Lecture 2.2

Probability and Random Numbers

Contents

1 Probability Basics

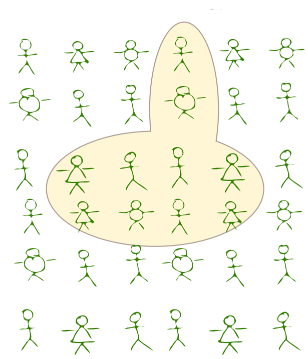
1.1 Population and Sample

A population is the set of **all entities** in a group.[1]

Examples:

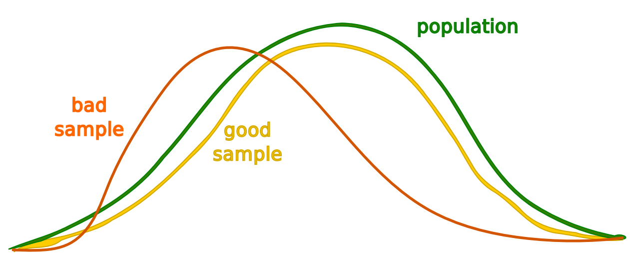
- American males,
- US voters
- stars in our galaxy

A sample is a **subset** of the population that is observed



A key goal in statistics is to **make inferences about the population from a sample**.

Constructing a good or "representative" sample is crucial.[2]



Bad: using NBA players to estimate the height of an American male

Bad: using New York voters to predict the outcome of a presidential race.

1.2 Random Variable

A random variable is a **well-defined attribute** of entities in a population.

Examples: population : random variable

- American males : height

[1] that are the object of a statistician's interest

[2] but not easy!

- stars in our galaxy : mass

A 1D random variable may be **discrete** or **continuous**.

Examples:

- roll of a die (discrete)
- single or married (discrete)
- number of children per household (discrete)
- person's height or weight (continuous)
- finish times at a marathon (continuous)
- fraction of women in a population (continuous)

1.3 Probability Distribution Function

The distribution of random variables can be described by the *probability distribution function*, or the *probability density function*. Luckily the abbreviation for both of them is the same - PDF.

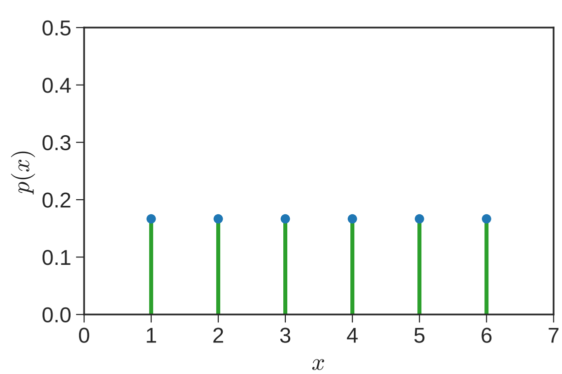
For example, the roll of a fair die may be described by the following PDF.

$$p_X(x) = \frac{1}{6}, \quad x \in \{1, 2, .., 6\} \tag{1}$$

Here X is the random variable. A discrete PDF $p_X(x)$ specifies the probability that the random variable $X = x$,

$$p_X(x) = \text{Pr}(X = x). \tag{2}$$

The PDF in eqn 1 says that the probability of each of the 6 possible outcomes is equal. Thus, the probability $\text{Pr}(X = 3) = \text{Pr}(X = 1) = 1/6$.



Often, the subscript "X" on $p_X(x)$ is suppressed for brevity, and the PDF is simply written as $p(x)$.[3] For this particular PDF note that the sum

$$\sum_{x=1}^{x=6} p(x) = 6 \cdot \frac{1}{6} = 1.$$

[3] Sometimes we use $f(x)$ or $\pi(x)$ to denote a PDF.

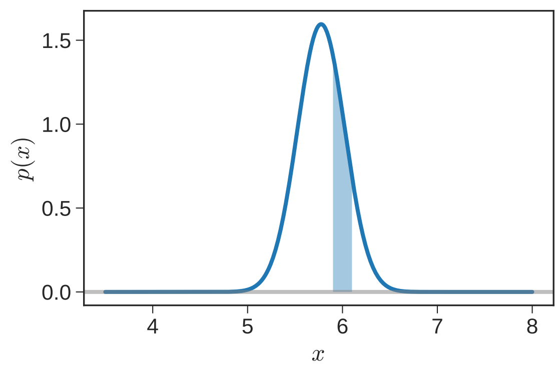
This is a *normalized* discrete PDF: probabilities over all possibilities add up to one.

