NNSE 784
Advanced Analytics Methods

Instructor: F Doyle (CESTM L210)

MW 4:30 – 5:50, NFN 203

Slide Set #5
Probability Distributions
(Discrete Distributions)

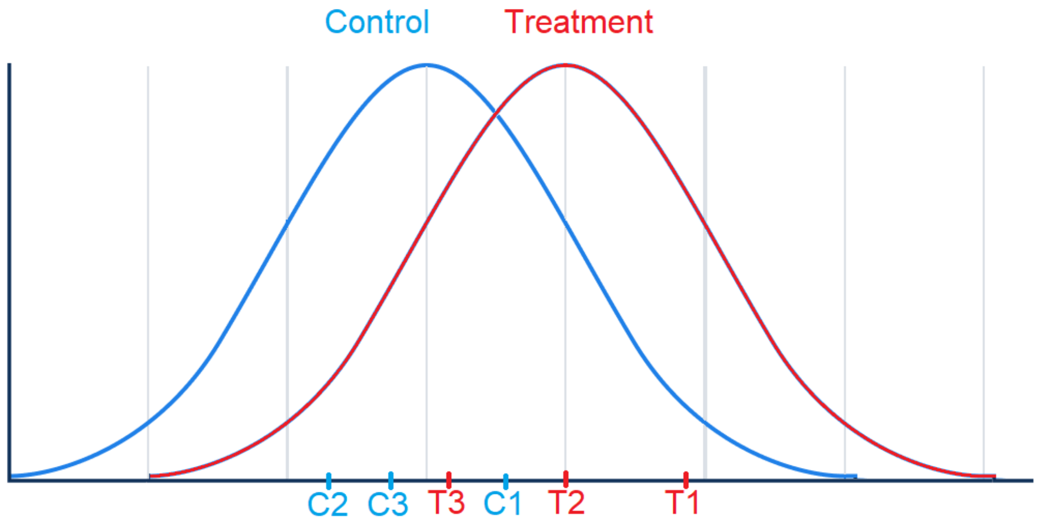
Recap - summary

- Descriptive statistics
  - Frequency plots (histograms)
  - Measures of central tendency
  - Measures of dispersion
- Probability
  - Marginal and joint probability – single event vs combinations
  - Independent vs dependent events – does the outcome of one event impact the other?
  - Conditional probability
    - Bayes’ Theorem
- Complexity – the scope of combinations/permutations and the calculation of probabilities are often not intuitive. Don’t “trust your gut”. Trust the numbers!

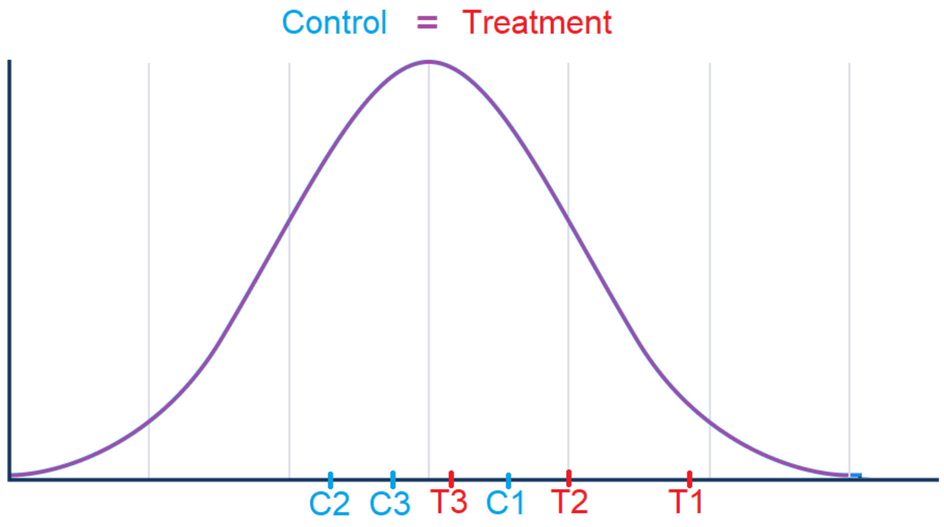
Numerous pilots have crashed into the ground because they believed their senses over their instruments.

Recap – moving toward inferential statistics

Understanding probability allows us to extend descriptive statistics and make **inferences** based on limited ("experimental") data.



VS



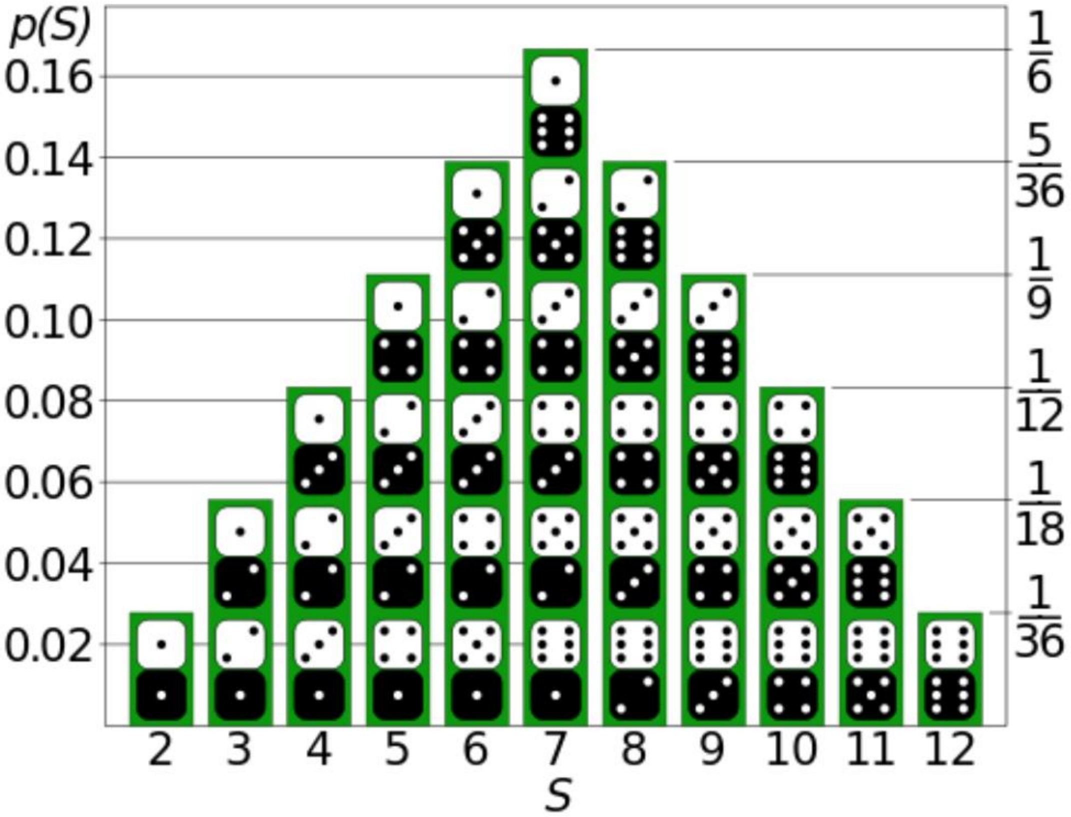
Given experimental data points C1,C2,C3 (controls) and T1,T2,T3 (treatments), both of the above scenarios are possible. **<u>Inferential statistics allows us to determine the probabilities of each.</u>**

This Lecture:
Probability Distributions for Discrete Variables

- Cumulative probability
- Binomial Distribution
- Poisson Distribution

What is a “probability distribution”?

Consider the population of all the possible outcomes when rolling two dice.
How probable is each sum “S” of counts from the two dice?



The probability distribution provides the probability of occurrence of all possible outcomes in an experiment.

Probability distribution of a population

It is generally **not known**. However, we may have sufficient information about this distribution to meet the goals of our research.

Statistical modeling: It is possible to rely on a small set of probability distributions that capture the key characteristics of a population.

We can then **infer the parameters** of these model probability distributions from our sample.

