### Whither the Binomial…

- Recall example of sending bit string over network:
  - \( n = 4 \) bits sent over network where each bit had independent probability of corruption \( p = 0.1 \)
  - \( X \) = number of bit corrupted. \( X \sim \text{Bin}(4, 0.1) \)
  - In real networks, send large bit strings (length \( n = 10^4 \))
  - Probability of bit corruption is very small \( p = 10^{-6} \)
  - \( X \sim \text{Bin}(10^4, 10^{-6}) \) is unwieldy to compute
- Extreme \( n \) and \( p \) values arise in many cases:
  - \# bit errors in file written to disk (\# of typos in a book)
  - \# of elements in particular bucket of large hash table
  - \# of servers crashes in a day in giant data center
  - \# Facebook login requests that go to particular server

### Binomial in the Limit

- Recall the Binomial distribution:
  \[
P(X = i) = \frac{n!}{i!(n-i)!} p^i (1-p)^{n-i}
\]
- Let \( \lambda = np \) (equivalently: \( p = \lambda/n \))

\[
P(X = i) = \frac{n!}{i!(n-i)!} \left( \frac{\lambda}{n} \right)^i \left( 1 - \frac{\lambda}{n} \right)^{n-i} = \frac{n(n-1)...(n-i+1)}{n^i} \frac{\lambda^i}{i!} \left( 1 - \frac{\lambda}{n} \right)^{n-i}
\]
- When \( n \) is large, \( p \) is small, and \( \lambda \) is "moderate":

\[
\frac{n(n-1)...(n-i+1)}{n^i} \approx 1 \quad (1 - \lambda/n)^{n-i} \approx e^{-\lambda i/n} \quad (1 - \lambda/n)^{n-i} \approx 1
\]
- Yielding:

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

### Poisson Random Variable

- \( X \) is a **Poisson** Random Variable: \( X \sim \text{Poi}(\lambda) \)
  - \( X \) takes on values \( 0, 1, 2, ..., \)
  - and, for a given parameter \( \lambda > 0 \),
  - has distribution (PMF):

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

\[
e^\lambda = \frac{\lambda^0}{0!} + \frac{\lambda^1}{1!} + \frac{\lambda^2}{2!} + ... + \sum_{i=0}^{\infty} \frac{\lambda^i}{i!}
\]
- So:

\[
\sum_{i=0}^{\infty} P(X = i) = \sum_{i=0}^{\infty} e^{-\lambda} \frac{\lambda^i}{i!} = e^{-\lambda} \sum_{i=0}^{\infty} \frac{\lambda^i}{i!} = e^{-\lambda} e^\lambda = 1
\]

### Sending Data on Network Redux

- Recall example of sending bit string over network:
  - Send bit string of length \( n = 10^4 \)
  - Probability of (independent) bit corruption \( p = 10^{-6} \)
  - \( X \sim \text{Bin}(10^4, 10^{-6}) = 0.01 \)
  - What is probability that message arrives uncorrupted?

\[
P(X = 0) = e^{-\lambda} \frac{\lambda^0}{0!} = e^{-0.01} (0.01)^0 = 0.990049834
\]
- Using \( Y \sim \text{Bin}(10^4, 10^{-6}) \):

\[
P(Y = 0) \approx 0.990049829
\]

Caveat emptor: Binomial computed with built-in function in R software package, so some approximation may have occurred. Approximation are closer to you than they may appear in some software packages.

### Simeon-Denis Poisson

- Simeon-Denis Poisson (1781-1840) was a prolific French mathematician

- Published his first paper at 18, became professor at 21, and published over 300 papers in his life
  - He reportedly said “Life is good for only two things, discovering mathematics and teaching mathematics.”
  - Definitely did not look like Charlie Sheen

### Poisson Random is Binomial in Limit

- Poisson approximates Binomial where \( n \) is large, \( \lambda \) = \( np \) is "moderate"
  - Different interpretations of "moderate"
    - \( n > 20 \) and \( p < 0.05 \)
    - \( n > 100 \) and \( p < 0.1 \)
  - Really, Poisson is Binomial as

\[
n \to \infty \quad \text{and} \quad p \to 0, \text{ where } np = \lambda
\]
Bin(10, 0.3), Bin(100, 0.03) vs. Poi(3)