As an example of a *continuous* PDF, consider the heights of American males or females. It can be described by a *normal* distribution:[4]

$$p_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad x \in (-\infty, \infty) \tag{3}$$

The probability of the height X lying between x and $x + dx$ is given by $p_X(x)\,dx$,

$$p_X(x)\,dx = \text{Pr}(x \le X < x + dx).^5 \tag{4}$$



The area of the shaded region is equal to the probability that the height of a randomly chosen individual from this population is between 5.9 and 6.1 feet,

$$\text{Pr}(5.9 \le X \le 6.1) = \int_{5.9}^{6.1} p(x)\,dx. \tag{5}$$

The *normalization* condition for continuous PDFs is

$$\int_{-\infty}^{\infty} p(x)\,dx = 1. \tag{6}$$

In other words, the area under the curve is one. The corresponding normalization condition for discrete PDFs is

$$\sum_{x=-\infty}^{+\infty} p(x) = 1.$$

Exercise: See problem (v) in Thought Questions.

1.4 Cumulative Distribution Function

The *cumulative distribution function* or CDF $F_X(x)$ represents the probability that the random variable $X \le x$.

$$\text{Pr}(X \le x) = F_X(x) \tag{7}$$

[4] actual data for US men fits $\mu = 69.3$, and $\sigma = 3.0$ inches, while for women, $\mu = 64.0$ and $\sigma = 3.0$ inches

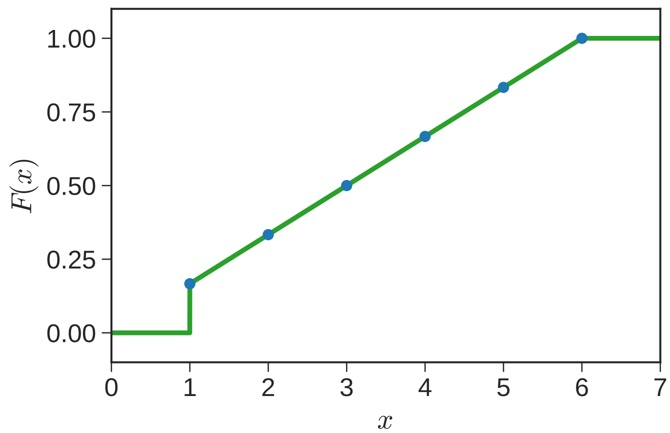
[5] again, the subscript "X" is often suppressed

For a discrete PDF $p(x)$,

$$F_X(x) = \sum_{x'=-\infty}^{x} p_X(x')$$

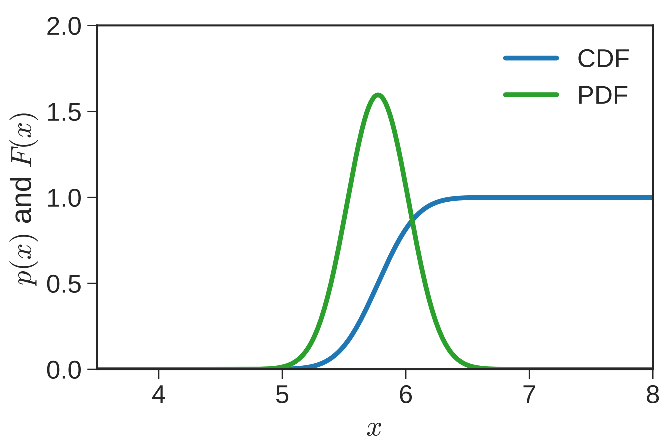
For the roll of a die:[6]

$$F_X(x) = \begin{cases} 0 & x < 1 \\ x/6 & 1 \le x \le 6 \\ 1 & x > 6 \end{cases}$$



For a continuous PDF,

$$F_X(x) = \int_{-\infty}^{x} p_X(x')\, dx'$$



1.5 Relation Between CDF and PDF

We can get the PDF from the CDF, and vice versa.

	PDF $\rightarrow$ CDF	CDF $\rightarrow$ PDF
discrete	$F(x) = \sum_{x'=-\infty}^{x} p(x')$	$p(x) = F(x) - F(x-1)$
continuous	$F(x) = \int_{-\infty}^{x} p(x')\, dx'$	$p(x) = \dfrac{dF(x)}{dx}$

[6]again, the subscript "X" is often suppressed

Exercise: What is the CDF of a continuous uniform distribution?

$$p_X(x) = 1, \quad 0 \leq x < 1. \tag{8}$$

1.6 Expected Values

The expected value of a function $g(x)$ of a random variable x drawn from a probability distribution $p(x)$ is given by,

$$E_p[g] = \begin{cases} \sum\limits_{-\infty}^{\infty} g(x)\,p(x) & \text{if } x \text{ is discrete} \\ \int_{-\infty}^{\infty} g(x)\,p(x)\,dx & \text{if } x \text{ is continuous} \end{cases} \tag{9}$$

Often, the subscript indicating the distribution is dropped for brevity. Two special cases are the mean and variance. They are related to the first and second moments of the PDF. The **mean** or **expected value** of a random variable,

$$E_p[x] = \begin{cases} \sum\limits_{-\infty}^{\infty} x\,p(x) & \text{if } x \text{ is discrete} \\ \int_{-\infty}^{\infty} x\,p(x)\,dx & \text{if } x \text{ is continuous} \end{cases} \tag{10}$$

The **variance** of the random variable is given by,

$$V_p[x] = E_p[x^2] - E_p[x]^2. \tag{11}$$

Exercise: What is the mean and variance of the PDF given by eqn 8?

