In this project, we’re going to do three things: 1) extend the work we did with the fundamental counting principle (FCP) from your last project, 2) show you how, when all else fails, computer simulations can oftentimes save the day when dealing with probability, and 3) discover a wonderful mathematical construct.


Years ago, in the era before cell phone ubiquity, a friend of mine had a little pocket electronic organizer that kept track of his phone numbers. It was protected, as many digital items are, by a password. He, however, had forgotten his, and, knowing that I was a math geek, wanted to know if I had some kind of shortcut to help him rediscover it.

I asked him what he remembered about his password, and he told me, “It’s two digits long.” At this point, I simply by typing “00”, then tried “01”, then “02”, and so on. He freaked a little, saying, “What are you doing? I thought you were a math whiz?” Even I could do that…but I didn’t, because I knew it would take too much time!

In the time it had taken him to say that and swear a few times, I had cracked the password (if I remember correctly, it was 23). As he stood there slack – jawed, I explained to him why I had simply “brute forced” it…but you all already know why, right?

You see, there are 10 digits (0, 1, 2, 3, 4, 5, 6, 7, 8 and 9). So, if the password were only one digit long, there would have been 10 possibilities. But his password was 2 digits long. According to the FCP, to find out how many 2-digit passwords exist, I multiply the possibilities for each of the digits:

\[
\text{number of 2-digit passwords} = \text{number of possible first digits} \times \text{number of possible second digits} = 10 \times 10 = 100
\]

Then, realizing that I could check a 2-digit number about once every second, I just started in. And, about 30 seconds later, I had it.

(Did you catch how that was “sampling with replacement”? That’s because I could use the same digit in both places. If I weren’t allowed to replicate digits, I’d have only had \(10 \times 9 = 90\) passwords)

Of course, things have changed. 2-digit passwords are about as retro as acid washed jeans. Nowadays, passwords are usually required to be very long. Why? Well, security, that’s why!

Years ago, when I started my first email address, all that was required was a 4-digit password. Since there are 10 digits (0 through 9), there were only \(10^4\), or 10,000 possible passwords, from 0000 to 9999, and everything in between.

Well, as computers got faster and faster, it became easier and easier to crack these passwords. For example, a computer that can generate 10 passwords each second (a very, very slow computer) will get to your password in a maximum of around 15 minutes.

Next, companies began requiring you to use both letters and digits in your passwords. Now, each element of your 4-character password has 36 possibilities, right? 10 digits or 26 letters. You now have \(36^4\), or about 1.7 million, passwords…orders of magnitude more!
Then, these companies began differentiating between lowercase and uppercase. Now, each character of your 4–character password has 62 possibilities (10 digits, or 26 lowercase letters or 26 uppercase letters), thus creating $62^4$, or 14.8 million options. Wowsers!

And then the requirements for passwords got longer and longer. For example, an 8–character password under the previous paragraph’s rules would have $62^8$, or over 218 trillion possibilities!

To further thwart hackers, passwords are often required to obey a rule like this: “Your password must have between 6 and 10 case specific characters, including digits.” Well, that sounds like we’ve created some disjoint sets there...our password could be 6 digits long, or 7, or 8, or 9, or 10. Remember that “or” means to add...therefore, there are $62^6 + 62^7 + 62^8 + 62^9 + 62^{10}$, or just over 853 quadrillion possible passwords.

To put it in perspective, a hacker with a good desktop computer with a reasonably fast processor can test about 10,000,000 passwords per second (this is sometimes called a “brute force” method, and assumes the hacker isn’t locked out after a certain number of failed trials). The previous requirement of password length would keep that computer busy for a maximum of about

$$\text{853 quadrillion passwords} \times \frac{1 \text{ second}}{10000000 \text{ passwords}} \times \frac{1 \text{ year}}{31556926 \text{ seconds}} \approx 2700 \text{ years}$$

Of course, that’s the maximum amount of time it would need. A supercomputer is about 100 times faster...so it would “only” take it 27 years.

Also, the abovementioned 2700 years assumes that the hacker has already found your login name, as well. If (s)he has to guess at both the login and the password, it would take him/her much, much longer.

Ready to try a couple of your own?

* * * * * * * *

At COCC, you each have a Banner account. Here, you can register for classes, check your transcript, pay parking fines...all kinds of good stuff.

1. (3 points) (w) Your COCC Banner password (PIN) was originally your birthday in MMDDYY form, but you were asked to change it immediately to a more secure one. Paraphrased from the Banner login page: “Your new PIN must be between 6 and 15 characters inclusive, using any combination of letters and numbers. If you use letters, note that the system is case sensitive.” So, for each of the characters, as above, there are 62 possibilities. About how many different possible Banner passwords are there, using this new rule? You can answer as I did above, using “illions”. Or you may use scientific notation, but be sure to write it properly...no “E”’s!

