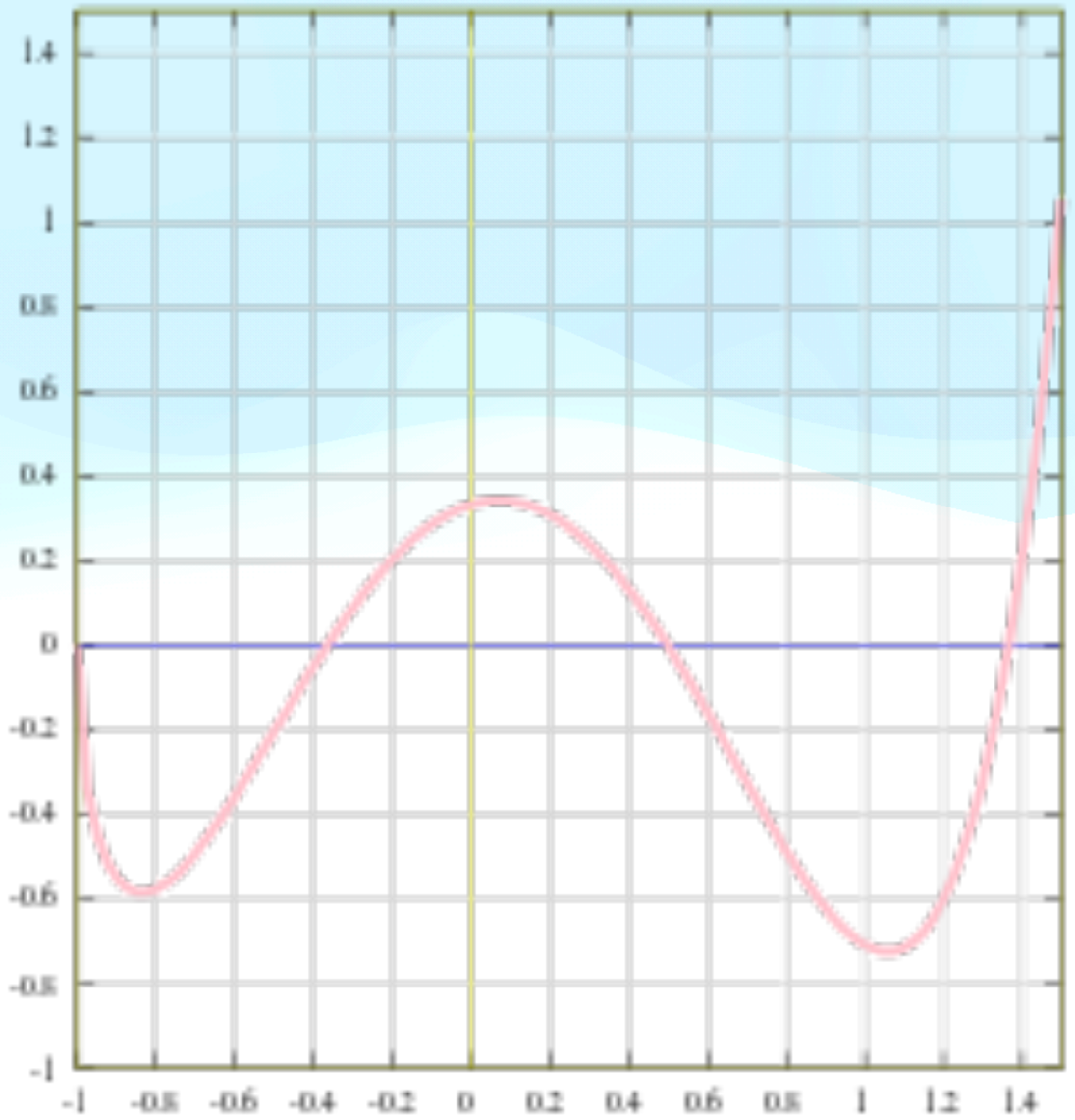
Probability Density Functions

PHYS 2601

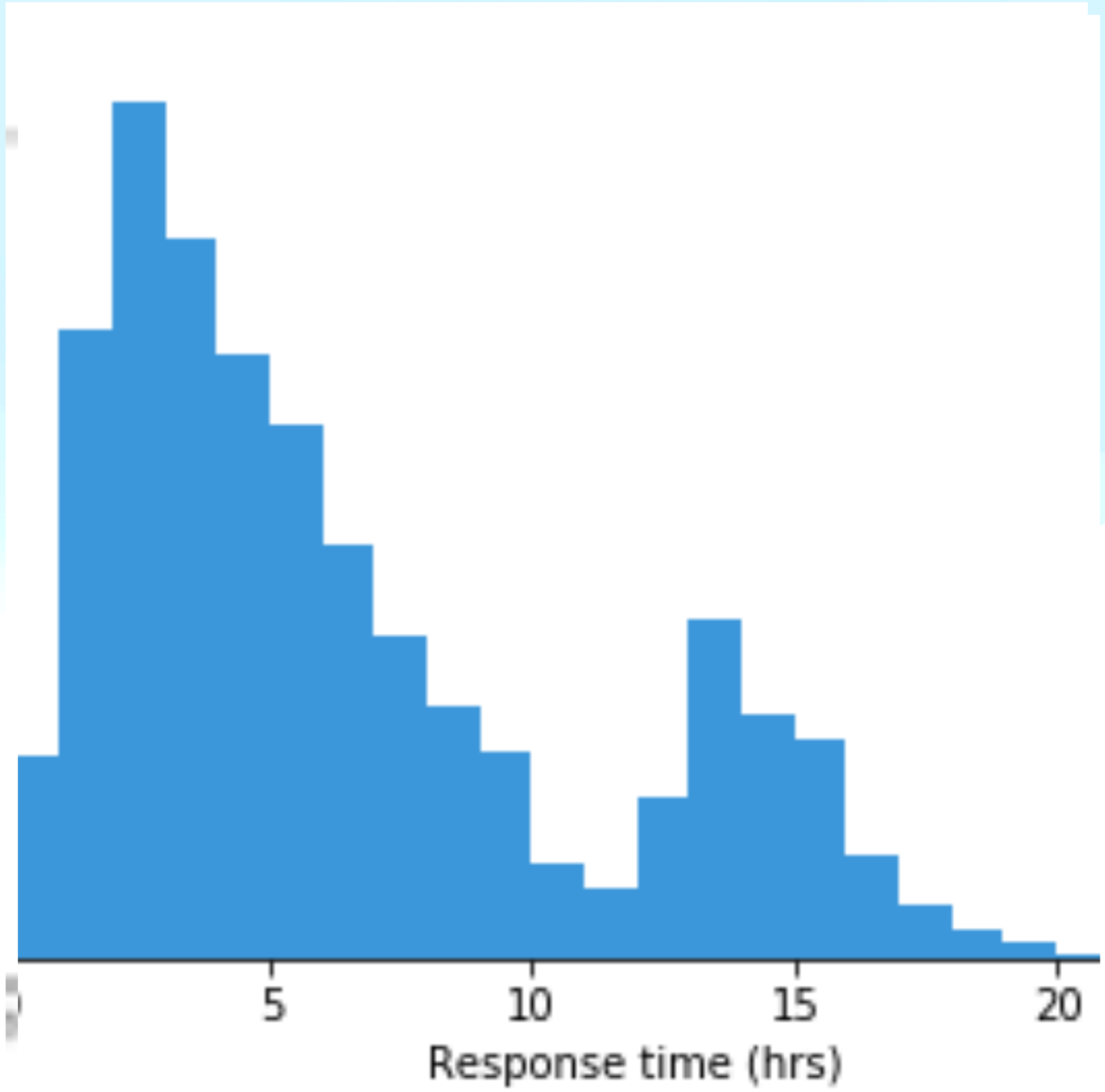
Functional Form

- We have discussed summary statistics that can give us a few numbers to describe our distribution, but they seem to loose a lot of information.
- It might be nice if we had a continuous function that completely described the distribution.
- But how would we do this? Normally a function gives us a y value based on some x value.
- Distributions are many many x values. What is y?



