Solutions to Problem Set 3

1. (MU 3.3) Suppose that we roll a standard fair die 100 times. Let X be the sum of the numbers that appear over the 100 rolls. Use Chebyshev’s inequality to bound $P(|X - 350| \geq 50]$.

Let $X_i$ be the number on the face of the die for roll $i$. Let $X$ be the sum of the dice rolls. Therefore $X = \sum_{i=1}^{100} X_i$. By linearity of expectation, we write $\mathbb{E}[X] = \sum_{i=1}^{100} \mathbb{E}[X_i]$. We can compute

$$\mathbb{E}[X_i] = \sum_{j=1}^{6} j \mathbb{P}[X_i = j] = \sum_{j=1}^{6} j (1/6) = (1/6) \frac{6(7)}{2} = 7/2,$$

where we use the fact that $\sum_{j=1}^{n} j = \frac{n(n+1)}{2}$. Then we have

$$\mathbb{E}[X] = 100(7/2) = 350.$$

To use Chebyshev’s inequality, the only remaining value we need to compute is the variance of $X$. By the independence of the dice rolls we have

$$\text{Var}(X) = \text{Var} \left( \sum_{i} X_i \right) = \sum_{i=1}^{100} \text{Var}(X_i)$$

To compute the variance of a single dice roll, we use $\text{Var}(X_i) = \mathbb{E}[X_i^2] - \mathbb{E}[X]^2$

$$\mathbb{E}[X_i^2] = \sum_{j=1}^{6} j^2 \mathbb{P}[X_i = j]$$

$$= \sum_{j=1}^{6} j^2 (1/6)$$

$$= \frac{1}{6} \cdot \frac{6(7)(13)}{6}$$

$$= 91/6$$

where we use the fact that $\sum_{j=1}^{n} j^2 = \frac{n(n+1)(2n+1)}{6}$. Now we can finish computing the variance of $X_i$ as

$$\text{Var}(X_i) = \mathbb{E}[X_i^2] - \mathbb{E}[X]^2 = 91/6 - (7/2)^2 = 35/12.$$

And the variance of $X$ is $\text{Var}(X) = 100(231/4)$. Finally, we can by Chebyshev’s inequality we have

$$P[|X - 350| \geq 50] \leq \frac{100(35/12)}{50^2} = 7/60.$$

2. (MU 3.5) Given any two random variables $X$ and $Y$, by the linearity of expectations we have $\mathbb{E}[X - Y] = \mathbb{E}[X] - \mathbb{E}[Y]$. Prove that, when $X$ and $Y$ are independent, $\text{Var}[X - Y] = \text{Var}[X] + \text{Var}[Y]$. 
From the definition of variance, we write
\[
\text{Var}[X - Y] = \mathbb{E}[(X - Y)^2] - \mathbb{E}[X - Y]^2
\]
\[
= \mathbb{E}[X^2 - 2XY + Y^2] - (\mathbb{E}[X] - \mathbb{E}[Y])^2
\]
\[
= \mathbb{E}[X^2] - 2\mathbb{E}[XY] + \mathbb{E}[Y^2] - (\mathbb{E}[X]^2 - 2\mathbb{E}[X]\mathbb{E}[Y] + \mathbb{E}[Y]^2)
\]
\[
= \mathbb{E}[X^2] - \mathbb{E}[X]^2 + \mathbb{E}[Y^2] - \mathbb{E}[Y]^2,
\]
since by independence \( \mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y] \). Finally, we see that
\[
\text{Var}[X - Y] = \text{Var}[X] + \text{Var}[Y].
\]

3. (MU 3.6) For a coin that comes up heads independently with probability \( p \) on each flip, what is the variance in the number of flips until the \( k \)th head appears?