2. (2 points) (w) How long (maximum) would it take a hacker to crack your PIN using brute force, assuming his/her computer can test one billion PINS a second? Answer in years, and use either “illions” or scientific notation.

* * * * * * * *

Remember when I brute forced my buddy’s pocket organizer? I had a maximum time of about 2 minutes (100 seconds), but found it in about 30 seconds. On average, you would expect this computer to find it roughly halfway through its exhaustive list...but that would still be after about 1350 years (13.5 for the supercomputer). Both are too long to be useful to a hacker.
The previous examples referred to passwords. Do you agree that, if we moved the digits of a password around, we get a new password? For example, do you agree that 1234 is a different password than 2341?

Arrangements of things where order is important are called permutations. There are two kinds of permutations...with and without repetition. These two distinctions parallel independent and dependent sampling from your last project. In the previous examples, we looked at permutations with repetition, since it was OK to reuse characters in a password (i.e., ABB777 for a six – digit password).

Now, let’s consider a different kind of permutation, one where you aren’t allowed to repeat elements. For example, suppose you have the letters in the word “MATH”. How many different ways can you scramble them, using all 4 letters?

Well, you have 4 choices for the first letter position of the word. But then, you only have 3 letters left to choose for the second position, since you can’t reuse the letter you put in the first position. Moving on, you only have 2 letters left for the third position, and, lastly, only 1 choice left for the last position. Using the fundamental counting principle, you thus have $4 \times 3 \times 2 \times 1$, or 24 possible arrangements of these letters.

This idea of starting with an integer, and then multiplying down by one less until you get to 1 happens often in supercounting...so frequently that it has a name and a special symbol:

**Definition of Factorial (also known as “n bang”)**

$$n! = n \times (n - 1) \times (n - 2) \times \ldots \times 3 \times 2 \times 1$$

**Example 1:** What’s 5 factorial (that is, symbolically, “5!”)?

**Answer 1:** “5!” just means to start with 5, and multiply down until you get to 1. So it would be $5 \times 4 \times 3 \times 2 \times 1 = 120$.

**Example 2:** What’s 10 factorial (“10!”)?

**Answer 2:** $10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 3,628,800$. Whoa!

As you can see, factorials grow very, very quickly. Check it out!

![Factorial Chart]

Huge numbers. The growth is faster than exponential growth.

Now, imagine this scenario: you take a deck of 52 poker cards, shuffle them well, and then deal all 52 out, face up. Into how many different orders can these 52 cards be arranged?

Well, the quick and easy answer is 52!, isn’t it? However, that masks just how big that number is:

---

This might remind you of the first day of class. 😊
Here are some basic number crunchers for you to try. We’ll build on these ideas in class later.

3. (2 points) $8!$

4. (2 points) $\frac{9!}{6!}$

5. (2 points) $\frac{10!}{5! \cdot 5!}$

6. (2 points) $0!$

7. (2 points) Why do you think #6 has the answer it does?

When dealing with permutations, realize that order matters. Things like passwords, standing in lines, telephone numbers...in any of these situations, if placements of items are changed, the arrangement becomes different. That’s why a “combination” lock is really a “permutation” lock.

So, what if moving the placements doesn’t result in a change? For example, suppose you are playing poker, and are dealt 5 cards from a standard deck of 52. Is this a permutation?

No! Why? Think about it; when you play cards, and you’re dealt a hand, like in poker, or rummy, or go fish, what’s the first thing you usually do? You move the cards around in your hand. You might put that pair of tens together, and make sure the three kings are all next to each other, but...does that make a new hand? No! Those five cards, in any order, are the same hand.

So, if it’s not a permutation, what is it? Well, it’s called a “combination”. A combination is a collection of distinct objects (without repetition) where order does not matter. There is a formula for combinations (just like there is one for permutations); however, as wonderful as I think the formula is, there’s a non-formulaic workaround we will learn later in this course, when we discover the wonderful binomial distribution.


For years, I taught this course with a focus on supercounting and mathematical ideas of probability. Well, the older I get, the more I realize two things:

1) Ten weeks (or, more realistically, a fraction of ten weeks) isn’t long enough to devote to such a complex idea;

2) That’s OK, because it isn’t all that crucial (unless you plan on becoming a math or stats major, in which case you’ll have entire courses devoted to them).

