest News

- Sudoku Rules: Using A Decision Engine To Solve Sudoku
- Data Community Job Satisfaction Survey
- 10 Amazing Machine Learning Projects of 2020
- Forget Telling Stories; Help People Navigate
- Top Stories, Mar 8-14: How To Overcome The Fear of Math...
- AI Industry Innovation: Making the Invisible Visible

**Top Stories
Last Week**

Most Popular

1. How To Overcome The Fear of Math and Learn Math For Data Science
2. Know your data much faster with the new Sweetviz Python library
3. Must Know for Data Scientists and Data Analysts: Causal Design Patterns
4. Are You Still Using Pandas to Process Big Data in 2021? Here are two

SHARES

5. **3 Mathematical Laws Data Scientists Need To Know**

Most Shared

1. **How To Overcome The Fear of Math and Learn Math For Data Science**
2. **A Machine Learning Model Monitoring Checklist: 7 Things to Track**
3. **Must Know for Data Scientists and Data Analysts: Causal Design Patterns**
4. **Know your data much faster with the new Sweetviz Python library**
5. **4 Machine Learning Concepts I Wish I Knew When I Built My First Model**

More Recent Stories

- AI Industry Innovation: Making the Invisible Visible
- Kedro-Airflow: Orchestrating Kedro Pipelines with Airflow
- Must Know for Data Scientists and Data Analysts: Causal Design...
- Know your data much faster with the new Sweetviz Python library **[Silver Blog]**
- Top February Stories: We Don't Need Data Scientists, We ...
- Advance your career in Data Science with HSE Master in Data Sc...
- A Beginner's Guide to the CLIP Model
- The Inferential Statistics Data Scientists Should Know
- A Machine Learning Model Monitoring Checklist: 7 Things to Track **[Silver Blog]**
- Read This Before You Apply to a Business Analytics Master...
- How to Speed Up Pandas with Modin
- How To Overcome The Fear of Math and Learn Math For Data Science **[Gold Blog]**
- DeepMind's AlphaFold & the Protein Folding Problem
- KDnuggets 21:n10, Mar 10: More Resources for Women in AI, D...
- A Solid Investment: Banking on Talent Development.
- Understanding NoSQL Document Databases
- 4 Machine Learning Concepts I Wish I Knew When I Built My Firs...
- Is It Too Late to Learn AI?
- 8 Women in AI Who Are Striving to Humanize the World
- Top Stories, Mar 1-7: Top YouTube Channels for Data Science

KDnuggets Home » News » 2020 » Dec » Tutorials, Overviews » Essential Math for Data Science: The Poisson Distribution (20:n48)

Subscribe to KDnuggets News
X

SHARES