The number of coin flips until a head is a geometric random variable, \( X_i \), with parameter \( p \). Let \( X \) be the number of coin flips until \( k \) heads. Then \( X = \sum_{i=1}^{k} X_i \).

\[
\text{Var}[X] = \text{Var}\left[ \sum_{i=1}^{k} X_i \right]
\]
\[
= \sum_{i=1}^{k} \text{Var}[X_i]
\]
\[
= \sum_{i=1}^{k} \frac{(1-p)/p^2}
\]
\[
= k(1-p)/p^2
\]

4. (MU 3.19) Let \( Y \) be a non-negative integer-valued random variable with positive expectation. Prove
\[
\frac{\mathbb{E}[Y]^2}{\mathbb{E}[Y^2]} \leq \mathbb{P}[Y \neq 0] \leq \mathbb{E}[Y].
\]

First, we consider the upper bound. By Markov’s inequality, we have\[
\mathbb{P}[Y \neq 0] = \mathbb{P}[Y \geq 1] \leq \mathbb{E}[Y].
\]

Now, for the lower bound. Notice that one might think of using \( \mathbb{P}[Y \neq 0] = 1 - \mathbb{P}[Y = 0] \), and upper bounding \( \mathbb{P}[Y = 0] \) by Chebyshev’s inequality. However, this will not work, because Jensen’s inequality, for converting \( \mathbb{E}[X]^2 \) to \( \mathbb{E}[X^2] \), cannot provide a bound in the proper direction.
(Note that the first solution provided to this problem was incorrect in attempting to use proceed via Chebyshev’s and Jensen’s inequalities.)

A much more useful approach is to use conditional expectation to obtain an inequality that contains \( \mathbb{P}[Y \neq 0] \) as one of the coefficients. Recall that Jensen’s inequality tells us that
\[
\mathbb{E}[X]^2 \leq \mathbb{E}[X^2]
\]
Let $X$ be a random variable derived from $Y$ where $X$ has sample space $\Omega' = \{ \Omega \setminus \{ z \in \Omega | Y(z) = 0 \}\}$ where $\Omega$ is the sample space for $Y$, and where the random variable $X$ satisfies $X(z) = Y(z) \ \forall z \in \Omega'$. So, we have defined $X$ such that $X = (Y|Y \neq 0)$. Then, the above Jensen’s inequality tell us that

$$E[Y|Y \neq 0]^2 \leq E[Y^2|Y \neq 0].$$

Now we compute each side of the above inequality. For the left-hand side we have

$$E[Y|Y \neq 0] = \left( \sum_{i=0}^{\infty} i P[Y = i|Y \neq 0] \right)^2$$

$$= \left( \sum_{i=0}^{\infty} \frac{i P[Y = i, Y \neq 0]}{P[Y \neq 0]} \right)^2$$

$$= \left( \sum_{i=1}^{\infty} \frac{i P[Y = i]}{P[Y \neq 0]} \right)^2$$

$$= \frac{E[Y]^2}{P[Y \neq 0]^2}.$$

For the right-hand side, we have

$$E[Y^2|Y \neq 0] = \sum_{i=0}^{\infty} i^2 P[Y = i|Y \neq 0]$$

$$= \sum_{i=1}^{\infty} i^2 P[Y = i] \frac{P[Y \neq 0]}{P[Y \neq 0]}$$

$$= \frac{E[Y^2]}{P[Y \neq 0]}.$$

Putting everything together, we have

$$\frac{E[Y]^2}{P[Y \neq 0]^2} \leq \frac{E[Y^2]}{P[Y \neq 0]}$$

$$\frac{E[Y]^2}{E[Y^2]} \leq P[Y \neq 0],$$

which concludes the proof.