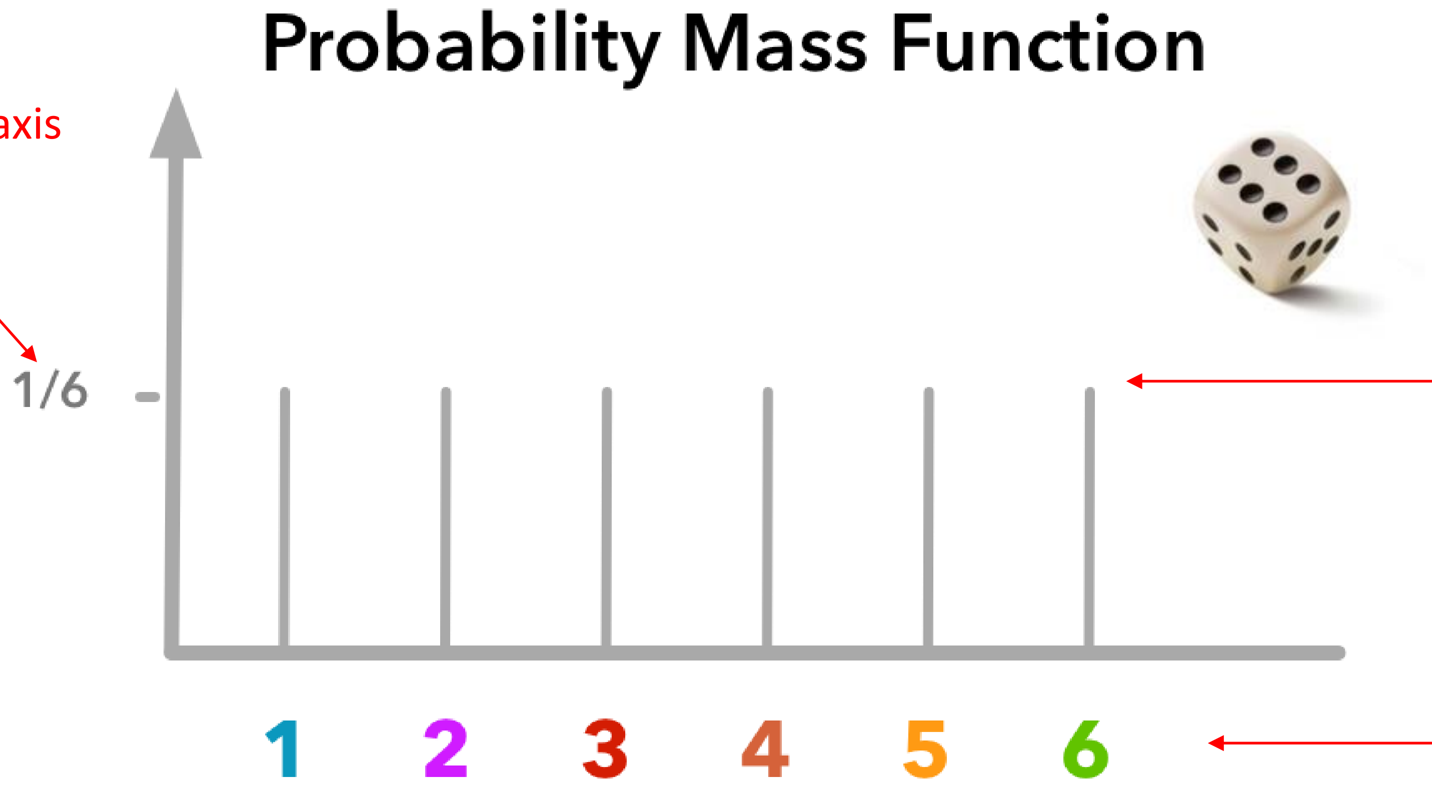
> How? We expect that the sample contains some information about the population from which it was sampled.

Probability Distribution Types

- **Probability Mass Function** - for discrete variables (e.g., die roll results)

- **Probability Density Function** – for continuous variables (e.g., height of a randomly chosen person)

Probability on y axis



Example of a uniform distribution

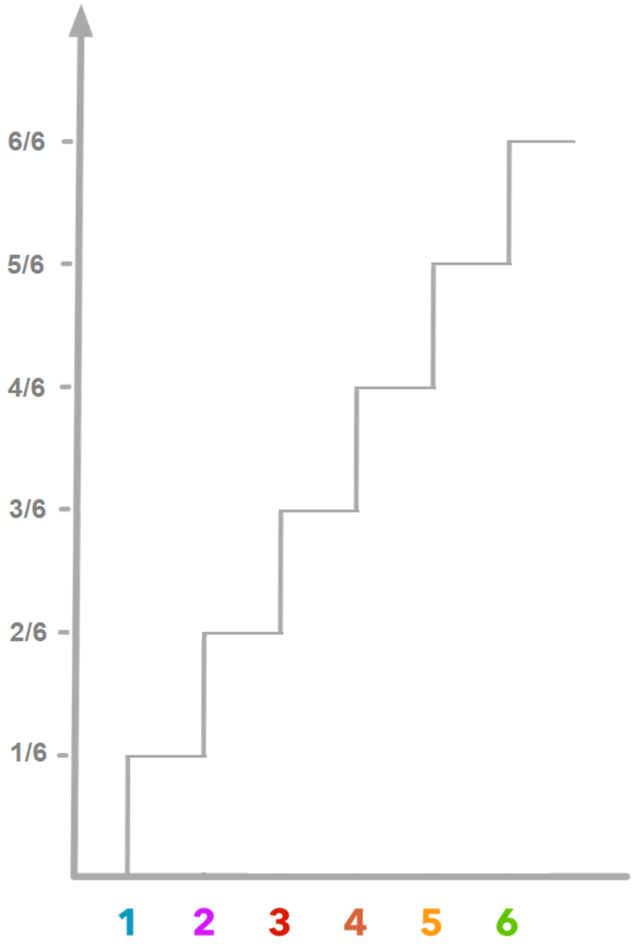
Possible outcomes on x axis

Remember each outcome had same probability: P(y=1) = P(y=2) = P(y=3) = P(y=4) = P(y=5) = P(y=6) = $\frac{1}{6}$

Cumulative Distribution Function

What is the probability of rolling a particular number or lower?