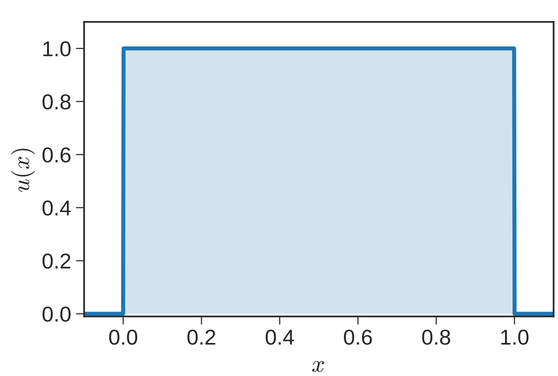
2 Pseudo-Random Number Generators

We shall first discuss the generation of pseudo-random numbers $x \in [0, 1]$ from the continuous uniform distribution. Once we have these, we can sample other distributions.

2.1 Uniform Random Number Generators

If you did the exercises in the notes, you have already met the continuous uniform distribution. It is given by,

$$u_X(x) = \begin{cases} 1, & 0 \leq x < 1 \\ 0, & \text{elsewhere} \end{cases}$$



Our goal now is to *generate* a random number X that is uniformly distributed between 0 and 1. What does uniform mean here? If we generate 1000 random numbers, approximately 10% or 100 should lie in any arbitrary interval of width 0.1.

The basic idea is to "toss a coin" for a (say) 32-bit binary number, so that each of the 2^{32} possibilities

$$\begin{array}{c} 0000\ldots000 \\ 0000\ldots001 \\ 0000\ldots010 \\ 0000\ldots011 \\ \vdots \\ 1111\ldots111 \end{array}$$

is visited in an "apparently" random fashion. Usually random numbers used in Monte Carlo simulations are based on a deterministic algorithm, and are hence called "pseudo"-random.

2.1.1 Linear Congruential Generators

The simplest RNGs are linear congruential sequence RNGs. They involve multiplication and truncation of leading bits of an integer.

$$n_{i+1} = (a n_i) \mod m, \tag{12}$$

where n_i is an integer, a is the *multiplier*, and m is the *modulus*. $x \mod y$ is the modulo or remainder operator, *e.g*, $8 \mod 3 = 2, 5 \mod 5 = 0$.

To get a **real number** between 0 and 1, we compute n_i/m, which is guaranteed to be less than 1. See the python program `LinCongGen(a, m, n0, size)` in section A.3.

n_0, the initial seed, has to be supplied. Thus, a particular choice of a and m specify a particular method. Not all choices of a and m result in a good RNG. It is useful to illustrate a bad choice for these parameters. Consider $a = 3$ and $m = 7$.

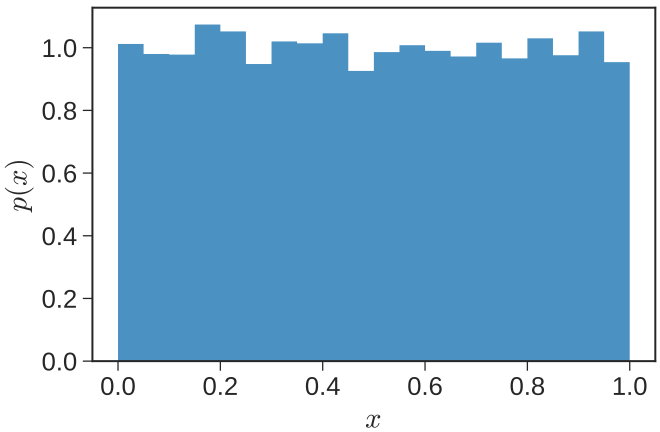
```
print(LinCongGen(a=3, m=7, n0=12, size=50))

[12  1  3  2  6  4  5  1  3  2  6  4  5
     1  3  2  6  4  5  1  3  2  6  4  5
     1  3  2  6  4  5  1  3  2  6  4  5
     1  3  2  6  4  5  1  3  2  6  4  5
     1]
```

Notice that the sequence 1, 3, 2, 6, 4, 5 is periodic. Using a different initial seed $n_0 = 8$, doesn't help.