Histograms

- We have made a plot with distributions, the histogram.
- In that case the y values are the number of occurrences
- So our function could be the expected number of occurrences in each bin. But these are discrete values not continuous.
- As a continuous function we could consider the probability of an occurrence in each bin.



Probability Density Function

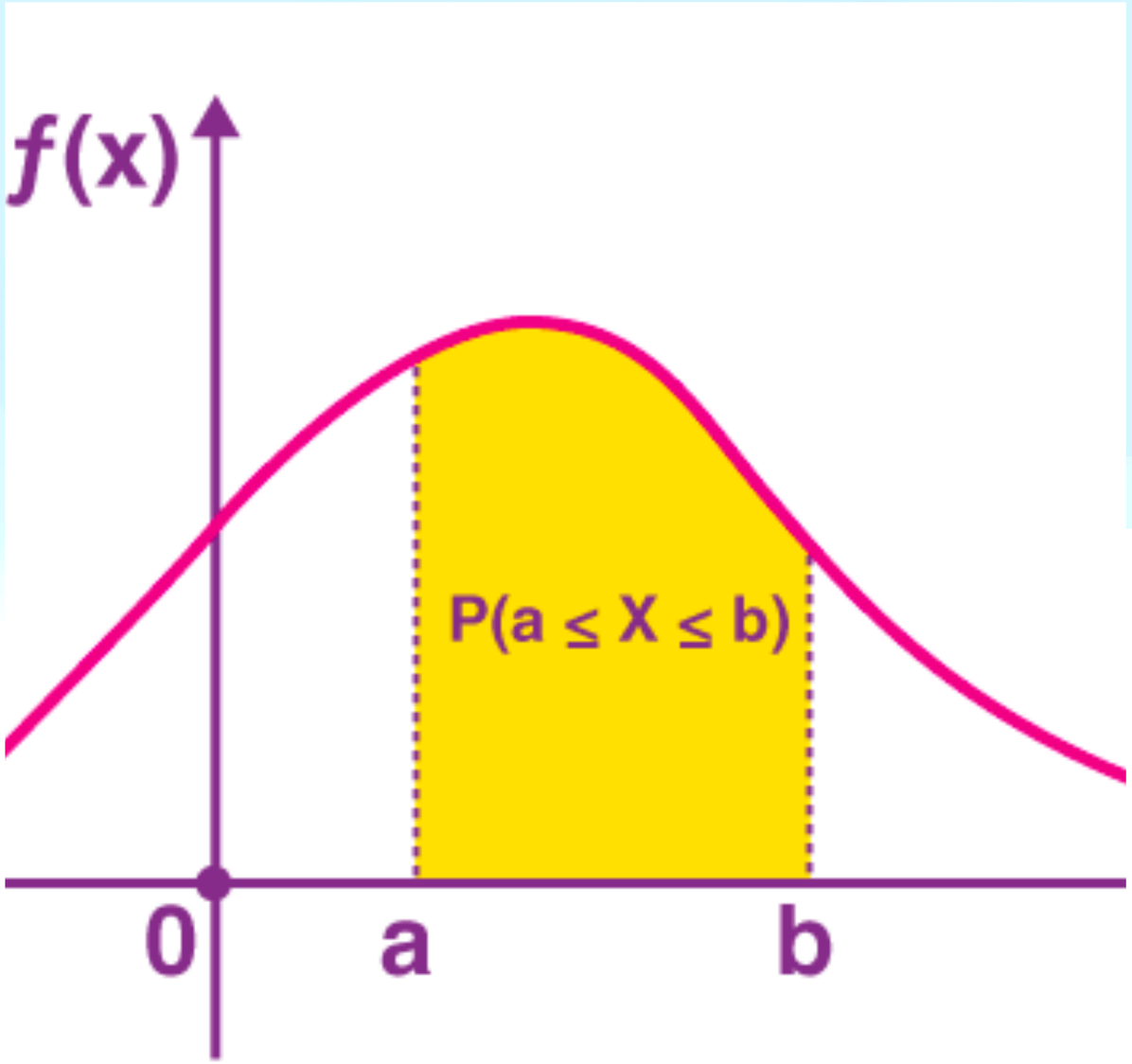
- A probability density function, gives the probability that some variable x, will have a value between a and b when integrated.

$$P(a > x > b) = \int_a^b p(x)dx$$

- of course the probability of getting exactly x is infinitesimally small

$$P(x) = p(x)dx$$

- Note that this is why this is referred to as a density, to get a probability one has to integrate over some range



Distributions are finite samples of pdfs

- A probability density function (pdf) can be used to generate a finite distribution of a variable. But the finite distribution can never be the same as the pdf.
- As the number of values in our sample approaches infinity the distribution will become closer and closer to the pdf.
- Real samples will differ from the pdf substantially and from one another if randomly generated. That is a two samples drawn from the same pdf will not be the same.
- In fact a lot of this course will be trying to determine if a sample was drawn from a pdf or if two samples could be drawn from the same pdf.

Common probability density functions

In statistics and physics

- There are a number of common pdfs that either can be calculated from probability theory or are observed in physics.
    - Binomial distribution
    - Poisson distribution
    - Gaussian (normal) distribution
    - log-normal distribution
    - uniform distribution
    - power-law distribution
    - exponential distribution
    - gamma distribution

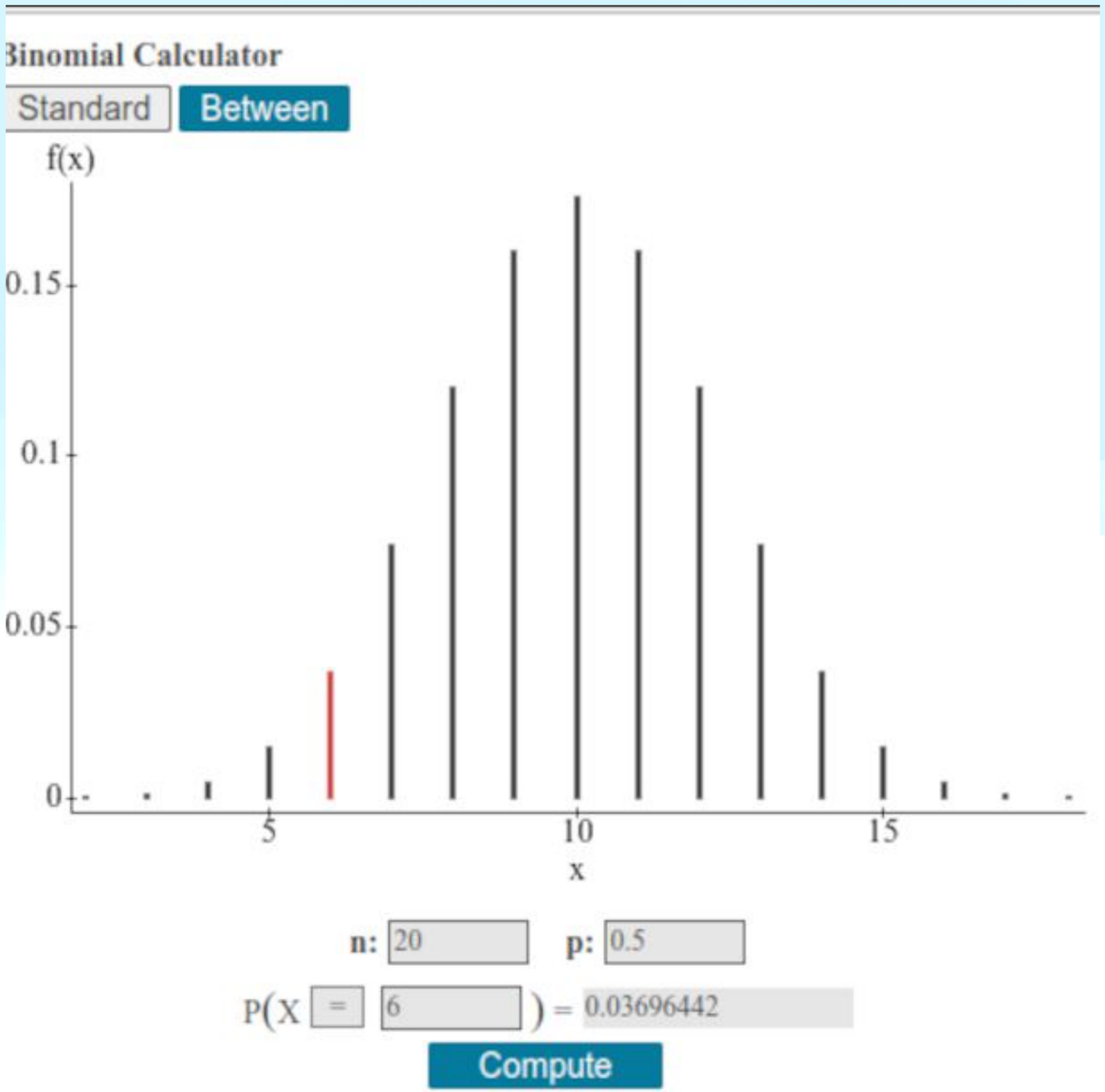
Binomial Distribution

discrete distribution

- The binomial distribution describes situations where there is some chance p of something happening and you want to know if you do it n times what is the chance that is happens k times. If they are independent then the answer is