5. (MU 3.20)

(a) Chebyshev’s inequality uses the variance of a random variable to bound its deviation from its expectation. We can also use higher moments. Suppose that we have a random variable $X$ and an even integer $k$ for which $E[(X - E[X])^k]$ is finite. Show that

$$P\left[|X - E[X]| > t \sqrt{E[(X - E[X])^k]}\right] \leq \frac{1}{t^k}.$$
Let \( Y = (X - \mathbb{E}[X])^k \). By Markov’s inequality we have \( \mathbb{P}[Y \geq t^k \mathbb{E}[Y]] \leq \frac{\mathbb{E}[Y]}{t^k \mathbb{E}[Y]} = \frac{1}{t^k} \). Now, we have
\[
\mathbb{P} \left[ Y \geq t^k \mathbb{E}[Y] \right] = \mathbb{P} \left[ \sqrt[k]{Y} \geq t \sqrt[k]{\mathbb{E}[Y]} \right] = \mathbb{P} \left[ |X - \mathbb{E}[X]| \geq t \sqrt[k]{\mathbb{E}[|X - \mathbb{E}[X]|^k]} \right]
\]
where the first step is true since we take the \( k \)th root of both sides of the inequality, and the second step is true since the \( k \)th root of a number, where \( k \) is even, is the absolute value. Putting this together with the Markov’s inequality, we have
\[
\mathbb{P} \left[ |X - \mathbb{E}[X]| \geq t \sqrt[k]{\mathbb{E}[|X - \mathbb{E}[X]|^k]} \right] \leq \frac{1}{t^k}.
\]

(b) Why is it difficult to derive a similar inequality when \( k \) is odd? Since \( X \) is any random variable, the value \( (X - \mathbb{E}[X])^k \) may be negative for odd values \( k \). Therefore Markov’s inequality would not apply.

6. **(MU 3.21)** A fixed point of a permutation \( \pi : [1, n] \to [1, n] \) is a value for which \( \pi(x) = x \). Find the variance in the number of fixed points of a permutation chosen uniformly at random from all permutations.

Let \( X_i \) be an indicator random variable for the event that \( \pi(i) = i \), making \( i \) a fixed point, i.e. \( X_i = 1 \) when \( i \) is a fixed point, and \( X_i = 0 \) otherwise. We can easily compute the \( \mathbb{E}[X_i] \). Let \( X = \sum_{i=1}^{n} X_i \) be the number of fixed points.

First, we notice that \( \text{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2 \). Next, we compute the expectation of the number of fixed points. Since the \( \mathbb{E}[X_i] = \mathbb{P}[X_i] = 1/n \), we have
\[
\mathbb{E}[X] = \mathbb{E} \left[ \sum_{i=1}^{n} X_i \right] = \sum_{i=1}^{n} \mathbb{E}[X_i] = \sum_{i=1}^{n} \left( \frac{1}{n} \right) = 1.
\]

Now, we compute the first term in the variance,
\[
\mathbb{E}[X^2] = \mathbb{E} \left[ \left( \sum_{i=1}^{n} X_i \right)^2 \right]
\]
\[
= \left( \sum_{i=1}^{n} \mathbb{E}[X_i^2] \right) + \left( \sum_{i=1}^{n} \sum_{j \neq i} \mathbb{E}[X_i X_j] \right)
\]
\[
= \left( \sum_{i=1}^{n} \mathbb{E}[X_i] \right) + \left( \sum_{i=1}^{n} \sum_{j \neq i} \mathbb{E}[X_i X_j] \right)
\]
\[
= 1 + \left( \sum_{i=1}^{n} \sum_{j \neq i} \mathbb{P}[X_i = 1|X_i X_j]|X_i = 1] \right)
\]
\[
= 1 + \left( \sum_{i=1}^{n} \sum_{j \neq i} \frac{1}{n(n-1)} \right)
\]
\[
= 1 + 1
\]
\[
= 2
\]
The third line follows since for indicator variables $X_i^2 = X_i$. The forth line is obtained by using conditional expectation, conditioning on the event $X_i = 1$. The fifth line comes from knowing that $\mathbb{P}[X_i = 1] = 1/n$, and conditioning on $X_i = 1$, there are $n - 1$ choices for mapping element $j$, yielding $1/(n - 1)$ as the conditional probability of $j$ being a fixed point.

Putting everything together we have

$$\text{Var}[X] = 2 - 1 = 1.$$ 

7. **(Balls and Bins)** This problem involves a balls and bins experiment in which $m$ balls are tossed independently into $n$ bins with each ball equally likely to land in any bin.