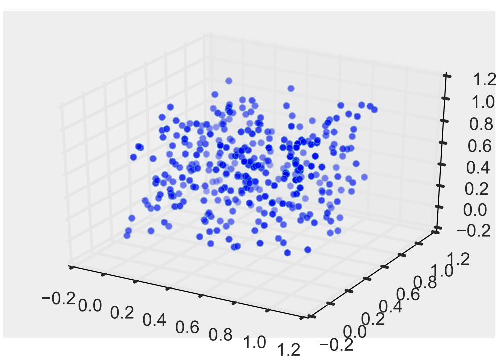
```
print(LinCongGen(a=3, m=7, n0=8, size=50))

[8  3  2  6  4  5  1  3  2  6  4  5
 1  3  2  6  4  5  1  3  2  6  4  5
 1  3  2  6  4  5  1  3  2  6  4  5
 1  3  2  6  4  5  1  3  2  6  4  5
 1]
```

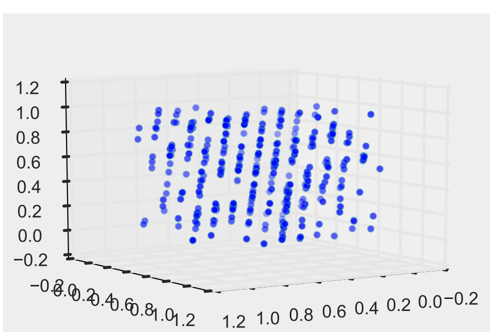
In general, we choose an m that is very large to delay this periodic behavior as much as possible. Usually, we set m close to the maximum integer representable on the computer.

Consider a better choice: $a = 16807$ and $m = 2147483647$. Note I am converting the integers to real numbers by dividing n_i by m. A histogram of 10,000 numbers is approximately flat.



Flat histograms, while a useful diagnostic measure, are not sufficient. For example, a famous disastrous bad choice is the RANDU RNG, which used $a = 65539$ and $m = 2^{31}$. The histogram looks flat. However, if you plot three successive deviates in 3D, you observe a disturbing pattern, when you look at it from the "right" angle.





This was first reported by fellow Seminole George Marsaglia (1924-2011) who was a faculty member in Statistics, and a predecessor of this department.[7] His work also helped us address the question

[7]"Random Numbers Fall Mainly in the Planes", *PNAS* **1968**

"What makes for a good RNG?", through the so called "Diehard battery of Tests for Randomness". The history of RNGs is quite fascinating.[8] Good RNGs include the following:

- Mersenne-Twister
- SIMD-oriented Fast Mersenne-Twister
- Well Equidistributed Long-period Linear (WELL)
- Xorshift

Most modern libraries use sophisticated RNGs. Furthermore, libraries like `scipy.stats` allow you to generate random numbers from various standard distributions like normal, Poisson etc. (see sec. A.2). In any case, as long as you have a decent uniform random number generator, you can generate random numbers from any other distribution.

3 Standard 1D Distributions

Let us catalog some standard 1D distributions:

Discrete	**Continuous**
Uniform	Uniform
Binomial	Gaussian
Poisson	Exponential

3.1 Discrete Distributions

3.1.1 Discrete Uniform Distribution

We have seen an example (roll of a die) of this before. More generally,

$$f(x) = \frac{1}{n}, \quad x \in \{a_1, a_2, .., a_n\}$$

where a_i are the (exhaustive set of) $i = 1, 2, ..., n$ different possible outcomes.

Each outcome is equally likely. A fair die or coin is one example. Can you think of other examples?

3.1.2 Binomial Distribution

Consider N trials of an experiment with possible outcomes "success" or "failure" ("heads" or "tails", 0 or 1 etc.) Suppose the probability of "success" is p.

The discrete random number n is the number of successes in N trials. The probability of a particular outcome say 'ssffs',

$$\text{Pr[`ssffs']} = pp(1-p)(1-p)p = p^3(1-p)^2$$

Suppose, the order of successes and failures is unimportant (ssffs $\leftrightarrow$ sssff $\leftrightarrow$ sfsfs $\leftrightarrow$ etc.). Then the number of ways to generate n successes from N trials is,

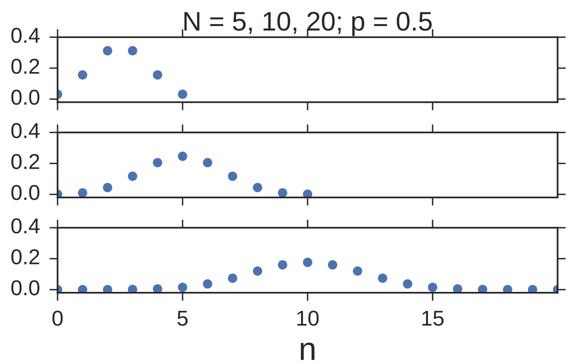
$$^N C_n = \frac{N!}{(N-n)!\, n!}$$

[8] P. L'Ecuyer, "History of uniform random number generation," 2017 Winter Simulation Conference, **2017**.

