Homework 1

Kevin Chao

October 10, 2019

Question 1:

a. Let's make **X** the number of disk failures in one year. As it checks **n** with the probability **p**, the number of disk failures **X** has a binomial distribution (X∼Bin(n,p)). The expected value of this binomial is:
E[X] = np = 3p
It is reliable when **x<k** (since p is the probabilty of individual disks *failing* in a one year period, you only want to have 0 or 1 disk failures to have 2 disks succeed) where **k=2**, then the probability of the whole array to continue w/o any data loss is

$$\begin{aligned} P(x<2) &= P(x=0) + P(x=1) \\ &= \binom{3}{0}(1-p)^3 + \binom{3}{1}p(1-p)^{3-1} \\ &= (1-p)^3 + 3p(1-p)^2 \end{aligned}$$

b. Now **n=5** and **k=3**, which means that the expected value is now:
E[X] = 5p
and the system's reliability requires that less than 3 disk failures occurs (**x<k**), so the probabilty of reliability is:

$$\begin{aligned} P(x<3) &= P(x=0) + P(x=1) + P(x=2) \\ &= \binom{5}{0}(1-p)^5 + \binom{5}{1}p(1-p)^{5-1} + \binom{5}{2}p^2(1-p)^{5-2} \\ &= (1-p)^5 + 5p(1-p)^4 + 10p^2(1-p)^3 \end{aligned}$$

c. When p=0.05, then the probability of reliability in (a) is

$$\begin{aligned} &= (.95)^3 + 3(.05)(.95)^2 \\ &= .99275 \end{aligned}$$

and the probability of reliability in (b) is

$$\begin{aligned} &= (.95)^5 + 5(.05)(.95)^4 + 10(.05)^2(.95)^3 \\ &= .99884 \end{aligned}$$

so (b) is more reliable than (a)

d. When p=0.65, then the probability of reliability in (a) is

$$
\begin{aligned}
&= (.35)^3 + 3(.65)(.35)^2 \\
&= .28175
\end{aligned}
$$

The probability of reliability in (b) is

$$
\begin{aligned}
&= (.35)^5 + 5(.65)(.35)^4 + 10(.65)^2(.35)^3 \\
&= .23517
\end{aligned}
$$

so (a) is more reliuable than (b)

Question 2:

a. Since there are six digits for a passcode ranging from 0-9, the total number of passcodes would be: 10^6. If m is the number of users on the social media, let (m-1) be the users that are not the self. Since a safe passcode is not used in someone elses account, subtract 1 from 10^6 as it is no longer a possible passcode for other users. The probability of having a safe password is then

$$
\begin{aligned}
P(Safe) &= \frac{(10^6 - 1)^{m-1}}{10^{6(m-1)}} \\
&= .99999^{m-1}
\end{aligned}
$$

b. The probability of having a not safe passcode is
$1 - 0.99999^{m-1}$
Now it needs to be 50% , so the probability of not being safe being 50% or over is

$$
\begin{aligned}
1 - 0.99999^{m-1} &\geq 0.5 \\
0.99999^{m-1} &\leq 0.5 \\
m - 1 &\geq \frac{\ln 0.5}{\ln 0.99999} \\
m - 1 &\geq 693146.83 \\
m &\geq 693147.83
\end{aligned}
$$

Thus, it would take 693148 users to have a probability of 50% or higher of an unsafe passcode

c. Since the order in which the digits are assigned matters, the probability of all users having a safe passcode is the number of permutations of 10^6 passcodes from m at a time divided by the total number of ways to assign each person in 10^6 ways:

$$
\frac{\frac{10^6!}{(10^6 - m)!}}{10^{6m}}
$$

d. Using the formula for (c), the probability that atleast one user's passcode is not safe is:

$1 - \dfrac{\dfrac{10^6!}{(10^6 - m)!}}{10^{6m}}$

Just like in (b), now change it as an inequality to 50% :

$$1 - \frac{\dfrac{10^6!}{(10^6 - m)!}}{10^{6m}} \geq 0.5$$
$$\frac{\dfrac{10^6!}{(10^6 - m)!}}{10^{6m}} \leq 0.5$$

From here, I used python to dwindle down the exact answer. To do that, I simplified the inequality to

$$\frac{10^6!}{(10^6 - m)!} \leq \frac{10^{6m}}{2}$$

I kept changing the first for loop's start, stop, and increment parameter in the range function (0-999999, inc 100000, then to 399999-499999, inc 10000, etc.) . The code is the following:

```
for i in range(494300,494310, 1):
  x = 1
  y = (1000000**i)
  y = y//2
  for i2 in range(1000000,i, -1):
    x = x * i2
  if x <= y:
    print(i)
```

The amount of users for there to be greater than 50% is **494304**

Question 3:

a. A pair of nodes can be selected from a set of n nodes in ${}_nC_2$

$$\begin{aligned} {}_nC_2 &= \frac{n!}{2!(n-2)!} \\ &= \frac{n(n-1)}{2} \end{aligned}$$

And since the probability of of a pair of people has a probability of 1/2 and there is 1 edge, the expected value would be ${}_nC_2$ * Probability of link

$$\begin{aligned} E[X] &=_n C_2 * \frac{1}{2} \\ &= \frac{n(n-1)}{4} \end{aligned}$$

For n=10, the expected number of friend relationships would be:
$$\frac{10*9}{4} = 22.5$$

b. All three pairs are linked by an edge, so now it can be selected from a set of n nodes in ${}_nC_3$

$$\begin{aligned} {}_nC_3 &= \frac{n!}{3!(n-3)!} \\ &= \frac{n(n-1)(n-2)}{6} \end{aligned}$$