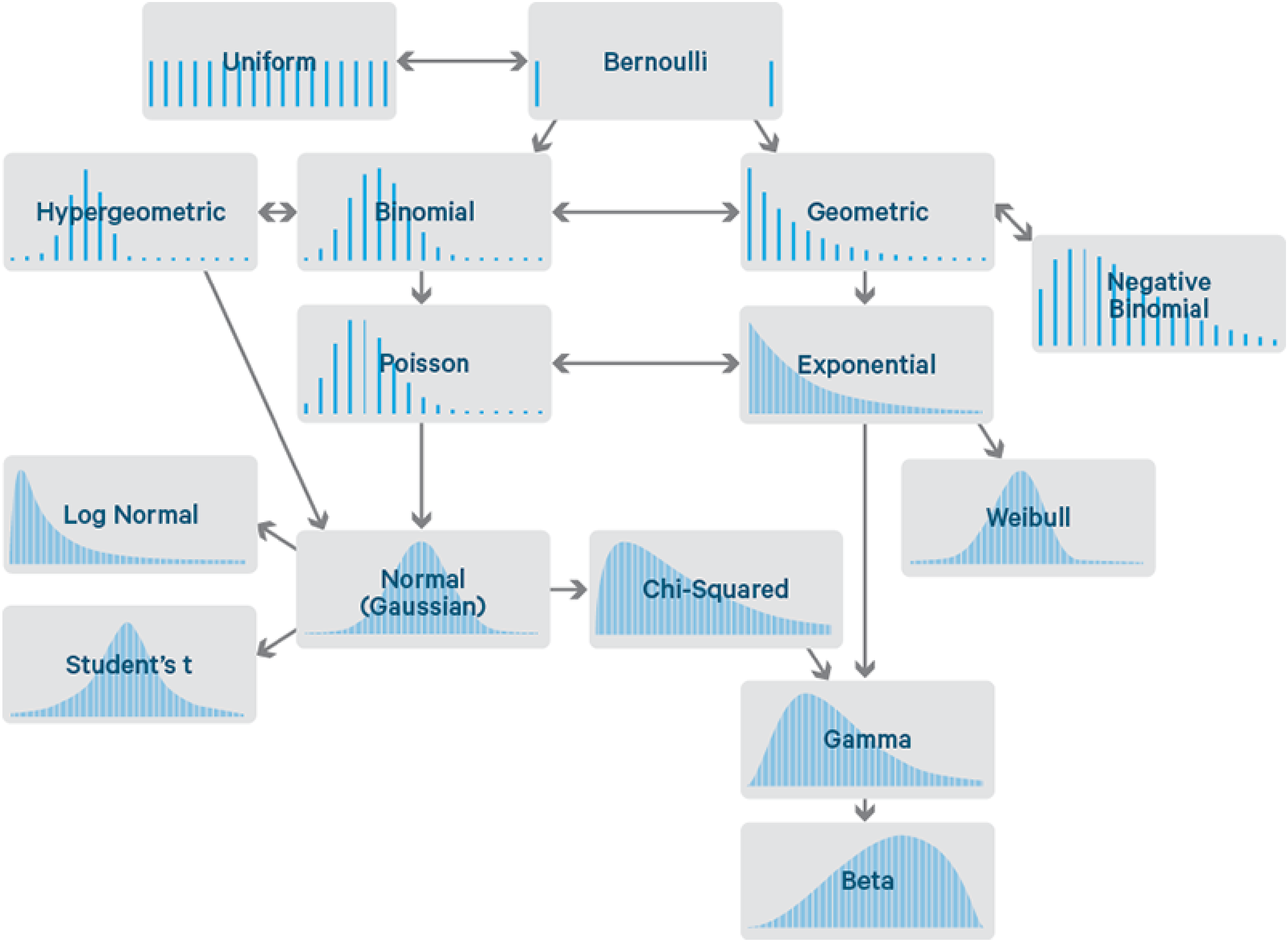
Probability on y axis



Remember each **individual** outcome had same probability:
P(y=1) = P(y=2) = P(y=3) = P(y=4) = P(y=5) = P(y=6) = $\frac{1}{6}$

Also…the probabilities of all possible outcomes must sum to 1

Common Distributions in Statistics



Binomial Distribution

CHARACTERISTICS OF BINOMIAL DISTRIBUTION

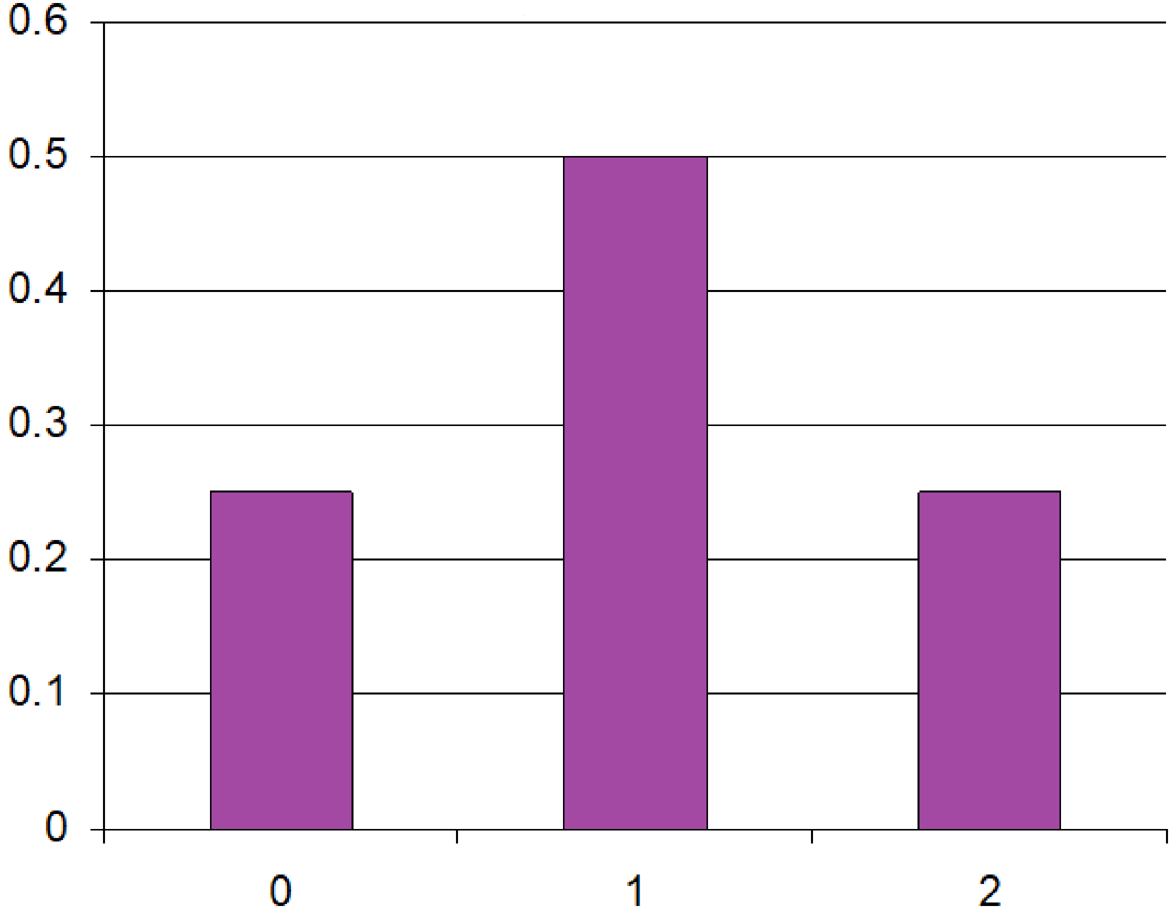
- One of the most widely encountered distributions in applied statistics
- Derived from “Bernoulli Process”, which is a sequence of “Bernoulli trials”
- A Bernoulli trial is a test with only two possible outcomes (e.g., “success” or “failure”, “heads” or “tails”, etc.)
- The probability of a ‘success’ is given as ***p***. The probability of failure (or “alternate event”) is equal to **1-*p*** and denoted by ***q****.*
- Each trial is independent

Probability distributions: Permutations and Combinations

What is the probability distribution of number of girls in families with two children?