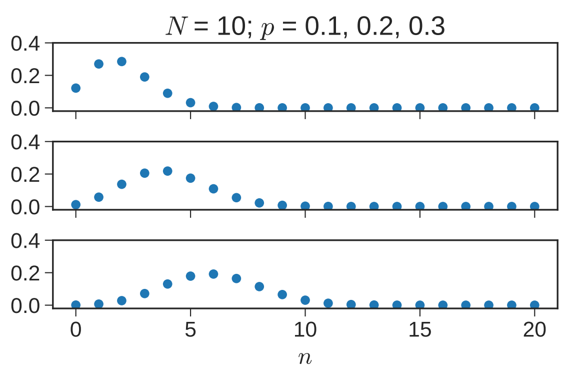
Therefore, the binomial distribution of n successes in N trials, when p is the probability of success is given by

$$f(n; N, p) = \frac{N!}{(N-n)!n!} p^n (1-p)^{N-n}. \tag{13}$$

We can plot this distribution for different N at constant p. For large enough Np, the binomial distribution starts to look like a normal distribution.



Or, increase p, keeping N constant



Exercise: Show that $E_f[x] = Np$ and $V_f[x] = Np(1-p)$.[9]

3.1.3 Poisson Distribution

Wikipedia has a succinct description of the Poisson distribution:

> "the Poisson distribution expresses the probability of a given number of events occurring in a fixed interval of time and/or space if these events occur with a known average rate and independently of the time since the last event."

Examples: the number of

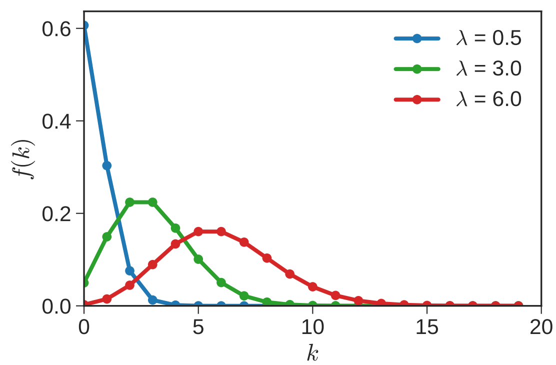
- phone calls received by a call center per hour
- taxis passing a particular street corner per hour

[9] see solution

- decay events per second from a radioactive source
- homicides in a city in a year

The probability of observing k events in an interval where the average rate is λ is given by,

$$f(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}, \quad k = 0, 1, ..., \infty \tag{14}$$



The mean and variance of the Poisson distribution are both equal to λ.

The Poisson distribution is related to the binomial distribution in the limit of $N \to \infty$, $p \to 0$, but mean $Np = \lambda$. For derivation see appendix sec. A.1.

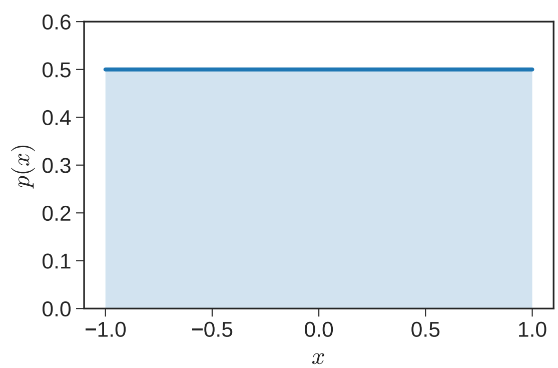
3.2 Continuous Distributions

3.2.1 Continuous Uniform Distribution

A general continuous uniform distribution function has the form:

$$p(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b \\ 0 & \text{elsewhere.} \end{cases}$$

As a particular example consider $a = -1$, and $b = 1$.



Example: Calculate the mean $E_p[x]$ of a continuous uniform distribution.

$$\begin{aligned} E_p[x] &= \int_a^b x\, p(x)\, dx \\ &= \int_a^b x\, \frac{1}{b-a}\, dx \\ &= \frac{b^2 - a^2}{2} \frac{1}{b-a} = \frac{a+b}{2} \end{aligned}$$

Exercise: Calculate the variance $V_p[x]$ by first computing the integral,

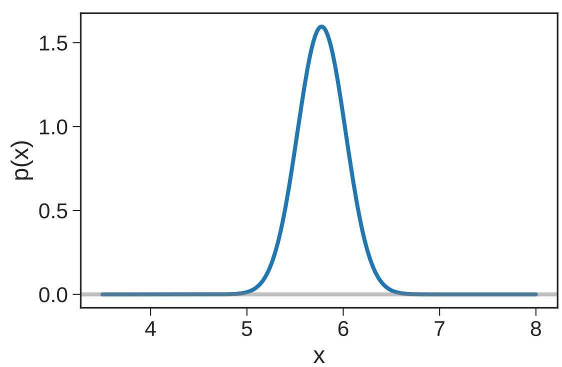
$$E_p[x^2] = \int_a^b x^2\, p(x)\, dx.$$