A triplet of nodes means that there are 3 edges, which means that 1/2 * 3 = 1/ 8. And for n=10, the expected number of 3-cliques would be
$$\frac{(10)(9)(8)}{6*8} = 15$$

c. Looking at (b) since it fits the criteria of $2 \leq k \leq n$, it can be established that k nodes can be selected in ${}_nC_k$ ways. Since out of k nodes, it chooses two people to connect and create an edge, the probability that k nodes can connect to each other is $(\frac{1}{2})^{{}_kC_2}$
So, the expected number of cliques of size k is:
${}_nC_k * (\frac{1}{2})^{{}_kC_2}$
When k=4 and n=10, then the expected number of 4-cliques is:

$$\begin{aligned} {}_nC_4 &= \frac{n!}{4!(n-4)!} && (1) \\ &= \frac{n(n-1)(n-2)(n-3)}{24} && (2) \\ (\frac{1}{2})^{{}_kC_2} &= (\frac{1}{2})^6 && (3) \\ &= \frac{1}{64} && (4) \\ {}_nC_4 * \frac{1}{64} &= \frac{(10)(9)(8)(7)}{64*24} && (5) \\ E[X] &= 3.28125 && (6) \end{aligned}$$

Question 4:

a. Under the empirical distribution, the variance of random variable S and T are:
$$Var[S] = \frac{1}{n}\sum_{i=1}^{n} s_i^2 - (\frac{1}{n}\sum_{i=1}^{n} s_i)^2$$
$$Var[T] = \frac{1}{n}\sum_{i=1}^{n} t_i^2 - (\frac{1}{n}\sum_{i=1}^{n} t_i)^2$$
The code to computer these values are as followed:

```
S = np.load('eruptions.npy')  # vector of observed eruption times
T = np.load('waiting.npy')    # vector of observed waiting times
n = S.shape[0]                # number of observations
squaredS = 0
for item in S:
  squaredS = squaredS + (item * item)
squaredS = squaredS/n
squaredT = 0
for item in T:
  squaredT = squaredT + (item * item)
squaredT = squaredT/n
meanS = np.sum(S)/n * (np.sum(S)/n)
meanT = (np.sum(T)/n) * (np.sum(T)/n)
varS = squaredS - meanS
varT = squaredT - meanT
```

The value for var[S] = **1.298** and the value for var[T] = **184.144**

b. Since the proibability is under the empirical distribution, each eruption and waiting time has varying probability from 0.0-1.0. The code to find these times are as followed:

```
S = np.load('eruptions.npy')  # vector of observed eruption times
T = np.load('waiting.npy')    # vector of observed waiting times
n = S.shape[0]                # number of observations
eruptionX = np.sort(S)
waitingX = np.sort(T)
timesY = np.arange(1, n+1)/n
eruptionTimes = []
waitingTimes = []
for i in range(len(y)):
  if (y[i]*4).is_integer():
    eruptionTimes.append((x[i],y[i]))
    waitingTimes.append((x[i],y[i]))
```

The eruption times are: $[\bar{s}_1 = 2.15, \bar{s}_2 = 4.0, \bar{s}_3 = 4.45]$
The waiting times are: $[\bar{t}_1 = 58.0, \bar{t}_2 = 76.0, \bar{t}_3 = 82.0]$

c. For joint probability mass function, the table below was made by python, observing the frequency of the four patterns.

Table 1: Joint Probability Mass Function

	X=0	X=1
Y=0	100/272	3/272
Y=1	4/272	165/272

From this table, we can now get the Marginal Probability Mass Function.

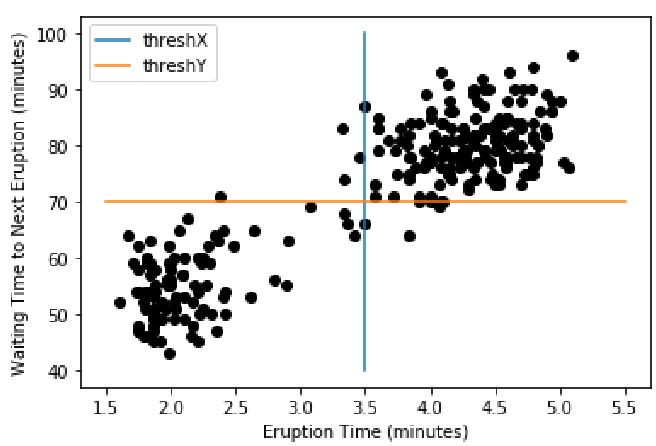
$$p_X(x) = \begin{cases} 104/272 & x = 0 \\ 168/272 & x = 1 \end{cases}$$

$$p_Y(y) = \begin{cases} 103/272 & y = 0 \\ 169/272 & y = 1 \end{cases}$$

Below is the code to determine how many of the patterns there were:

```
x1y1 = 0
x0y0 = 0
x1y0 = 0
x0y1 = 0
for i in range(0,n):
  if S[i] >= threshX:
    if T[i] >= threshY:
      x1y1 += 1
    else:
      x1y0 += 1
  else:
    if T[i] >= threshY:
      x0y1 += 1
    else:
      x0y0 += 1
```

d. The random variable X and Y are dependent. The amount of dependence that X and Y have on each other are strong. Looking at the Joint Probability Mass Function, the probability that Y=X is extremely high (Y=X=0 or Y=X=1), with a 265/272 probability. And on the other spectrum, its extremely low how often they will be different, with a 7/272 probability.



In addition to the plot graph of S and T, I added lines to represent the thresholds of X and Y, effectively creating a quadrant. The strength of the dependency is shown in this graph such that it keeps it mostly consistent that when x (eruption time) gets bigger, so too does y (waiting time).