$$P(k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$$

- For example in the case of flipping a coin 10 times the chance of getting 5 heads would p=0.5, n=10, k=5.
- The chance of rolling a six on a six sided die 5 times with 5 rolls would be given by p= 1/6, n=5, k=5.



scipy.stats.binom.pmf(k=5, n=10, p=0.5)

Note pmf, probability mass function, not pdf for discrete distributions

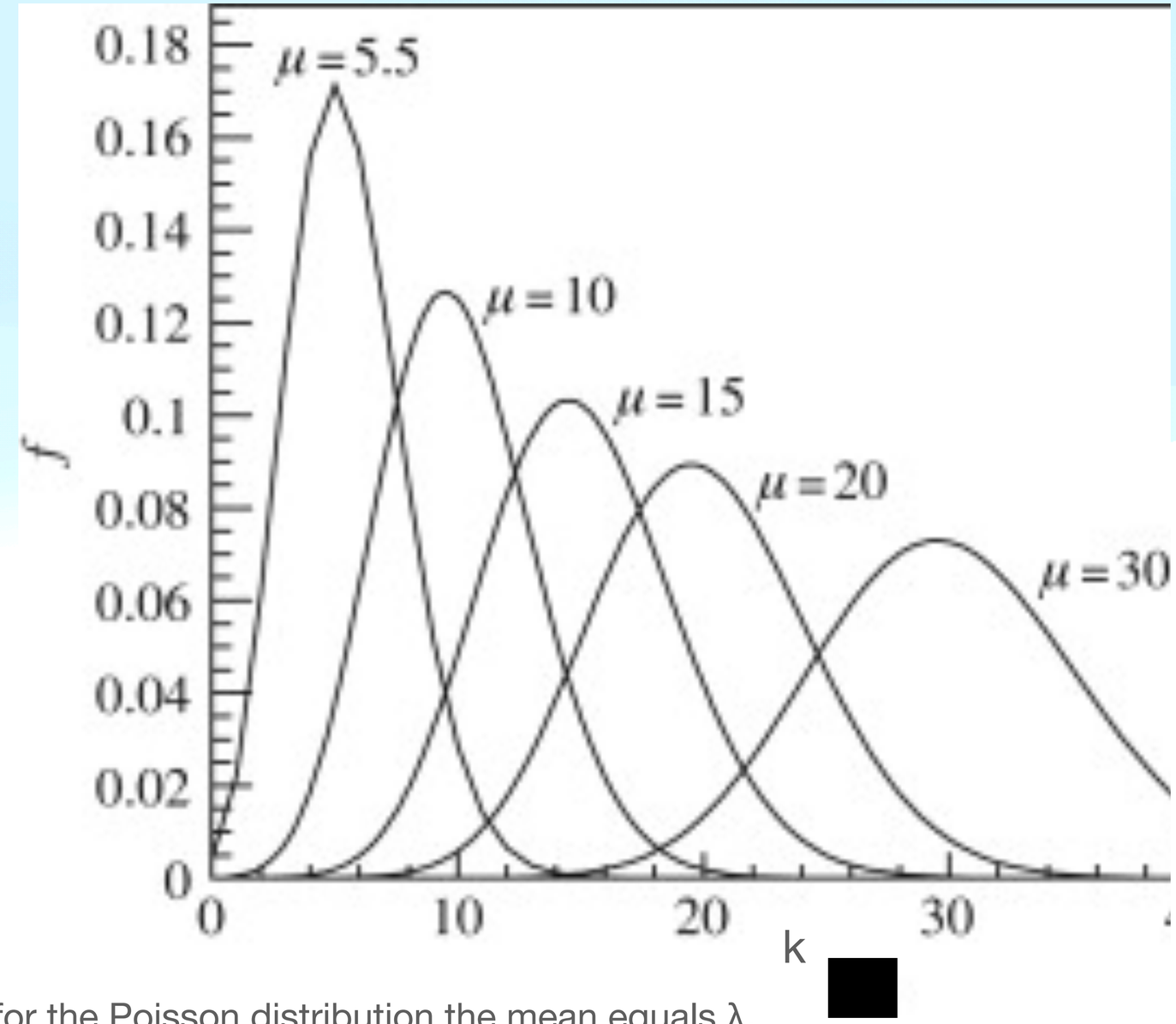
Poisson Distribution

discrete distribution

- If the average rate of events for some process is r and they are independent, then during some interval, t, the probability of having exactly k events is

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

- where λ = rt.
- For example, if the decay rate of a certain atom is one in five minutes, then you can calculate the probability that 0, 1 or 2 atoms decayed during a 1 minute interval.



scipy.stats.poisson.pmf(k, λ)

Note for the Poisson distribution the mean equals λ.

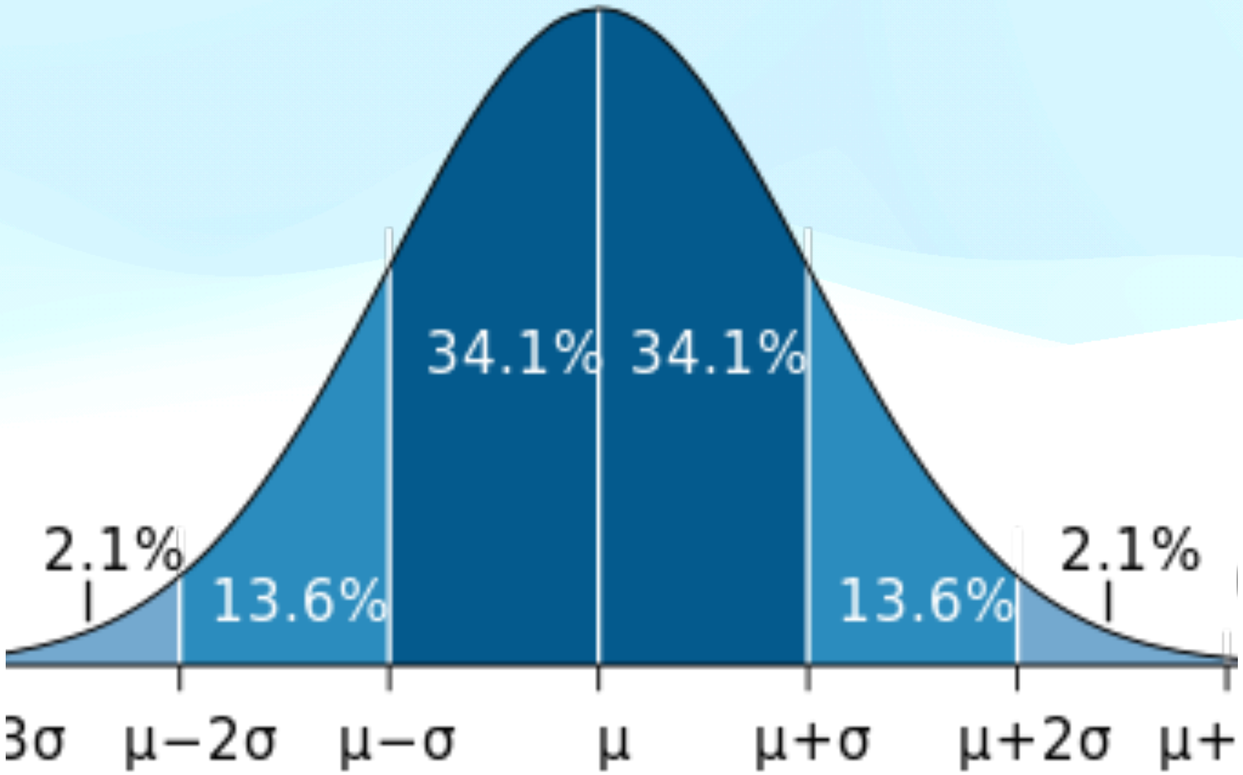
Gaussian Distribution

continuous distribution

- The most famous pdf is the Gaussian, or normal, distribution. This distribution is the result of an infinite number of random independent events. Thus the binomial and Poisson distributions approach the Gaussian for large n.
- For a mean of μ and a standard deviation of σ, the Gaussian pdf is