Number of Girls	Child #1	Child #2
2	G	G
1	B	G
1	G	B
0	B	B

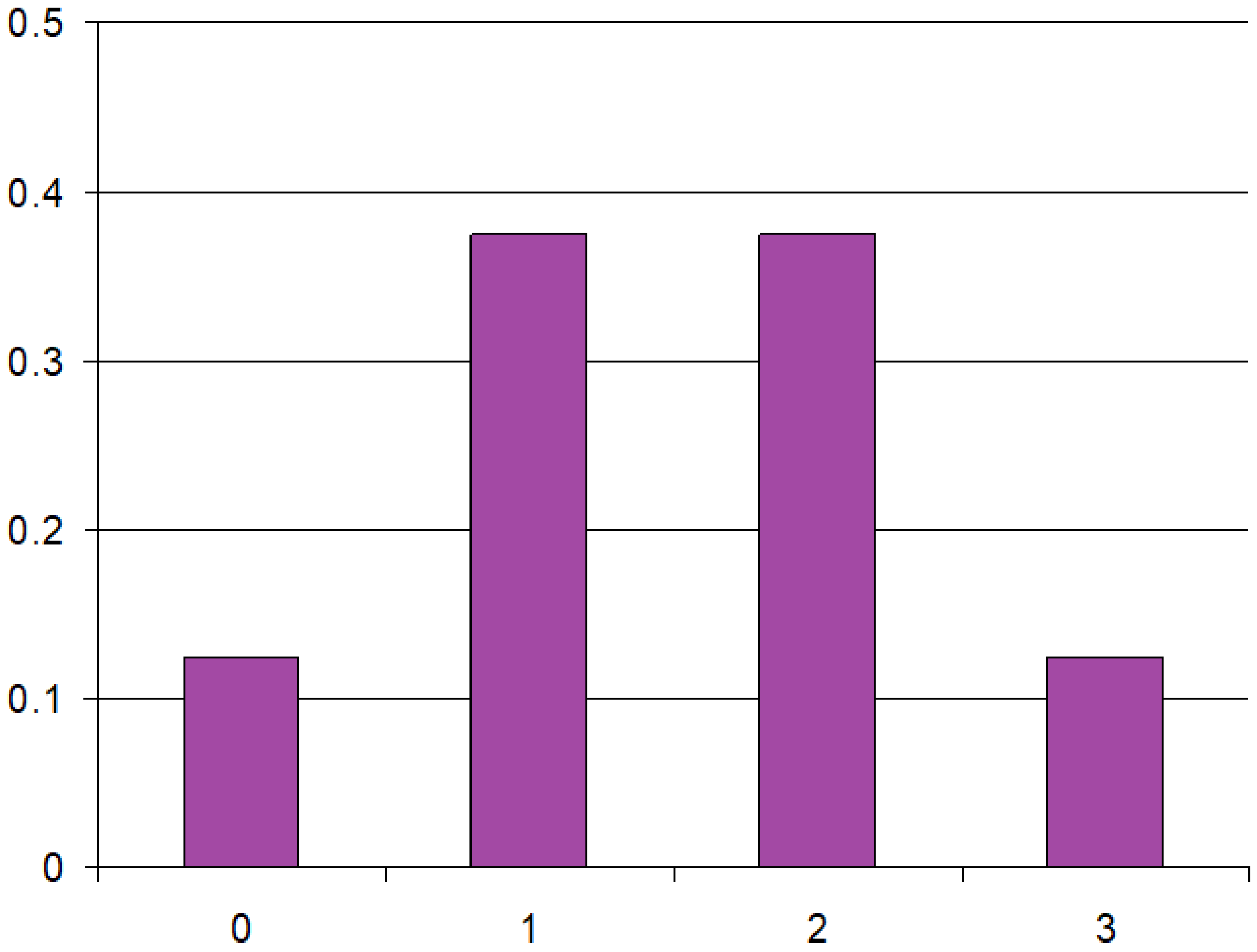
Probability Distribution for Number of Girls in Two Child Family



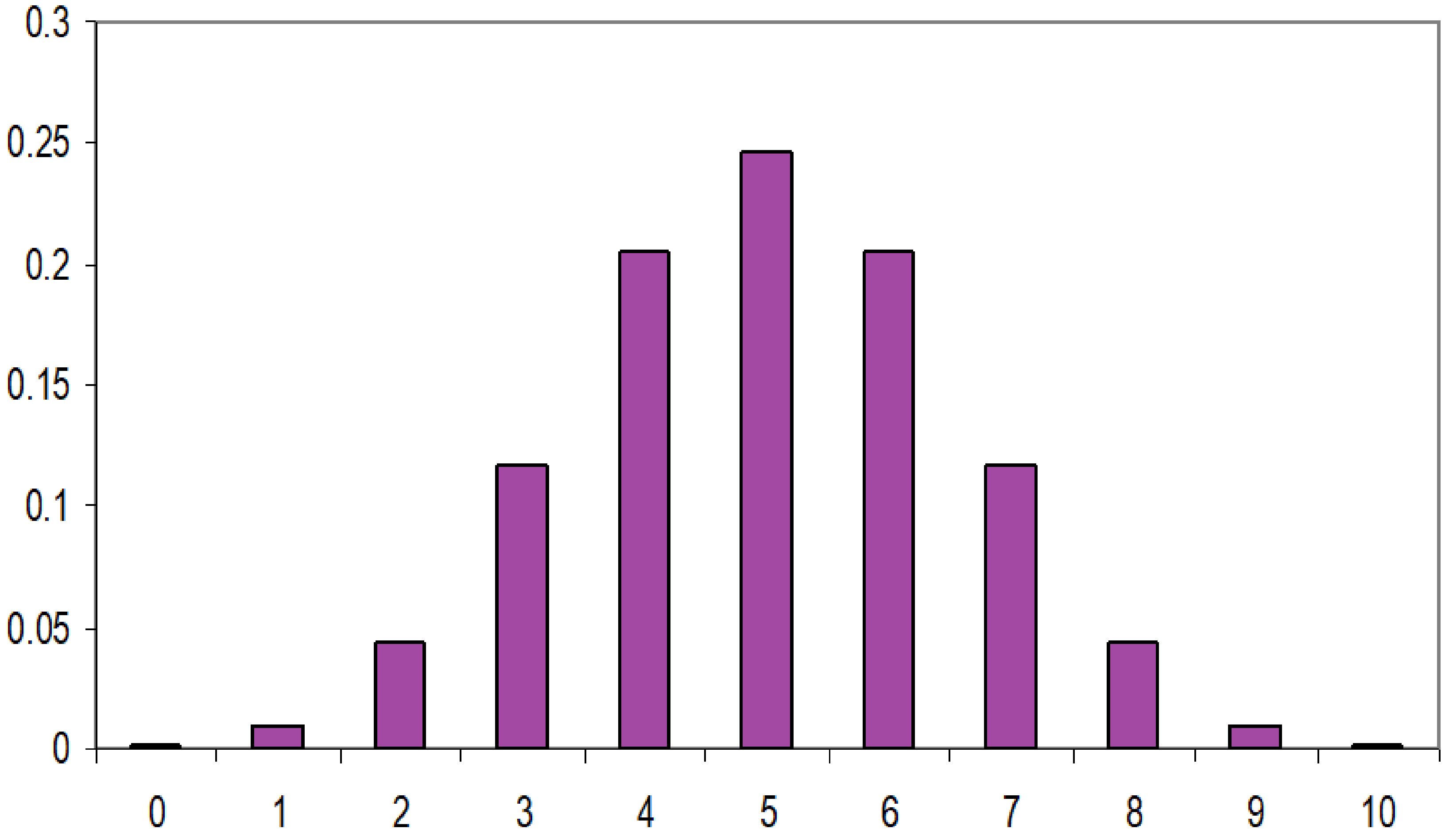
How about family of three?

Num. Girls	child #1	child #2	child #3
0	B	B	B
1	B	B	G
1	B	G	B
1	G	B	B
2	B	G	G
2	G	B	G
2	G	G	B
3	G	G	G

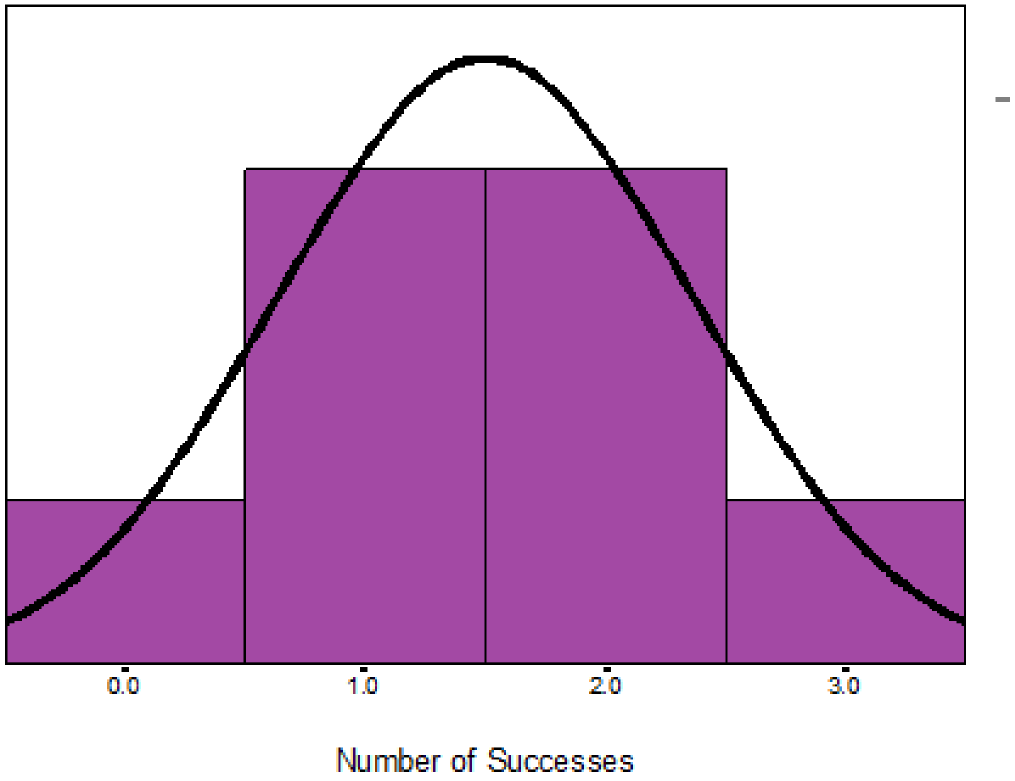
Probability Distribution for Number of Girls in Three Child Family