3.2.2 Gaussian or Normal Distribution

We have seen the Gaussian distribution before,

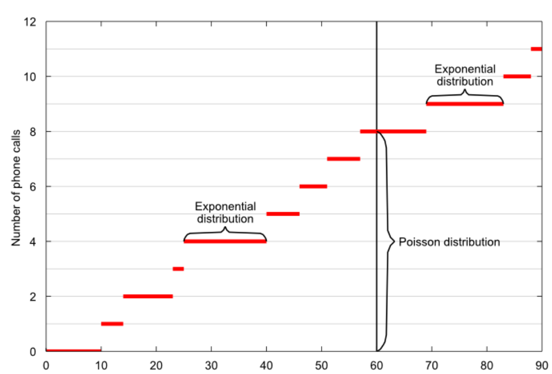
$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad x \in (-\infty, \infty)$$

The variance is σ^2 and the mean is μ.



3.2.3 Exponential Distribution

The exponential distribution is closely related to the Poisson distribution. The figure below visually explains the relationship.



The exponential distribution describes the time between events in a Poisson process, i.e. a process in which events occur continuously and independently at a constant average rate.

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0, \\ 0 & x < 0. \end{cases} \tag{15}$$

Example: Suppose you am waiting for a Seminole bus at a stop. Let's say that typically a bus arrives at the stop every 5 mins. This sets the average arrival rate $\lambda = 1/5\,\text{min}^{-1}$ in both the Poisson and exponential distributions (eqns 14 and 15).

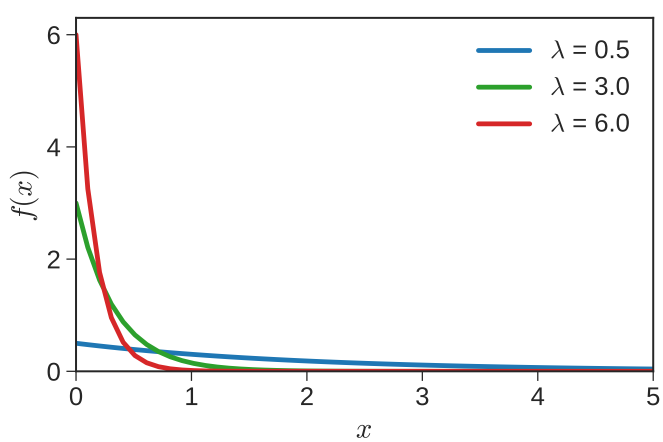
Suppose you want to calculate the probability that no bus arrives over the next minute. You can do this using either eqn 14 or 15.

According to the Poisson distribution, the probability of 0 arrivals in the next minute is $f(k = 0, \lambda = 0.2) = 0.8187$. Similarly according to the exponential distribution, the probability of having to wait for more than 1 minute for the bus is,

$$\int_1^\infty f(x,\lambda)\,dx = e^{-\lambda} = 0.8187.$$

Exercise: Show that the mean and the variance of the exponential distribution are $1/\lambda$ and $1/\lambda^2$, respectively.

Note: Consider the means from the Poisson and exponential distributions in the bus example, above. For Poisson, the mean is the average number of buses arriving per minute $E(k) = \lambda$ = 0.2 buses. For exponential, the average waiting time for a bus to arrive is $E(x) = 1/\lambda$ = 5 mins.



3.3 Lists of Probability Distributions

- NIST has a list of commonly encountered distributions
- Wikipedia has a fairly comprehensive list
- SciPy not only has a comprehensive list, but also has a convenient common interface for plotting, sampling, getting statistics etc.

4 Problems

4.1 Thought Questions

(i) What is the maximum value of a discrete PDF?

(ii) True or False: The maximum value of a continuous PDF is one.

(iii) Given a PDF $p(x)$ and a constant a, show that $E[ax] = aE[x]$ and $V[ax] = a^2 V[x]$. **Hint**: use eqns 9 and 10.

(iv) Consider a triangular distribution $p(x)$ with a peak at zero, that is nonzero on the domain $x \in [a, b]$, with $a < 0$ and $b > 0$.

$$p(x) = \begin{cases} \frac{2}{(b-a)} \left(1 - \frac{x}{a}\right) & \text{for } a \leq x < 0, \\ \frac{2}{(b-a)} \left(\frac{x}{b} - 1\right) & \text{for } 0 \leq x \leq b. \end{cases} \tag{16}$$

(a) Test whether this distribution is normalized

(b) Find the CDF corresponding to this distribution

(c) Sketch the PDF and the CDF

(d) What is the mean and variance of this distribution

(v) Consider a continuous power law distribution which is nonzero over $x \in [1, \infty]$,

$$p(x) = (\alpha - 1)x^{-\alpha}, \qquad \alpha > 1. \tag{17}$$

(a) Test whether this distribution is normalized

(b) Find the CDF corresponding to this distribution

(c) **Power laws can be weird**: Show that the mean of this distribution is infinite when $1 < \alpha < 2$. For $2 < \alpha < 3$ show that the mean is finite, but the variance is infinite.