$$P(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}}$$

- This distribution is often assumed for measurement errors, but rarely actually correct. The number of random events might be large, but is rarely that large.



scipy.stats.norm.pdf(x, loc=μ, scale=σ)

Log-normal Distribution

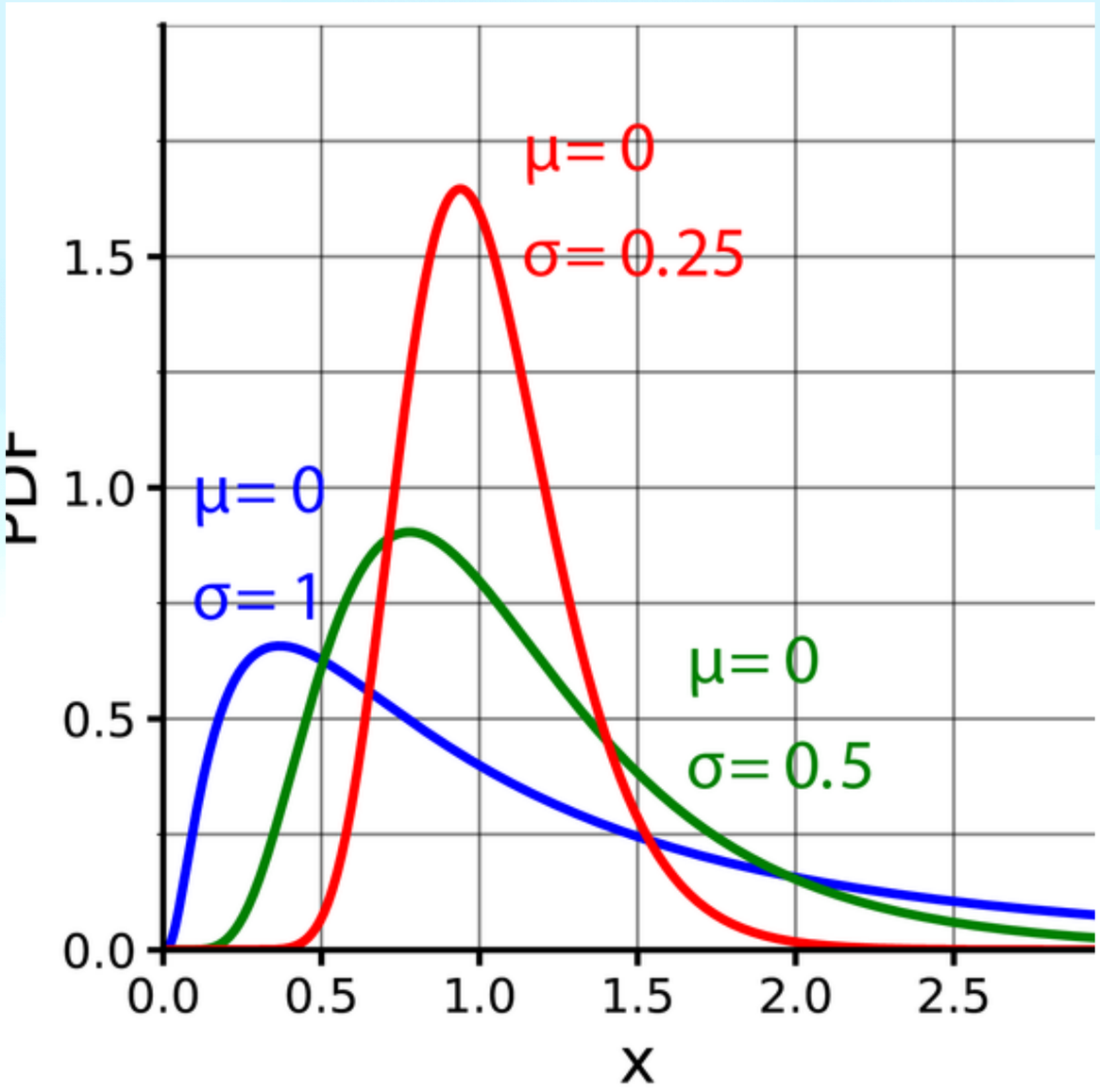
continuous distribution

- This distribution is exactly what its name is, if you take the log of your variable then that is normally distributed. If you have a normal distribution and you take e^x, then that has a log-normal distribution. Thus the pdf is

$$P(x;\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x-\mu)^2}{2\sigma^2}}$$

- One can think of this as the normal distribution has random numbers added to one another, while the log normal is random numbers multiplied.

- One important difference is that x is only positive.



scipy.stats.lognorm.pdf(x, sigma, loc, scale)