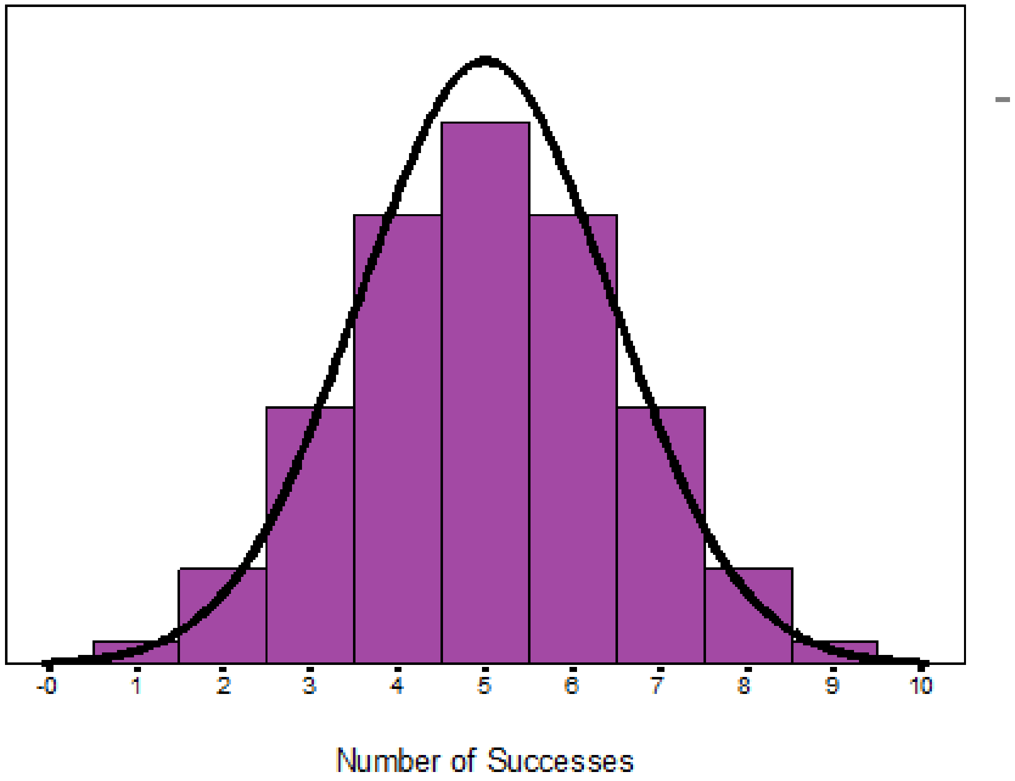


How about a family of 10?



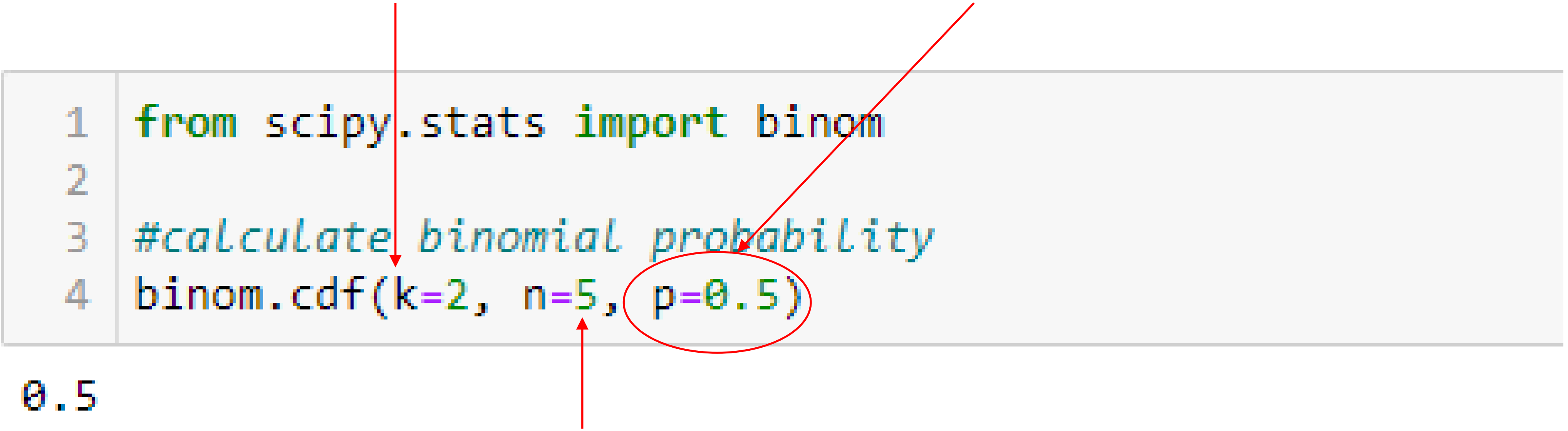
As trial number increases, the binomial distribution looks more and more like a normal distribution





Python Code for Binomial Distribution

- Marty flips a fair coin 5 times. What is the probability that the coin lands on heads 2 times or fewer?
- We use the cdf() method as we want the cumulative probability for 2 or fewer "successes".



```python
from scipy.stats import binom

#calculate binomial probability
binom.cdf(k=2, n=5, p=0.5)
```

```
0.5
```

Number of "successes" we are using as upper limit for cumulative probability calculation

Probability of a "success" (i.e., a "head" toss in this case) in each trial.

Number of trials (i.e., coin tosses)

