Overview

- Bernoulli Random Variables
- Binomial Random Variables
- Simulation

Jacob Bernoulli

Jacob Bernoulli (1655-1705) was a Swiss mathematician.

- Discovered e by looking at compound interest
- Was interested in how much one would expect to win in games of chance (only two outcomes, win or lose)

Bernoulli Random Variable

Suppose an experiment results in Success or Failure.

- X is a random indicator variable, $X = 1$ on success, $X = 0$ on failure
- $P(X = 1) = p$
- $P(X = 0) = 1 - p$
- X is a **Bernoulli** random variable.
- Sometimes write $X \sim Ber(p)$.

Examples:

- Coin flip
- Random binary digit
- Packet erasure in a wireless network
- Transmission by WiFi station

Binomial Random Variable

Consider n independent trials of a $Ber(p)$ random variable

- X is the number of successes in n trials
- X is the sum of n Bernoulli random variables, $X = X_1 + X_2 + \cdots + X_n$, where random variable $X_i \sim Ber(p)$ is 1 if success in trial i and 0 otherwise.
- X is a **Binomial** random variable: $X \sim Bin(n, p)$

$$P(X = i) = \binom{n}{i} p^i (1-p)^{n-i}, \; i = 0, 1, \cdots, n$$

(recall $\binom{n}{i}$ is the number of outcomes with exactly i successes and $n - i$ failures)

Examples:

- number of heads in n coin flips
- number of 1's in randomly generated bit string of length n
- number of packets erased out of a file of n packets

Binomial Random Variable

Binomial variable $X \sim Bin(n, p)$ is sum of n Bernoulli random variables $X_i \sim Ber(p)$, $i = 1, 2, \cdots, n$

- Suppose $X \sim Bin(n_1, p)$ and $Y \sim Bin(n_2, p)$ (its important than p is the same for both)
- Then $Z = X + Y \sim Bin(n_1 + n_2, p)$
- $Z = X_1 + X_2 + \cdots + X_{n_1} + Y_1 + Y_2 + \cdots + Y_{n_2}$. All terms are independent, all are $Ber(p)$.

Example: Error Correcting Codes

- Have information 4 bits to send across network
- Add 3 "parity" bits, send 7 bits total
- Each bit is idependently corrupted (flipped) in transition with probability 0.1
- $X =$ number of bits corrupted, $X \sim Bin(7, 0.1)$
- Parity bits allows us to correct at most 1 error
- Probability that a correctable message is received is $P(X < 2)$.

Example: Error Correcting Codes

- $P(X < 2) = P(X = 0) + P(X = 1)$.

$$P(X = 0) = \binom{7}{0} 0.1^0 0.9^7 \approx 0.48$$

$$P(X = 1) = \binom{7}{1} 0.1^1 0.9^6 \approx 0.37$$

$$P(X = 0) + P(X = 1) \approx 0.85$$

- Not using error correcting code, $X \sim Bin(4, 0.1)$

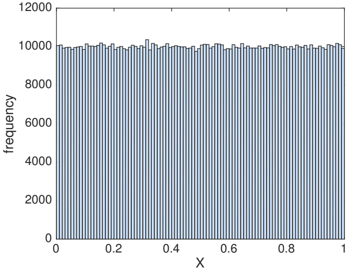
$$P(X = 0) = \binom{4}{0} 0.1^0 0.9^4 \approx 0.66$$

- So error correcting code improves reliability by about 30%

Stochastic Simulation

On a computer how can we obtain realisations of a Bernoulli random variable ?

- Assume we have a random number generator that picks an integer between 0 and maxint uniformly a random. We then divide by maxint to give value U between 0 and 1
- Always worth checking its not too bad though e.g. distribution of 1M samples generated by matlab rand() function[1]:



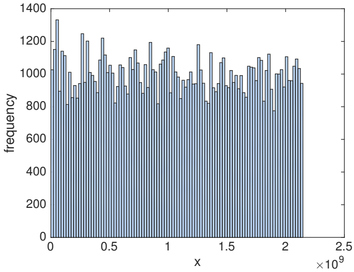
[1] r=rand(1,1000000); hist(r,100)

Digression: Pseudo-random number generators

- Traditional approach is to use something like:

$$X_{n+1} = (aX_n + c) \mod m$$

with $a = 1103515245$, $c = 12345$, $m = 2^{31}$ (used by glibc). E.g.[2]



- But known to be not so great. Much better (and more complicated) is the Mersenne Twister – used by Python, Matlab etc.
- Crypto uses specialised PRNGs. Dual_EC_DRBG likely has been weakened by the NSA

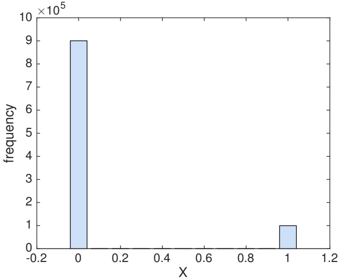
[2] x=100; y=[]; for i=1:100000,x=mod(1103515245*x+12345,2^31); y=[y;x]; end; hist(y,100)

Stochastic Simulation: Bernoulli Random Variable

- Generate values with a $Ber(p)$ distribution:

$$X = \begin{cases} 1 & U \leq p \\ 0 & U > p \end{cases}$$

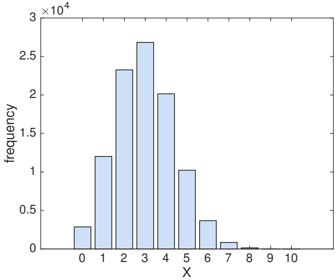
- Using matlab[3] with $p = 0.1$:



[3] r=rand(1,1000000); x=(r<=0.1); hist(x,[0:0.1:1])

Stochastic Simulation: Binomial Random Variable

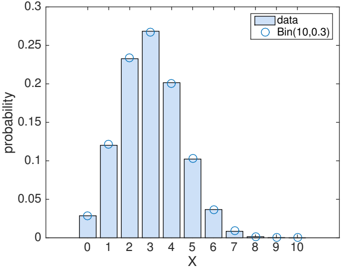
- Generate values with a $Bin(n, p)$ distribution.
- First generate n values from a $Ber(p)$ distribution, then sum them.
- Using matlab[4] with $n = 10$ and $p = 0.3$:



[4] y=[];p=0.3;n=10;for i=1:100000,r=rand(1,n); x=(r<=p); y=[y;sum(x)]; end; nn=hist(y,[0:10]); bar([0:10],nn)

Stochastic Simulation: Binomial Random Variable

- These plots are of frequencies. How can we convert to probabilities ?
- Normalise so that they sum to 1
- We can then plot the distribution of our numerical samples against the true distribution as a check.
- Using matlab[5] with $n = 10$ and $p = 0.3$:



[5] y=[];p=0.3;n=10;for i=1:100000,r=rand(1,n); x=(r<=p); y=[y;sum(x)]; end; x=[0:10];nn=hist(y,x);for i=x; ch(i+1)=nchoosek(n,i);end;bar(x,nn./sum(nn));hold on;plot(x,ch.*p.x̂.*(1-p).(̂10-x),'o')