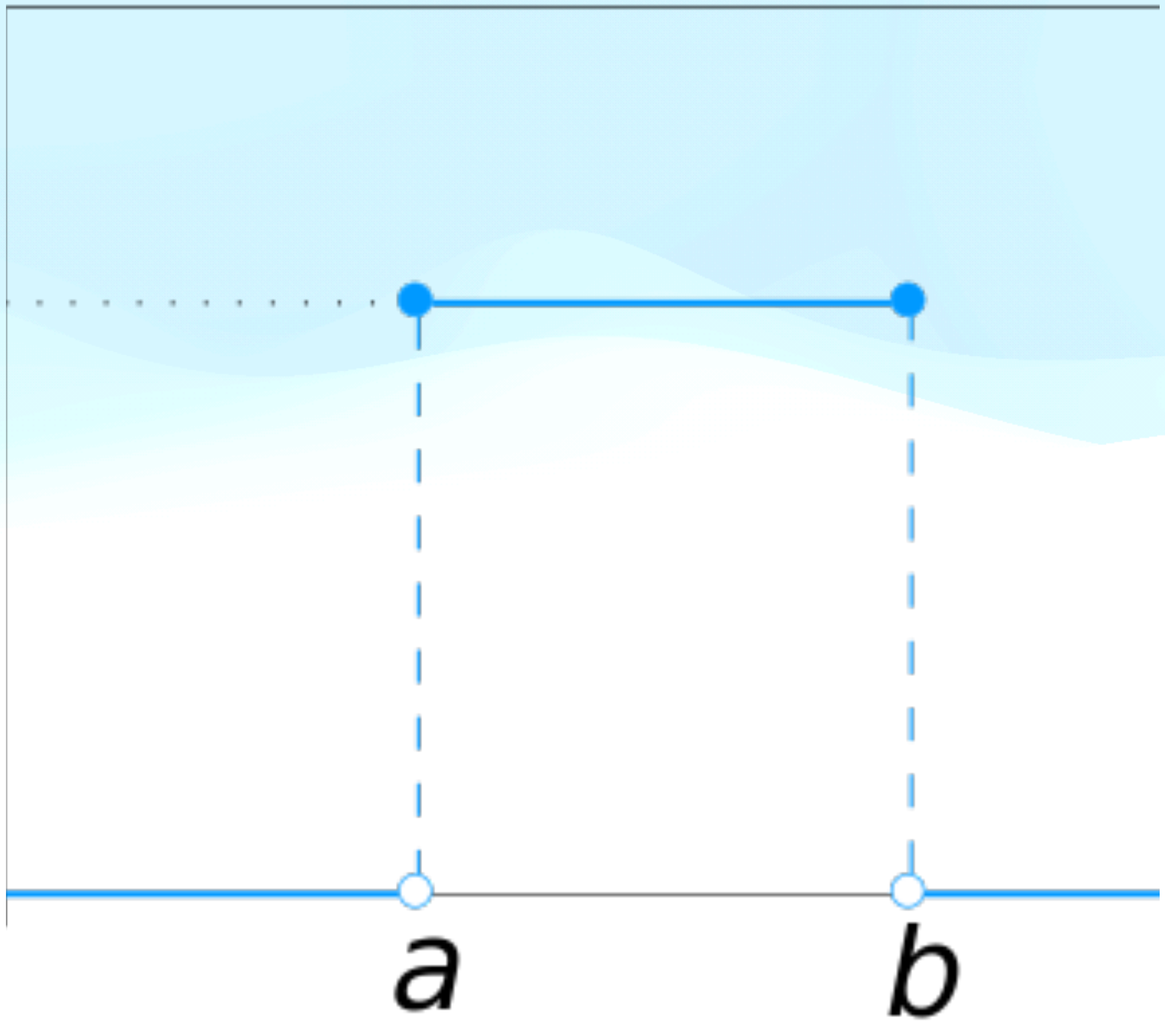
Uniform Distribution

discrete or continuous distribution

- The uniform distribution is just equal probability, between a finite range a to b.

$$P(x) = \frac{1}{b-a} \text{ for } \mathrm{a} \leq \mathrm{x} \leq \mathrm{b}$$

- The chance of rolling the values on a die, the chance of picking a card from a deck.
- The uniform distribution from 0 to 1 can be mapped to any other range.



scipy.stats.uniform.pdf(loc=a, scale=b-a)

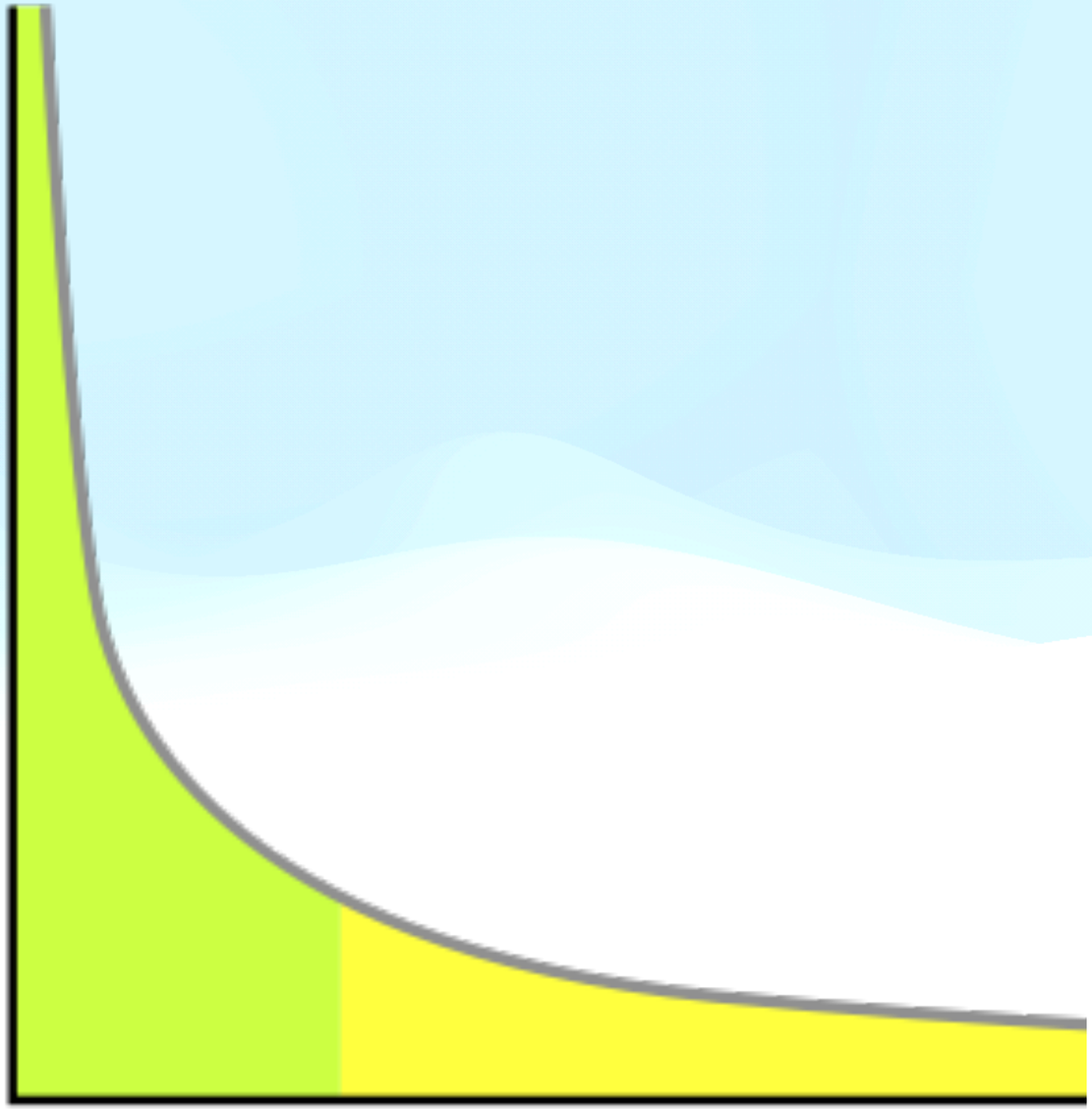
Power law Distribution

continuous distribution

- Another important distribution is the power law. This distribution will occur when there is no scale in the problem (the power law is scale free).

$$f(x) = cx^{-a}$$

- How this is normalized to a probability depends on k and the range. Rescaling x only changes the value of c. This function has a mean only if a > 2.
- Describes crater sizes on the moon, sizes of power outages, distribution of wealth.
- Usually only the tail of the distribution is well described by the power law.
- This distribution will be a line in a log-log plot.



scipy.stats.powerlaw.pdf(x,a,loc,scale)

Note the spicy.stats function has index a-1 not -a

Power Law

Is a line in log - log space

- We can use coordinate transformation to change the functional form of a distribution. The most common choice is to take the log of a variable as this will turn multiplication into addition.
- For the power law taking the log transforms our relationship into a linear one.