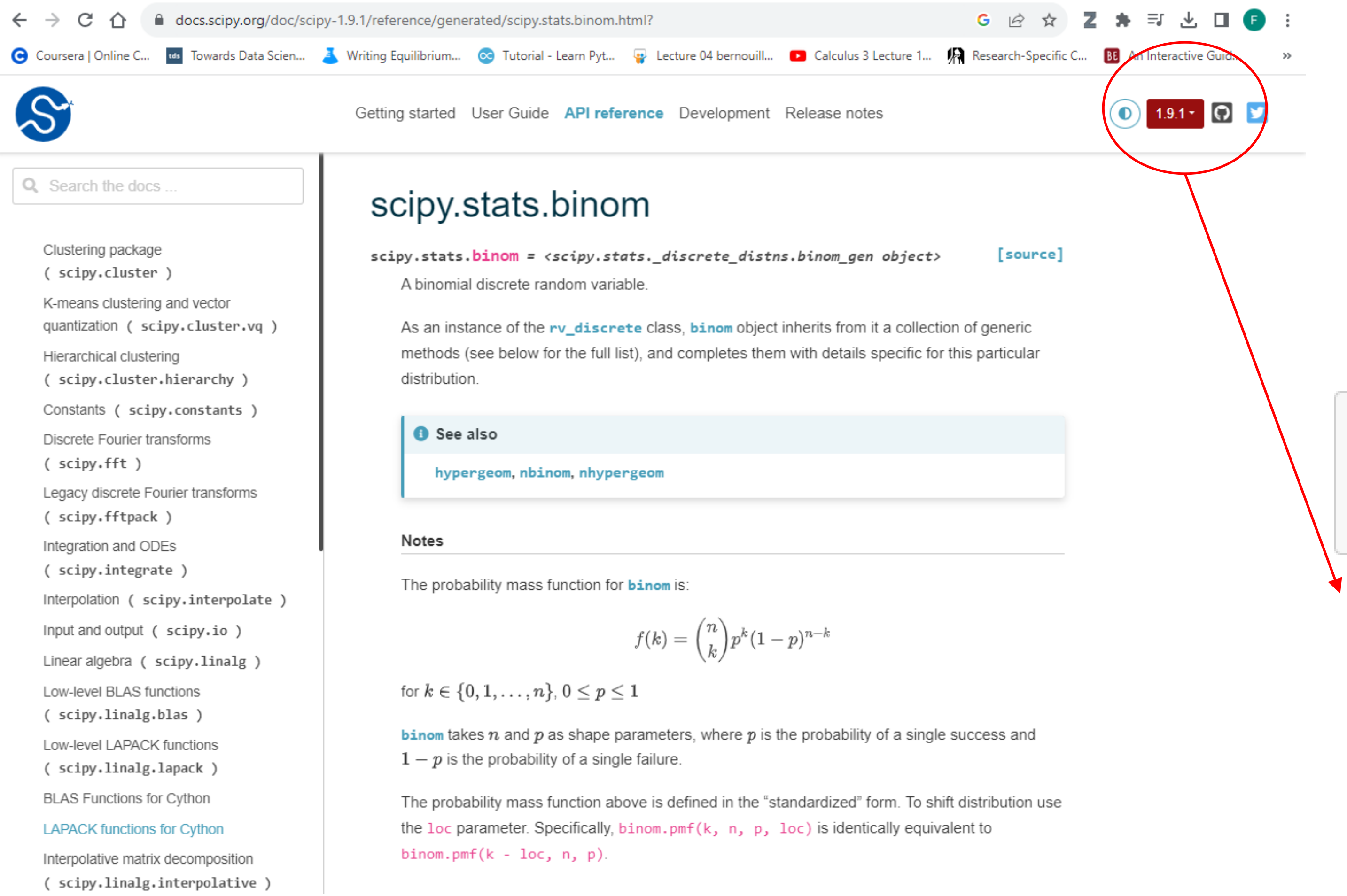
Documentation for Libraries

If you are unclear what the arguments passed to a library method actually represent or have other questions about the logic implementation of the library method, you should look at the documentation, particularly the **API** (application programming interface) which has details. Below are top level links to the documentation of the main libraries we are using.

- https://numpy.org/doc/
- https://docs.scipy.org/doc/
- https://pandas.pydata.org/docs/
- https://scikit-learn.org/stable/modules/classes.html
- https://matplotlib.org/stable/api/index.html
- https://seaborn.pydata.org/api.html

Versions

- When checking documentation, make sure the version of library you are using matches the documentation:


docs.scipy.org/doc/scipy-1.9.1/reference/generated/scipy.stats.binom.html?
Coursera | Online C...
Towards Data Scien...
Writing Equilibrium...
Tutorial - Learn Pyt...
Lecture 04 bernouill...
Calculus 3 Lecture 1...
Research-Specific C...
An Interactive Guid...
Getting started
User Guide
API reference
Development
Release notes
1.9.1
Search the docs ...
Clustering package (scipy.cluster)
K-means clustering and vector quantization (scipy.cluster.vq)
Hierarchical clustering (scipy.cluster.hierarchy)
Constants (scipy.constants)
Discrete Fourier transforms (scipy.fft)
Legacy discrete Fourier transforms (scipy.fftpack)
Integration and ODEs (scipy.integrate)
Interpolation (scipy.interpolate)
Input and output (scipy.io)
Linear algebra (scipy.linalg)
Low-level BLAS functions (scipy.linalg.blas)
Low-level LAPACK functions (scipy.linalg.lapack)
BLAS Functions for Cython
LAPACK functions for Cython
Interpolative matrix decomposition (scipy.linalg.interpolative)

scipy.stats.binom

`scipy.stats.binom = <scipy.stats._discrete_distns.binom_gen object>` [source]

A binomial discrete random variable.

As an instance of the `rv_discrete` class, `binom` object inherits from it a collection of generic methods (see below for the full list), and completes them with details specific for this particular distribution.

See also

`hypergeom`, `nbinom`, `nhypergeom`

Notes

The probability mass function for `binom` is:

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k}$$

for $k \in \{0, 1, \dots, n\}$, $0 \leq p \leq 1$

`binom` takes n and p as shape parameters, where p is the probability of a single success and $1 - p$ is the probability of a single failure.

The probability mass function above is defined in the "standardized" form. To shift distribution use the `loc` parameter. Specifically, `binom.pmf(k, n, p, loc)` is identically equivalent to `binom.pmf(k - loc, n, p)`.

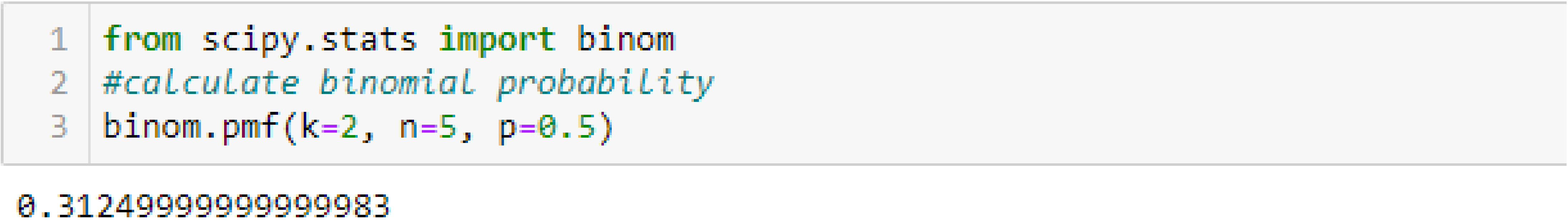
Use the "__version__" attribute from the library:

```
import scipy
print(scipy.__version__)
```

```
1.9.1
```

Python Code for Binomial Distribution

- Marty flips a fair coin 5 times. What is the probability that the coin lands on heads exactly 2 times?
- In this case, as we are looking for the probability of a specific success count and not a cumulative probability, we use the pmf() method



```
from scipy.stats import binom
#calculate binomial probability
binom.pmf(k=2, n=5, p=0.5)
```

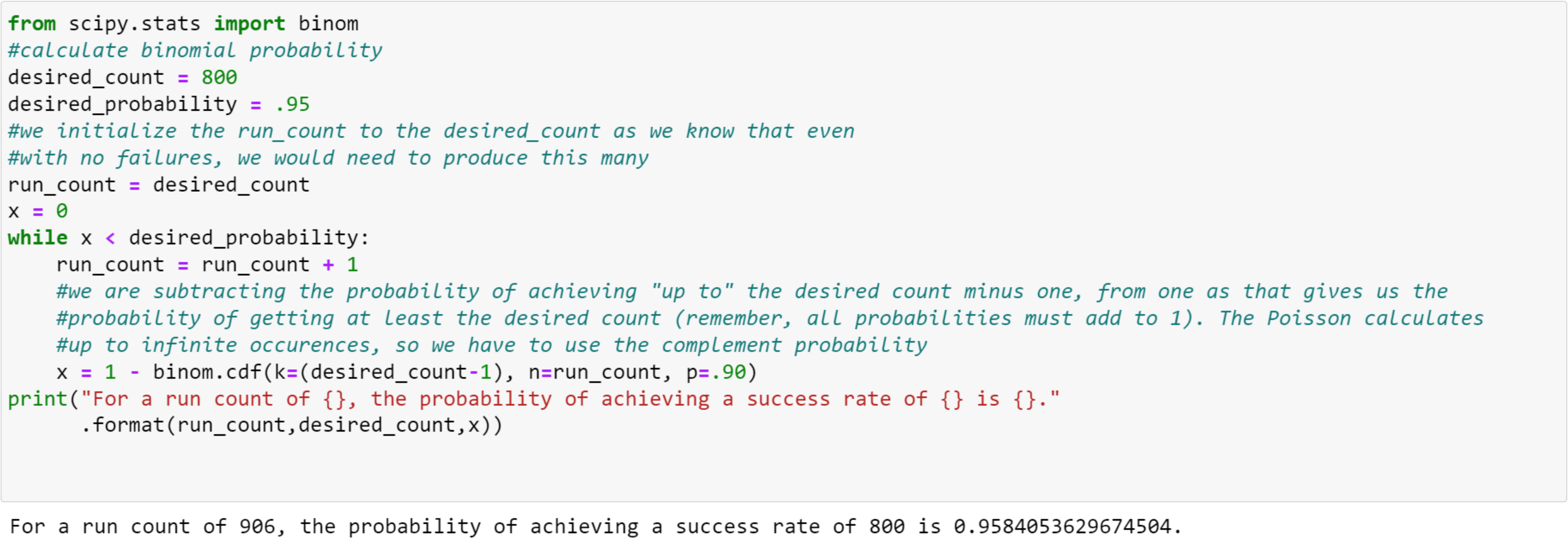
```
0.31249999999999983
```

Binomial Distribution
More Applicable Uses

- Fair coin tosses (50/50 odds) are an easy example, but not particularly relevant
- Consider known probability of product failure (.10) in a particular, complicated manufacturing process (meaning we have to plan the size of the manufacturing run in advance, not check each unit as it comes off the line and stop when we have enough). Over production is prohibitively costly, but underproduction for forecasted demand (800) must be avoided.
- A .95 probability of achieving goal is considered acceptable risk.
- How many units do we need to plan for production to achieve this?