Tender (Central) Moments with Poisson

- Recall: \( Y \sim \text{Bin}(n, \rho) \)
  - \( \mathbb{E}[Y] = np \)
  - \( \text{Var}(Y) = np(1 - p) \)

- \( X \sim \text{Poi}(\lambda) \) where \( \lambda = np \) \( (n \to \infty \text{ and } p \to 0) \)
  - \( \mathbb{E}[X] = np = \lambda \)
  - \( \text{Var}(X) = np \) \( = \lambda \)
  - Yes, expectation and variance of Poisson are same
    - It brings a tear to my eye…
  - Recall: \( \text{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 \)
    - \( \mathbb{E}[X^2] = \text{Var}(X) + (\mathbb{E}[X])^2 = \lambda + \lambda^2 = \lambda(1 + \lambda) \)

It’s Really All About Raisin Cake

- Bake a cake using *many* raisins and *lots* of batter
- Cake is enormous (in fact, infinitely so…)
  - Cut slices of “moderate” size (w.r.t. # raisins/slice)
  - Probability \( p \) that a particular raisin is in a certain slice is very small (\( p \to 0 \))
- Let \( X \) = number of raisins in a certain cake slice
  - \( X \sim \text{Poi}(\lambda) \), where \( \lambda = \frac{\text{# raisins}}{\text{# cake slices}} \)

CS = Baking Raisin Cake With Code

- Hash tables
  - strings = raisins
  - buckets = cake slices
- Server crashes in data center
  - servers = raisins
  - list of crashed machines = particular slice of cake
- Facebook login requests (i.e., web server requests)
  - requests = raisins
  - server receiving request = cake slice

Defective Chips

- Computer chips are produced
  - \( p = 0.1 \) that a chip is defective
  - Consider a sample of \( n = 10 \) chips
  - What is \( P(\text{sample contains } \leq 1 \text{ defective chip}) \)?
    - Using \( Y \sim \text{Bin}(10, 0.1) \):
      \[
P(\text{\# defective} \leq 1) = \binom{10}{0} (0.1)^0 (1 - 0.1)^{10} + \binom{10}{1} (0.1)^1 (1 - 0.1)^9 
      \approx 0.7361
      \]
    - Using \( X \sim \text{Poi}(\lambda = 0.1)(10) = 1 \):
      \[
P(X \leq 1) = e^{-1} \frac{1^0}{0!} + e^{-1} \frac{1^1}{1!} = 2e^{-1} \approx 0.7358
      \]

Efficiently Computing Poisson

- Let \( X \sim \text{Poi}(\lambda) \)
  - Want to compute \( P(X = i) \) for multiple values of \( i \)
    - E.g., Computing \( P(X \leq a) = \sum_{i=0}^{a} P(X = i) \)
  - Iterative formulation:
    - Compute \( P(X = i + 1) \) from \( P(X = i) \)
      \[
P(X = i + 1) = \frac{e^{-\lambda} \frac{\lambda^{i+1}}{(i+1)!}}{e^{-\lambda}} = \frac{\lambda}{i + 1} P(X = i)
      \]
    - Use recurrence relation:
      \[
P(X = 0) = e^{-\lambda} \frac{\lambda^0}{0!} = e^{-\lambda}
      \]
      \[
P(X = i + 1) = \frac{\lambda}{i + 1} P(X = i)
      \]
Approximately Poisson Approximation

- Poisson can still provide good approximation even when assumptions “mildly” violated
- “Poisson Paradigm”
- Can apply Poisson approximation when...
  - “Successes” in trials are not entirely independent
    - Example: # entries in each bucket in large hash table
  - Probability of “Success” in each trial varies (slightly)
    - Small relative change in a very small $p$
    - Example: average # requests to web server/sec. may fluctuate

Birthday Problem Redux

- What is the probability that of $n$ people, none share the same birthday (regardless of year)?
  - $n = \binom{365}{2}$ trials, one for each pair of people $(x,y)$, $x \neq y$
  - Let $E_{x,y} = x$ and $y$ have same birthday (trial success)
  - $P(E_{x,y}) = p = 1/365$ (note: all $E_{x,y}$ not independent)
  - $X \sim \text{Poi}(\lambda)$ where $\lambda = \binom{n}{2} \frac{1}{365} n(n-1) \frac{1}{730}$
  - $P(X = 0) = e^{-n(n-1)/730} \left(\frac{n(n-1)}{730}\right)^0 = e^{-n(n-1)/730}$
  - Solve for smallest integer $n$, s.t.: $e^{-n(n-1)/730} \leq 0.5$
    - $\ln(e^{-n(n-1)/730}) \leq \ln(0.5) \rightarrow n(n-1) \geq 730 \ln(0.5) \rightarrow n \geq 23$
    - Same as before!