$$y = cx^{-a}$$

$$\ln y = \ln(cx^{-a})$$

$$\ln y = \ln c + -a \ln x$$

$$\ln y = -a \ln x + \ln c$$

$$y' = mx' + b$$

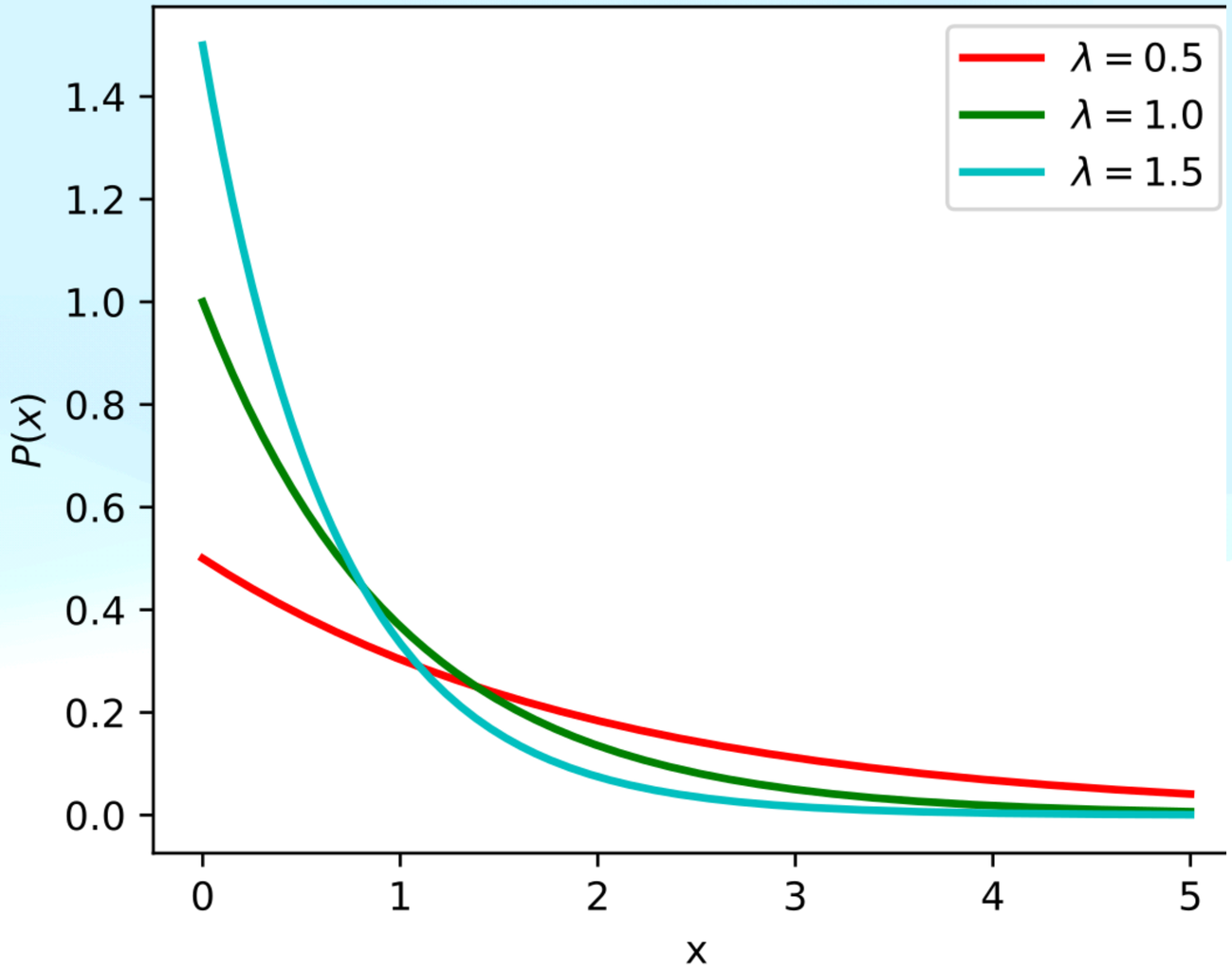
Exponential Distribution

continuous distribution

- An exponential distribution will occur if something depends on its derivative. The pdf is then

$$P(x) = \lambda e^{-\lambda x}$$

- This integrates to 1 over the range [0,∞].
- This distribution fits many things, radioactive decay, spread of infection.
- This distribution will be a line in a semi-log plot.



scipy.stats.expon.pdf(x, loc, scale = 1/λ)

Exponential

Is linear in semi-log space

- Likewise and exponential distribution can be rewritten as a linear one when transforming the

$$y = \lambda e^{-\lambda x}$$

$$\ln y = \ln(\lambda e^{-\lambda x})$$

$$\ln y = -\lambda x + \ln \lambda$$

$$y' = mx + b$$

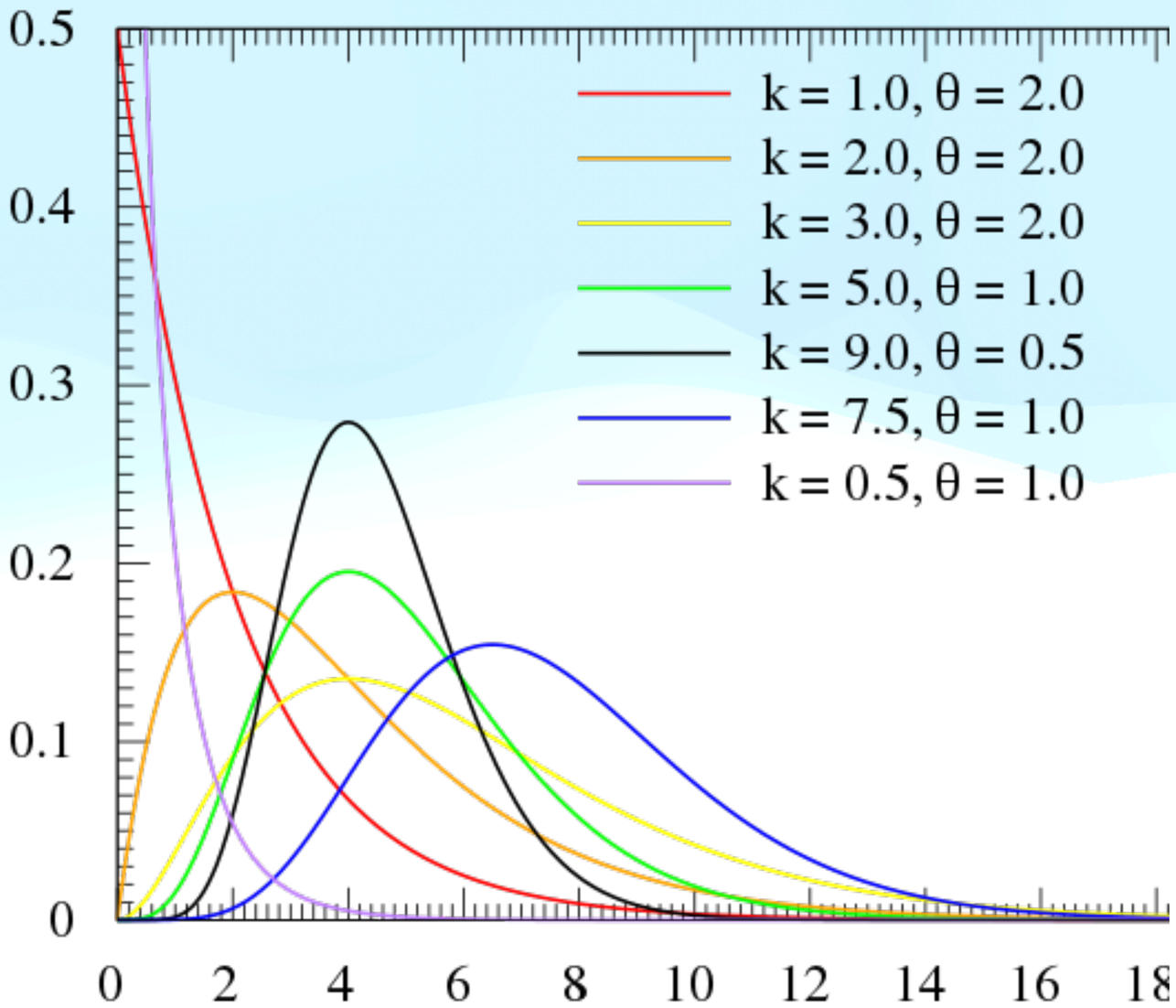
Gamma Distribution

continuous distribution

- There are many more complicated distributions that exist. Scipy stats has dozens.
- One example is the gamma distribution which is a power law with an exponential cutoff.

$$P(x;\alpha,\beta) = \frac{x^{\alpha-1}e^{-\beta x}\beta^{\alpha}}{\Gamma(\alpha)}$$

- Where Γ is the gamma function. For integers Γ(α) = (α-1)!

