Poisson random variables will be the third main discrete distribution that we expect you to know well. After introducing Poisson, we will quickly introduce three more. I want you to be comfortable with being told the semantics of a distribution, given the key formulas (for expectation, variance and PMF) and then using it.

**Binomial in the Limit**

Recall example of sending bit string over network. In our last class we used a binomial random variable to represent the number of bits corrupted out of four with a high corruption probability (each bit had independent probability of corruption \( p = 0.1 \)). That example was relevant to sending data to space craft, but for earthly applications like HTML data, voice or video, bit streams are much longer (length \( \approx 10^4 \)) and the probability of corruption of a particular bit is very small (\( p \approx 10^{-6} \)). Extreme \( n \) and \( p \) values arise in many cases: # visitors to a website, #server crashes in a giant data center.

Unfortunately \( X \sim \text{Bin}(10^4, 10^{-6}) \) is unwieldy to compute. However when values get that extreme we can make approximations that are accurate and make computation feasible. Recall the Binomial distribution. First define \( \lambda = np \). We can rewrite the Binomial PMF as follows:

\[
P(X = i) = \frac{n!}{i!(n-1)!} \left( \frac{\lambda}{n} \right)^i \left( 1 - \frac{\lambda}{n} \right)^{n-i}
\]

This equation can be made simpler by observing how some of these equations evaluate when \( n \) is sufficiently large and \( p \) is sufficiently small. The following equations hold:

\[
\frac{n(n-1)\ldots(n-i-1)}{n^i} \approx 1
\]

\[
(1 - \lambda/n)^n \approx e^{-\lambda}
\]

\[
(1 - \lambda/n)^i \approx 1
\]

This reduces our original equation to:

\[
P(X = i) = \frac{\lambda^i}{i!} e^{-\lambda}
\]

This simplification turns out to be so useful, that in extreme values of \( n \) and \( p \) we call the approximated Binomial its own random variable type: the Poisson Random Variable.

**Poisson Random Variable**

A Poisson random variable approximates Binomial where \( n \) is large, \( p \) is small, and \( \lambda = np \) is “moderate”. Interestingly, to calculate the things we care about (PMF, expectation, variance) we no longer need to know \( n \) and \( p \). We only need to provide \( \lambda \) which we call the rate.

There are different interpretations of "moderate". The accepted ranges are \( n > 20 \) and \( p < 0.05 \) or \( n > 100 \) and \( p < 0.1 \).
Here are the key formulas you need to know for Poisson. If $Y \sim Poi(\lambda)$:

$$P(Y = i) = \frac{\lambda^i}{i!}e^{-\lambda}$$

$$E[Y] = \lambda$$

$$Var(Y) = \lambda$$

**Example**

Let’s say you want to send a bit string of length $n = 10^4$ where each bit is independently corrupted with $p = 10^{-6}$. What is the probability that the message will arrive uncorrupted? You can solve this using a Poisson with $\lambda = np = 10^4 \times 10^{-6} = 0.01$. Let $X \sim Poi(0.01)$ be the number of corrupted bits. Using the PMF for Poisson:

$$P(X = 0) = \frac{\lambda^0}{0!}e^{-\lambda} = \frac{0.01^0}{0!}e^{-0.01} \approx 0.9990498$$

We could have also modelled $X$ as a binomial such that $X \sim Bin(10^4, 10^{-6})$. That would have been computationally harder to compute but would have resulted in the same number (up to the millionth decimal).

**Geometric Random Variable**

$X$ is Geometric Random Variable: $X \sim Geo(p)$ if $X$ is number of independent trials until first success and $p$ is probability of success on each trial. Here are the key formulas you need to know. If $X \sim Geo(p)$:

$$P(X = n) = (1 - p)^{n-1}p$$

$$E[X] = 1/p$$

$$Var(X) = (1 - p)/p^2$$

**Negative Binomial Random Variable**

$X$ is Negative Binomial: $X \sim NegBin(r, p)$ if $X$ is number of independent trials until $r$ successes and $p$ is probability of success on each trial. Here are the key formulas you need to know. If $X \sim NegBin(p)$:

$$P(X = n) = \binom{n-1}{r-1}p^r(1 - p)^{n-r} \text{ where } r \leq n$$

$$E[X] = r/p$$

$$Var(X) = r(1 - p)/p^2$$

**Zipf Random Variable**

$X$ is Zipf: $X \sim Zipf(s)$ if $X$ is the rank index of a chosen word (where $s$ is a parameter of the language).

$$P(X = k) = \frac{1}{k^s \cdot H}$$

Where $H$ is a normalizing constant (and turns out to be equal to the $N$th harmonic number where $N$ is the size of the language.)