(d) Think of practical (or research) implications of such fat-tailed distributions?

(vi) Suppose you are planning on running a parallel job on an HPC with $n = 1024$ CPUs. Let the average failure rate for a processor be $\lambda = 0.01/$ year. Your program is not fault tolerant, so it fails if any of the the n (independent) CPUs fails. On average, how many days do you expect your job to run before encountering a problem.

4.2 Numerical Problems

(i) Shuffling a Deck of Cards

Consider a standard deck of cards. For convenience, let us label the cards from 1 through 52 (or 0 through 51).

One may store the state of a deck using a list or array, e.g. deck = [5, 21, 16, ..., 50, 1].

Describe and implement an algorithm to shuffle the cards so that the i^{th} card is uniformly distributed,

$$p(\text{deck[i]} = 1) = p(\text{deck[i]} = 2) = ... = p(\text{deck[i]} = 52) = 1/52, \tag{18}$$

for $1 \leq i \leq 52$. You are not allowed to use to built-in function for random permutations of lists for this question.

Devise and perform 1 test to verify that your algorithm works.

(ii) Bootstrapping

Given an N dimensional array $\mathbf{a}$, construct another N dimensional array $\mathbf{a}'$ in which the elements of $\mathbf{a}$ are randomly chosen *with replacement*.

For example, if,

$$\mathbf{a} = [11, 22, 33, 44, 55, 66, 77, 88, 99, 110],$$

then, perhaps,

$$\mathbf{a}' = [22, 44, 55, 22, 11, 77, 110, 99, 88, 11].$$

Note that some elements from the original array are repeated, while some are not represented. Such an array or sample is called a *bootstrap sample*, and it can be used to find confidence intervals.

(a) Sketch out your proposed strategy or algorithm (in words)

(b) Implement it in a programming language of your choice

(iii) Bootstrap Application

Suppose a video game company suggests an educational intervention to improve student math skills. The intervention is tested in a small class with $n = 10$. The results are tabulated below:

status	**# students**	**score**
improve	2	+1
same	7	0
worsen	1	-1

There seems to have been net-improvement $s = (2 \times +1) + (7 \times 0) + (1 \times -1) = +1 > 0$.

We want to assess the robustness of the claim "there is net improvement due to the intervention", by generating 10,000 bootstrap samples.

Hint: Think of the original sample as $\mathbf{a} = \{$+1, +1, -1, 0, 0, 0, 0, 0, 0, 0$\}$, where "+1 = improve", "0 = same", and "-1 = worsen".

(a) In what fraction of the bootstrap samples is the net-score $s > 0$?

(b) How would your answer change if the original class size had been $n = 100$, and the numbers in the second column had been multiplied by 10?

(c) Stay with the $n = 100$ case. Suppose the intervention costs money, and we are only interested in it, if the measured mean positive effect, $s/n > +0.05$. What would the bootstrap analysis suggest?

(iv) LCG Random Number Generator (from 2021 quiz)

Consider a linear congruential generator random number generator (RNG) to sample uniform random numbers ($u_i = n_i/m$).[10]

$$n_{i+1} = (an_i) \mod m. \tag{19}$$

Assume that the initial seed $n_0 = 1$ for all the problems below. You may use a computer, but report all requested work on your answer sheet.

(i) Suppose $m = 2^k$ with $k = 3$.

(a) Generate and report the first 10 random numbers $n_1, n_2, \cdots, n_{10}$ using $a = 2$.

(b) Repeat the calculation for any other even $a > 2$.

(c) Summarize your observations: what happens to the sequence when a is even?

(ii) Thus, it might be advisable to set a = odd number.

(a) Generate and report the first 10 random numbers $n_1, n_2, \cdots, n_{10}$ using $a = 3$.

(b) Repeat the calculation for any other odd $a > 3$.

(c) What is the period (length of non-repeating sequence) of the RNG?

(iii) What is the period for $a = 3$ and $m = 2^4$, and $m = 2^5$?

(iv) Claim: "the period of a LCG RNG with $m = 2^k$ and odd a is 2^{k-2}". Do your numerical experiments support this claim?

(v) Based on the claim above, what is the period of the RANDU RNG? Is it more than 1 billion?

A Appendix

A.1 Poisson Distribution from Binomial Distribution

Poisson distribution is useful for modeling occurrence of rare events.

It can be derived from the binomial distribution,

$$f(n; N, p) = \frac{N!}{(N-n)!n!} p^n (1-p)^{N-n}$$

in the limit of $N \to \infty$, $p \to 0$, but mean $Np = \lambda$.[11]

When $N \to \infty$,

$$\frac{N!}{(N-n)!n!} = \frac{N(N-1)\cdots(N-n+1)}{n!} \approx \frac{N^n}{n!} \tag{20}$$

[10] Typically a is an odd integer, and $m = 2^k$, where $k \sim 32$ or 64 for single and double precision numbers, respectively.