(a) Using a Chernoff bound, estimate the probability that if $3n \ln n$ balls are tossed into $n$ bins, the maximum number of balls in any bin is less than or equal to $2 \ln n$.

This problem was not graded due to issues with the problem statement.

(b) Write a simple program that simulates the balls and bins experiment for given values of $m$ and $n$ up to one million. You will need a random number generator; the standard C library function `drand48()` is recommended. Consult the man page for details.

The most straight forward way to do the simulation is to use an array of dimension one million to record the number of balls that have landed in each bin. It is wasteful of space to use one million integers to store the bin loads. Since you are very unlikely ever to see a load greater than 15, you can in fact use a single byte (i.e., a character) to store each bin load.

There is an alternate way to do the simulation using an integer array which records, in position $i$, the number of bins that have received exactly $i$ balls. This eliminates the need for an array of size one million. If you see the trick, explain how to do the simulation this way. Using one of the above two methods, perform at least 20 (preferably 100) simulations with $m = n = 10^6$, and make a table of the distribution of the maximum loads.

Now, consider the following alternative scheme: balls are again thrown sequentially, but instead of simply choosing a single bin at random, each ball now chooses two bins at random, inspects their current loads, and goes to the less full of the two (breaking ties arbitrarily). Modify your program to implement this scheme. Again perform at least 20 simulations with $m = n = 10^6$, and make a table of the maximum loads that you observe. Are you surprised by the results?

Finally, would you expect an even more dramatic effect if the balls were allowed three choices rather than two? Modify your program again and see what happens.

For one choice the distribution is

<table>
<thead>
<tr>
<th>Max Load</th>
<th>$\mathbb{P}[\text{Max Load}]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0.400000</td>
</tr>
<tr>
<td>9</td>
<td>0.540000</td>
</tr>
<tr>
<td>10</td>
<td>0.050000</td>
</tr>
<tr>
<td>11</td>
<td>0.010000</td>
</tr>
</tbody>
</table>

For two choices the distribution is

<table>
<thead>
<tr>
<th>Max Load</th>
<th>$\mathbb{P}[\text{Max Load}]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1.0</td>
</tr>
</tbody>
</table>

For three choices the distribution is

<table>
<thead>
<tr>
<th>Max Load</th>
<th>$\mathbb{P}[\text{Max Load}]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1.0</td>
</tr>
</tbody>
</table>
```cpp
#include <vector>
#include <time.h>
#include <math.h>

using namespace std;

int main (int argc, char **argv)
{
    if (argc != 2) {
        printf("USAGE: %s <number of choices>\n", argv[0]);
        exit(-1);
    }

    int choices = atoi(argv[1]);

    long int seed = time(NULL);
    srand48(seed);

    int num_balls = (int) pow(10,6);
    int num_bins = num_balls;
    int num_simulations = 100;
    printf("Choices %i, Simulations %i, No. Balls %i, No. Bins %i\n\n", choices, num_simulations, num_balls, num_bins);

    vector<int> distribution;
    for (int i = 0; i < num_simulations; i++)
    {
        vector<int> bins(num_bins, 0);

        // throw the balls in the bins
        for (int j = 0; j < num_balls; j++)
        {
            int bin = -1;
            for (int c = 0; c < choices; c++)
            {
                double r = drand48()*100000000.0;
                // rest of the code...
            }
        }
    }
}
```
int b = ((int) r) % num_bins;
if (bin == -1) {
    bin = b;
} else {
    if (bins[b] < bins[bin])
        bin = b;
}
    
    bins[bin]++;

// find the max load
int max_load = 0;
for (int j = 0; j < bins.size(); j++) {
    if (bins[j] > max_load)
        max_load = bins[j];
}
while (distribution.size() <= max_load)
    distribution.push_back(0);
    distribution[max_load]++;

// print out the distribution
printf("Distribution of maximum load:\n");
for (int x = 0; x < distribution.size(); x++) {
    double probability_x = distribution[x]/(double) num_simulations;
    printf("[%i] %f\n", x, probability_x);
}