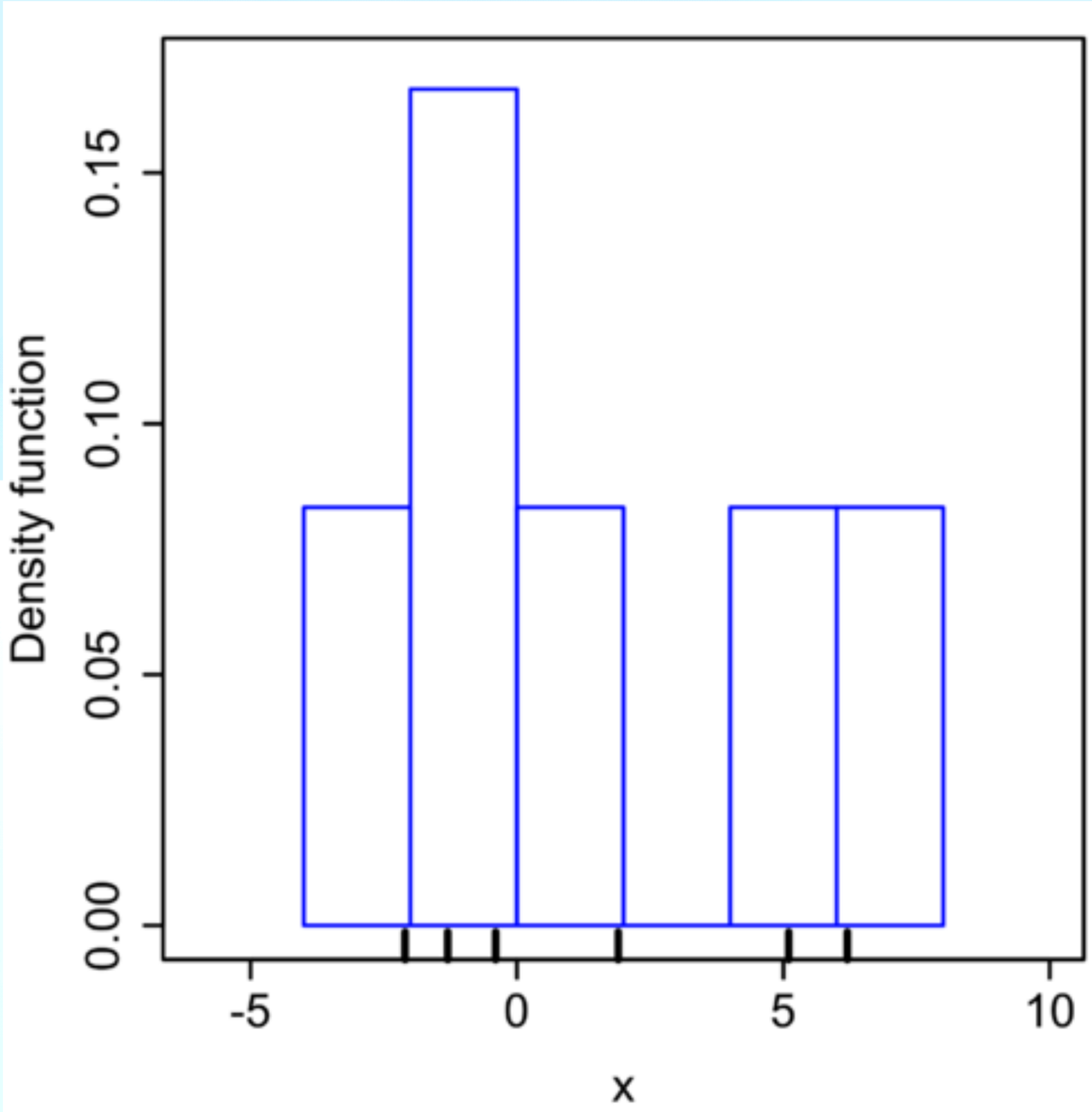


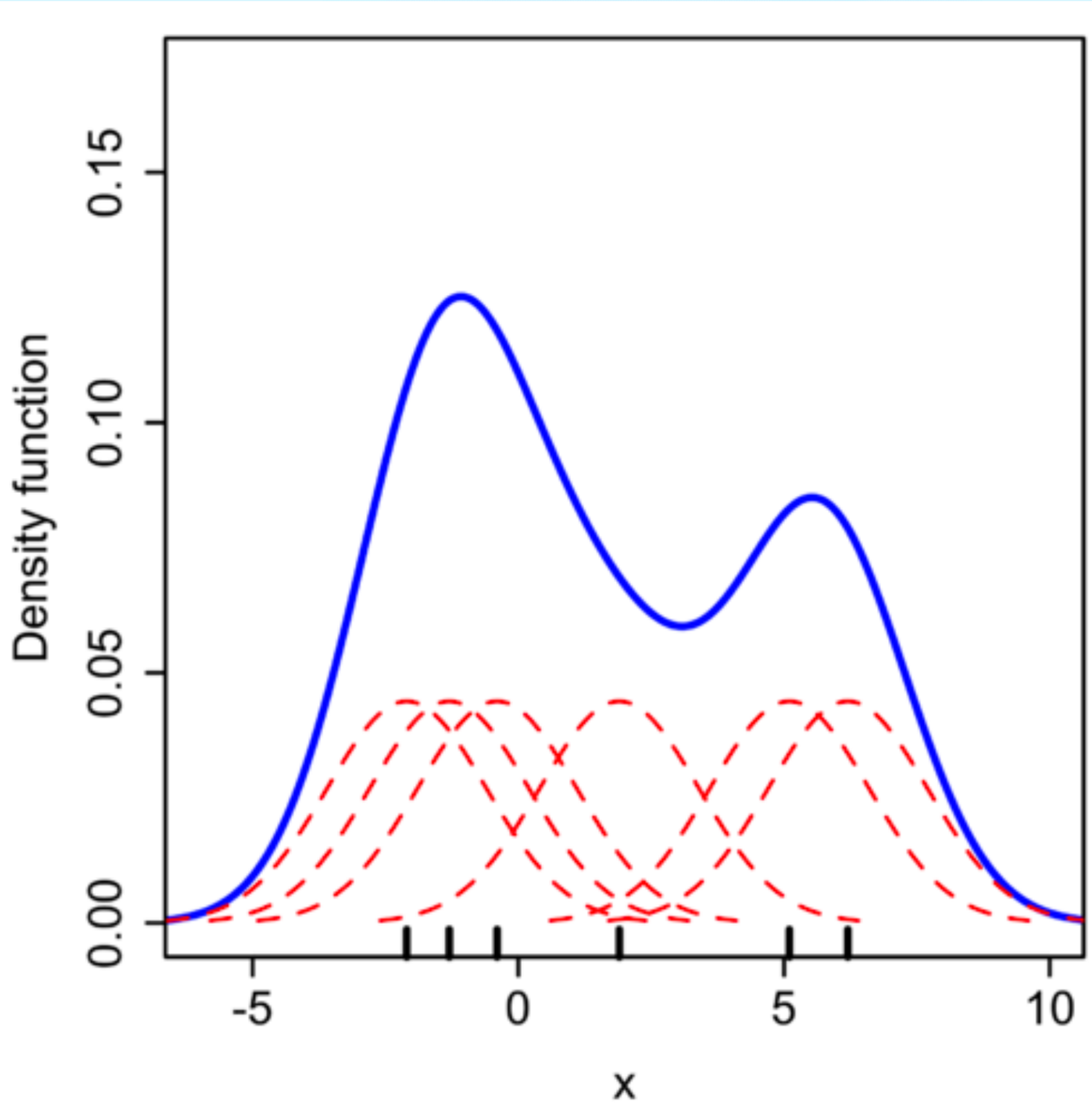
scipy.stats.gamma.pdf(x, a = α, loc, scale = 1/β)

Estimating the pdf

- If one has a distribution can one estimate the pdf it came from?
- Yes, but there will be a great deal of uncertainty, just as a pdf will create many different distributions, a single distribution could come from many different pdfs.
- One way to generate ‘a’ pdf is using a kernel density estimator (kde). This basically takes every bin in your histogram and replaces it with a function called the kernel or a window function. This function can very, one choice is the gaussian.
- There is still a free parameter, the width of that function. But making a choice for that, one can then generate a pdf for you histogram.

Kernel Density Estimator (KDE)





Kernel Density Estimator (KDE)

- However, the function will differ if a different width is chosen or a different kernel is used.
- The resulting function should be thought of as illustrative rather than a definitive answer.
- One could also solve for the best fitting of one of our known pdfs, a topic we will return to later.