[11] Intuitively, we can think of dividing a unit of time into N subintervals with a arrival probability of p per subinterval. Then the expected number of arrivals in unit time are $\lambda = Np$. The large N requirement pertains to the continuum approximation, while the small p requirement avoids clustering of successes.

Since $N = \lambda/p$, we have

$$\begin{aligned}\frac{N!}{(N-n)!n!}p^n(1-p)^{N-n} &= \frac{N^n}{n!}p^n(1-p)^{N-n} \\ &\approx \frac{\lambda^n}{n!}(1-p)^{\lambda/p} \\ &= \frac{\lambda^n}{n!}((1-p)^{1/p})^{\lambda} \\ &= \frac{\lambda^n}{n!}e^{-\lambda}\end{aligned}$$

where we used the formula,

$$\lim_{p\to 0}(1-p)^{1/p} = e^{-1},$$

in the last step. Thus, the Poisson distribution is given by,

$$f(n;\lambda) = \frac{\lambda^n e^{-\lambda}}{n!}, \quad n = 0, 1, ..., \infty.$$

As a rule of thumb when $N > 20$ and $p < 0.05$, the two distributions become nearly indistinguishable.

A.2 Standard Distributions in SciPy

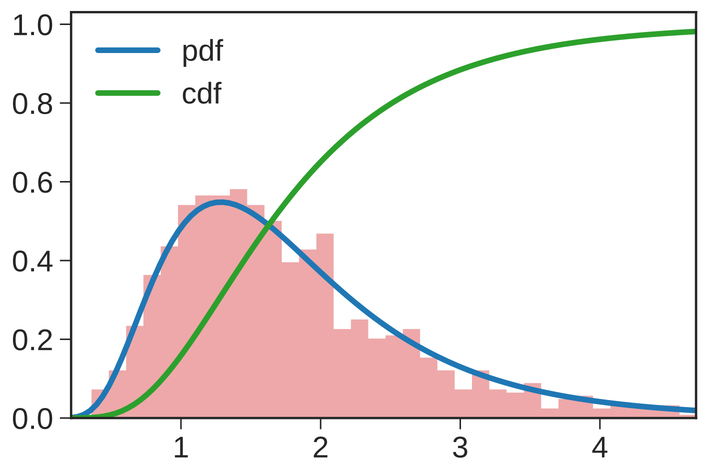
Scipy has a convenient interface to plot and sample from a variety of distributions. As an example ,consider the lognormal distribution:

$$f(x;\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}}\, e^{-\frac{(\ln x-\mu)^2}{2\sigma^2}} \tag{21}$$

Some initial setup to make sure distributions are described similarly.

```
from scipy.stats import lognorm

# standard distribution specfiers
sigma = 0.5
mu    = 0.5

# SciPy translation (from documentation)
s     = sigma
loc   = 0.
scale = np.exp(mu)

# ppf = inverse of cdf
# finds relevant range over which PDF is nonzero
x = np.linspace(lognorm.ppf(0.001, s),
                lognorm.ppf(0.999, s), 100)
plt.xlim(min(x), max(x))

# plot CDF and PDF
```

```
plt.plot(x, lognorm.pdf(x, s, loc, scale), lw=3, label='pdf')
plt.plot(x, lognorm.cdf(x, s, loc, scale), lw=3, label='cdf')

# sample from distribution
r = lognorm.rvs(s, loc, scale, size=1000)
plt.hist(r, 50, density=True, histtype='stepfilled', alpha=0.4)
plt.legend(loc='upper left')
```



A.3 Python Code

A.3.1 Linear Congruential Generator

```
def LinCongGen(a, m, n0, size):
        # a    = multiplier
        # m    = modulus
        # n0   = initial seed
    # size = size of array of random integers returned

    n = np.zeros(num, dtype=int)
    n[0] = n0

    for i in range(1,size):
        n[i] = a * n[i-1] % m # % = modulo operator

    return n
```

A.3.2 Binomial Distribution

```
def binomialdist(N, p):
    from scipy.misc import comb

    n = np.arange(0, N+1)
    f = np.zeros((n.shape))
```

```
    for i in np.arange(0, N+1):
        f[i] = comb(N, n[i]) * p**n[i] *
               (1-p)**(N-n[i])

    return n, f
```

A.3.3 Poisson Distribution

```
def PoissonDist(lam, kvec):
    from math import factorial, exp

    fvec = np.zeros(kvec.shape)
    i    = 0
    for k in kvec:
        fvec[i] = lam**k * exp(-lam)/factorial(k)
        i = i + 1
    return fvec
```