Poisson Processes

- Consider “rare” events that occur over time
  - Earthquakes, radioactive decay, hits to web server, etc.
  - Have time interval for events (1 year, 1 sec, whatever...)
  - Events arrive at rate: $\lambda$ events per interval of time
- Split time interval into $n \rightarrow \infty$ sub-intervals
  - Assume at most one event per sub-interval
  - Event occurrences in sub-intervals are independent
  - With many sub-intervals, probability of event occurring in any given sub-interval is small
  - $N(t) = \#$ events in original time interval $\sim \text{Poi}(\lambda)$

Web Server Load

- Consider requests to a web server in 1 second
  - In past, server load averages 2 hits/second
  - What is $P(X = 5)$?
- Model
  - Assume server cannot acknowledge > 1 hit/msec.
    - 1 sec = 1000 msec. (= large $n$)
  - $P($hit server in 1 msec$) = 2/1000 (= small $p$)
  - $X \sim \text{Poi}(\lambda = 2)$
    - $P(X = 5) = e^{-5} \frac{5^5}{5!} \approx 0.0361$

Geometric Random Variable

- $X$ is Geometric Random Variable: $X \sim \text{Geo}(p)$
  - $X$ is number of independent trials until first success
  - $p$ is probability of success on each trial
  - $X$ takes on values 1, 2, 3, ..., with probability:
    - $P(X = n) = (1 - p)^{n-1} p$
  - $E[X] = 1/p$ \hspace{1cm} $\text{Var}(X) = (1 - p)/p^2$
- Examples:
  - Flipping a fair ($p = 0.5$) coin until first “heads” appears.
  - Urn with N black and M white balls. Draw balls (with replacement, $p = N/(N + M)$) until draw first black ball.
  - Generate bits with $P($bit $= 1) = p$ until first 1 generated

Negative Binomial Random Variable

- $X$ is Negative Binomial RV: $X \sim \text{NegBin}(r, p)$
  - $X$ is number of independent trials until $r$ successes
  - $p$ is probability of success on each trial
  - $X$ takes on values $r, r + 1, r + 2, ...$, with probability:
    - $P(X = n) = \binom{n-1}{r-1} p^r (1 - p)^{n-r}$, where $n = r, r+1, ...$
  - $E[X] = np$ \hspace{1cm} $\text{Var}(X) = (1-p)/p^2$
- Note: $\text{Geo}(p) \sim \text{NegBin}(1, p)$
- Examples:
  - # of coin flips until $r$th “heads” appears
  - # of strings to hash into table until bucket 1 has $r$ entries
Hypergeometric Random Variable

- $X$ is **Hypergeometric** RV: $X \sim \text{HypG}(n, N, m)$
  - Urn with $N$ balls: $(N - m)$ black and $m$ white.
  - Draw $n$ balls **without** replacement
  - $X$ is number of white balls drawn
  
  $$P(X = i) = \binom{m}{i} \binom{N-m}{n-i} / \binom{N}{n}, \text{ where } i = 0, 1, ..., n$$

- $E[X] = n(m/N)$
- $\text{Var}(X) = \frac{nm(N - n)(N - m)}{N(N - 1)}$
  - Let $p = m/N$ (probability of drawing white on 1st draw)
  - As $n \to \infty$ and $m/N$ remains constant

Endangered Species

- Determine $N = \text{how many of some species remain}$
  - Randomly tag $m$ of species (e.g., with white paint)
  - Allow animals to mix randomly (assuming no breeding)
  - Later randomly observe another $n$ of the species
  - $X = \text{number of tagged animals in observed group of } n$
  - $X \sim \text{HypG}(n, N, m)$

- “Maximum Likelihood” estimate
  - Set $\hat{N}$ to be value that maximizes:
    
    $$\frac{m}{\binom{n}{i} \binom{N-m}{n-i} / \binom{N}{n}}$$

    for the value $i$ of $X$ that you observed $\rightarrow \hat{N} = \frac{mn}{i}$

- Similar to assuming: $i = E[X] = nm/N$