Code Solution

We can write a small piece of code that iteratively increase the number of “trials” (in this case products on the manufacturing line) until the probability of successfully making the desired number is at an acceptable level.



```
from scipy.stats import binom
#calculate binomial probability
desired_count = 800
desired_probability = .95
#we initialize the run_count to the desired_count as we know that even
#with no failures, we would need to produce this many
run_count = desired_count
x = 0
while x < desired_probability:
    run_count = run_count + 1
    #we are subtracting the probability of achieving "up to" the desired count minus one, from one as that gives us the
    #probability of getting at least the desired count (remember, all probabilities must add to 1). The Poisson calculates
    #up to infinite occurences, so we have to use the complement probability
    x = 1 - binom.cdf(k=(desired_count-1), n=run_count, p=.90)
print("For a run count of {}, the probability of achieving a success rate of {} is {}."
      .format(run_count,desired_count,x))
```

```
For a run count of 906, the probability of achieving a success rate of 800 is 0.9584053629674504.
```

Poisson Distribution

Poisson Distribution - overview

- Poisson distribution is for counts—if events happen at a constant rate over some interval of time or space (or volume of space), the Poisson distribution gives the probability of X number of events occurring in time T.
- The Poisson distribution has been used extensively to model probability in biology and medicine amongst applications in other fields including radioactive decay in physics.

Poisson Distribution - form

- The Poisson distribution models counts, such as the number of new cases of SARS that occur in women in New England next month.
- The distribution tells you the probability of all possible numbers of new cases, from 0 to infinity.
- If λ (“the parameter of the distribution”) is equal to average number of occurrences of the event in the specified interval (month), X= number of new cases next month and *X* ~ Poisson (λ), then the probability that *X*=*k* (a particular count) is:

$$\boxed{p(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}}$$

Don’t memorize formula!

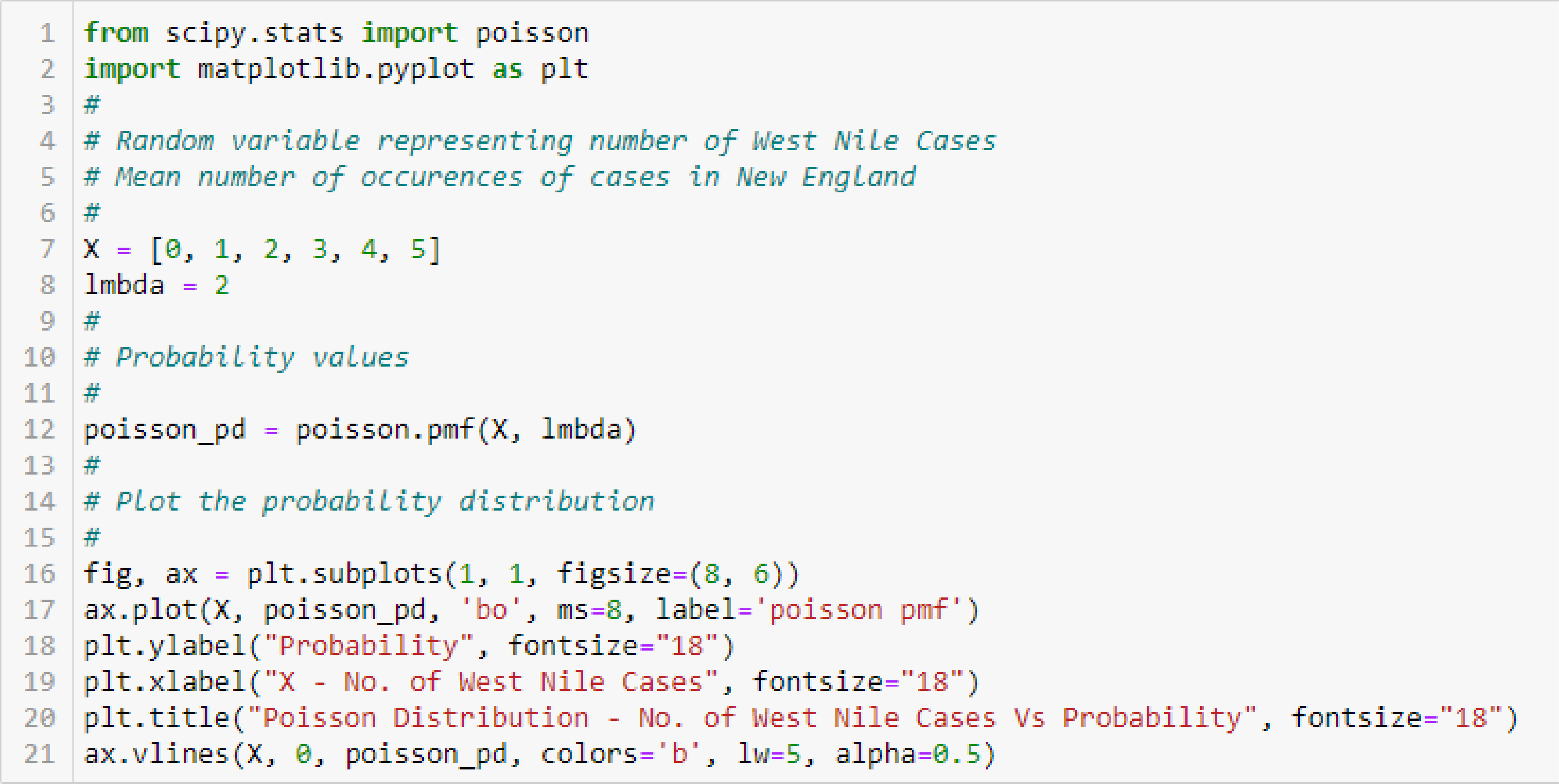
Poisson Distribution

- For example, if new cases of West Nile Virus in New England are occurring at a rate of about 2 per month, then let's calculate the probabilities that: 0,1, 2, 3, 4, or 5 cases will occur in New England in the next month…
- $\lambda = 2$

For X occurrences of West Nile in next month given previous information:

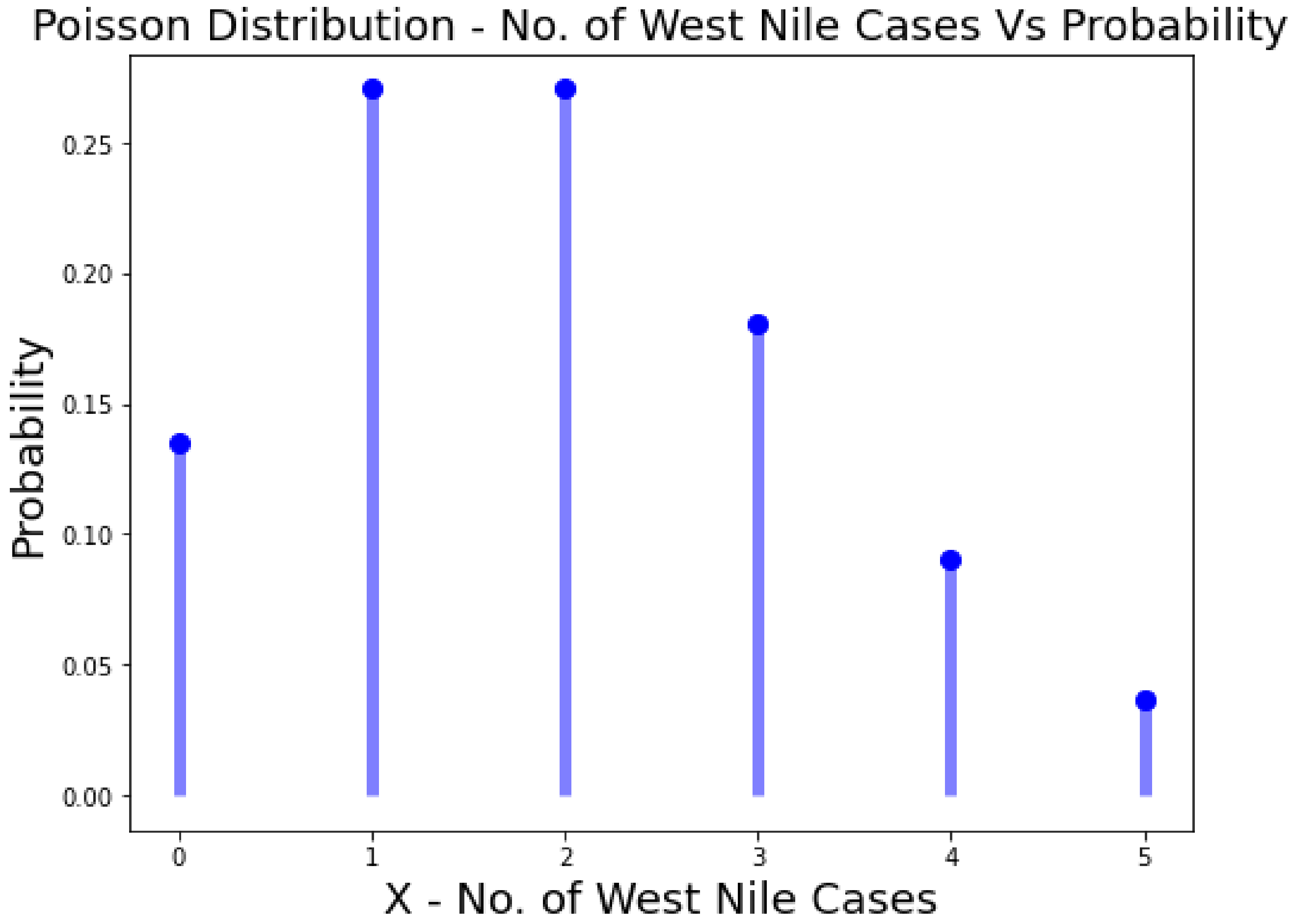
X	P(X)
0	0.135335
1	0.270671
2	0.270671
3	0.180447
4	0.090224
5	0.036089

Python Code to Generate Poisson Probabilities



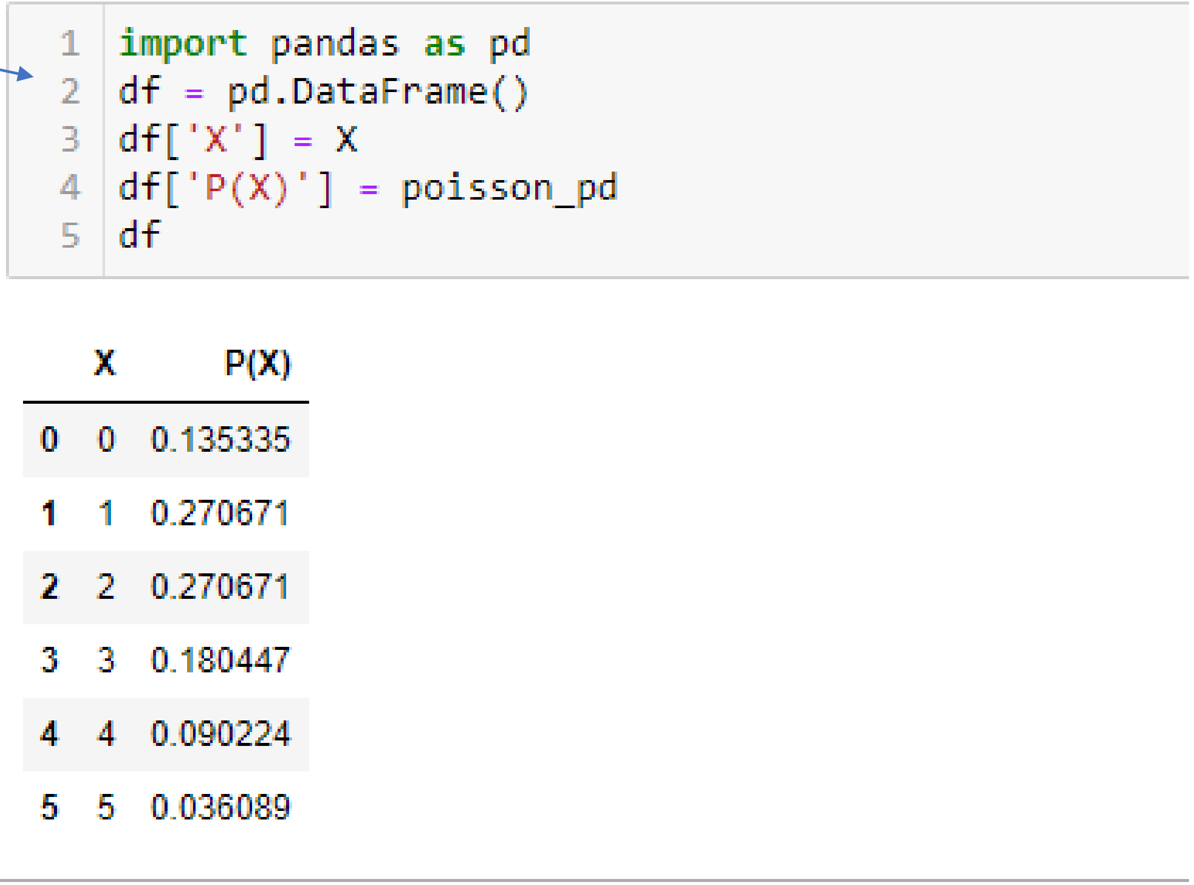
```python
from scipy.stats import poisson
import matplotlib.pyplot as plt
#
# Random variable representing number of West Nile Cases
# Mean number of occurences of cases in New England
#
X = [0, 1, 2, 3, 4, 5]
lmbda = 2
#
# Probability values
#
poisson_pd = poisson.pmf(X, lmbda)
#
# Plot the probability distribution
#
fig, ax = plt.subplots(1, 1, figsize=(8, 6))
ax.plot(X, poisson_pd, 'bo', ms=8, label='poisson pmf')
plt.ylabel("Probability", fontsize="18")
plt.xlabel("X - No. of West Nile Cases", fontsize="18")
plt.title("Poisson Distribution - No. of West Nile Cases Vs Probability", fontsize="18")
ax.vlines(X, 0, poisson_pd, colors='b', lw=5, alpha=0.5)
```

Poisson Distribution Plot



Python Code for Poisson Probabilities Table

Assumes ‘X’ and ‘poisson_pd’ variables were assigned as shown on previous slides!



```python
import pandas as pd
df = pd.DataFrame()
df['X'] = X
df['P(X)'] = poisson_pd
df
```

```
   X      P(X)
0  0  0.135335
1  1  0.270671
2  2  0.270671
3  3  0.180447
4  4  0.090224
5  5  0.036089
```

Poisson - How useful is it, really?

- As a stand alone tool applied to one example question without context… not very.
- However, if you consider a complicated context such as a manufacturing process with:
  - Average rates of equipment failure for multiple components
  - Budget limitations
  - Redundancy requirements
  - Acceptable levels of risk

The Poisson distribution can be used to allocate resources (e.g., spare parts, maintenance staff, etc.) in a manner that protects against all but the most unlikely scenarios.

It is also used in sales, and as mentioned previously, in medicine and the sciences.