Cumulative Density Functions

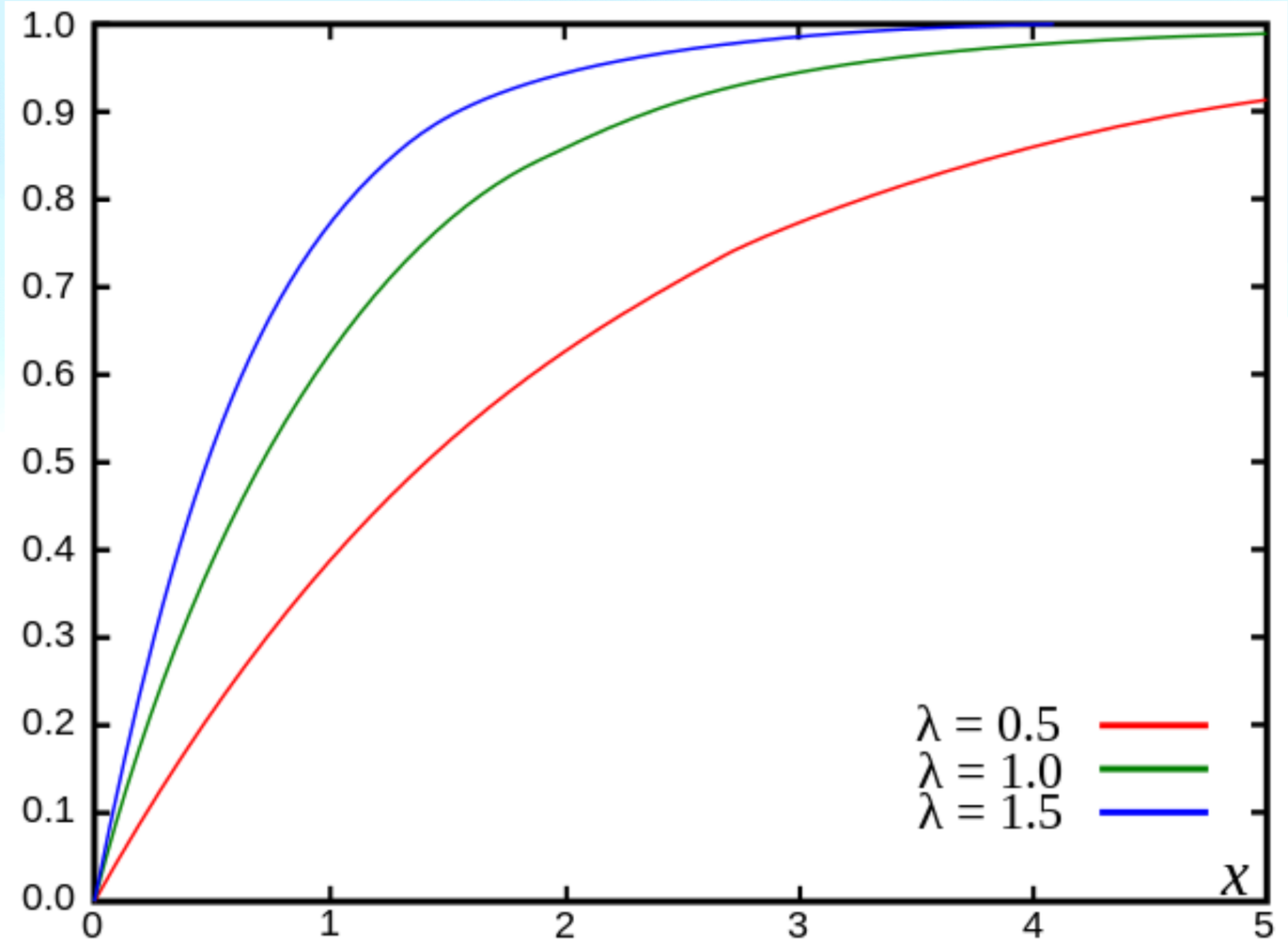
CDFs

- The cumulative density function is the integral of the probability density function. So if the pdf is given by p(x), the cdf is

$$P(x \leq a) = \int_{-\infty}^{a} p(x)dx$$

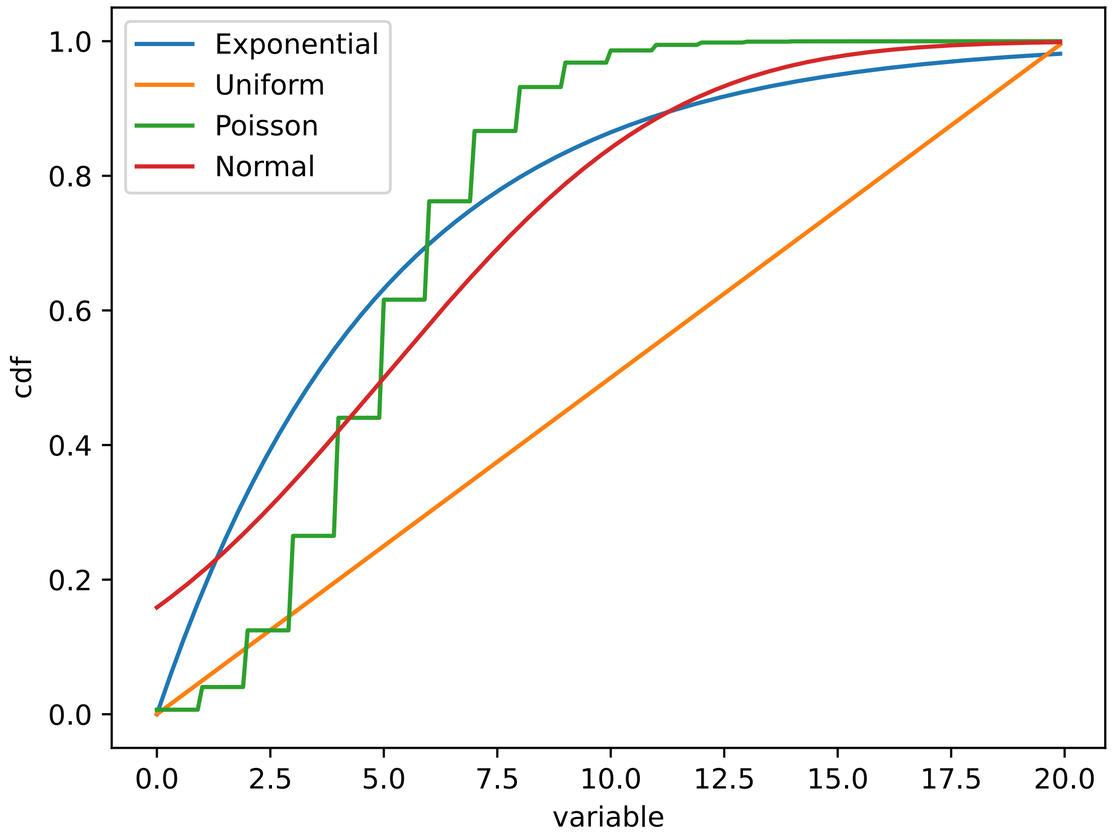
- Note if the probability density is only defined over a finite range then the cdf starts at the lower limit.

- The cdf is an alternative way of looking at the probability density distribution.



CDF

- The main advantage of the CDF is that it immediately shows one useful information, e.g. 25% of the distribution is below a, 75% is below b.
- However, the CDF always goes from 0.0 to 1.0 over the range of the distribution. For the same range, the CDF of different distributions is not all that different.
- This can make is a nice way to show different distributions since they will all be on the same scale.



scipy.stats

- To calculate the cdf of almost any know function we can make use of spicy.stats again.
- Now we just use cdf at the end to get the cumulative distribution. Note you can pass an array. So for example

```
v = np.linspace(-3, 3, num=100)
cdf = scipy.stats.norm.cdf(v, scale=2.0)
```

- Would give the cumulative probability for the normal distribution with a standard deviation of 2.

Plotting cdfs

- To plot the cumulative distribution of data, you can just use matplotlib.pyplot.hist(x, cumulative = True)
- Or in seaborn either seaborn.ecdfplot() or seaborn.displot(kind='ecdf'). The e is for empirical.
- To just calculate the values without plotting use numpy.histogram() and then numpy.cumsum() to add up your histogram.

```
h, x_edges = np.histogram(v, N=100)
cdf = np.cumsum(h)
```