However, problems like the ones you’ve encountered (and are about to encounter) here are valid; I was once told by one of my applied mathematics professors at the University of Delaware that the two most important areas of mathematics are linear programming and Monte Carlo simulation. In this section, I’ll give you an introduction to the latter, through some examples and Excel spreadsheets.

Monte Carlo simulations are performed by using random numbers to represent events in an experiment. For example, think of a company that builds airplanes. One plan of attack would be to build what they think will work, put it...

---

$^c$ This is part of the answer to a problem you haven’t even seen yet! ☺

$^d$ Right before you go all in.
up in the air, and see what happens. That would be expensive and very, very dangerous. Another way would be to model the plan using a computer interface, and “build” the plane, subjecting it to “forces” (“gravity”, “air resistance” “shear”, etc.) represented by numbers in a computer program. If you think about it, this is why flight simulators are used to train pilots. Actually, whenever you hear of “computer models” or “computer simulations” being used, you’re hearing about Monte Carlo methods.

So, let’s begin with an example of how to use Monte Carlo simulation.

**Example 3.** Imagine flipping a coin 10 ten times. What’s the probability that you get exactly 5 heads and 5 tails?

**Answer 3.** As inane as this might seem, this problem (if you remove the context) is an important cornerstone in many aspects of this course and MTH 244, from deciding whether or not touch therapy works, to trying to see if popular opinion about the president has changed. So let’s dive in.

In order to answer this, we’d have to calculate the size of the event space (that is, figure out how many ways you can flip 10 coins, and have exactly 5 come up heads), and then calculate the sample space (the number of ways to flip 10 coins, regardless of what happens). As in many probability problems, the sample space is fairly straightforward: every time you flip a coin, one of 2 things can happen: you get heads, or you get tails. Applying the FCP to 10 coin flips, you get

\[
\text{number of ways to flip 10 coins} = \left(\frac{\text{number of ways}}{\text{to flip coin #1}} \times \frac{\text{number of ways}}{\text{to flip coin #2}} \times \ldots \times \frac{\text{number of ways}}{\text{to flip coin #10}}\right) = 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 1024
\]

Now, the event space...well, that’s a tad more difficult at this point. In the appendix to this project, I will have worked it out for you, but, for now, let’s just agree that it’s too difficult to approach manually. This, in fact, is why Monte Carlo methods were created: to allow approximation of probabilities that were either too cumbersome to do manually, or **impossible** to do manually.

I’d like to use Excel to demonstrate the Monte Carlo solution to this problem. Please open up the spreadsheet “simulationsample.xlsx” from the schedule page. Once you’re in there, select (along the bottom) the “heads and tails” sheet. Here’s a quick look at the top:

![Excel spreadsheet](image)

OK, a bit of explanation. What I’ve done here is create many, many simulations of the experiment “flip a coin ten times.” In fact, I’ve created **1000** simulations (you should check to see I’m not full of it by scrolling down to see trials 1 through 1000). In each of the cells with an “H” or a “T”, I’ve simulated a coin toss by using the command `randbetween(1,2)`. This command randomly creates a 1 or a 2. I’ve assigned an “H” to the result 1, and a “T” to a 2, and that explains the cells that you see under the “Flip” columns.

---

*If you’re wondering about the name “Monte Carlo”, it’s a reference to a casino in Monaco. Apparently, one of the originators of Monte Carlo simulation methods had an uncle who regularly gambled there.*
In the column marked “heads?” I’ve counted (using the `countif` function), in each of the 1000 samples, how many heads occurred. Then, in the gray box out farther right next to “trials with 5 heads?”, I’ve counted in how many of the 1000 trials did 5 heads occur (this is the event space, but I wanted to name it something contextual). “p(5 heads out of 10)” is gotten by dividing the 261 by 1000 (1000 is the sample space).

Now, if you hit the button, the random numbers will re–randomize. Here are the results from above, along with 9 more iterations, done by pressing nine times:

This represents, basically, running the experiment 10,000 times. To get an approximation of the true probability, we can simply average those 10 together, and get the following:

\[ p(5 \text{ heads out of } 10 \text{ flips}) = \frac{0.261 + 0.255 + 0.231 + 0.251 + 0.236 + 0.265 + 0.257 + 0.228 + 0.227 + 0.232}{10} \]
\[ = 0.2443 \]

This result agrees almost perfectly with the exact answer, as you will see in part C of this project. And, if we did more trials, it would agree more and more.

**Example 4:** Roll 5 dice. What’s the chance their pip sum is less than 15?

**Answer 4:** In the same Excel sheet, please open the tab marked, cleverly, “roll 5 dice”.

It looks a tad different than the previous one, eh? This time, I’ve used `randint(1,6)` to “roll” dice. Each row represents the result of throwing 5 dice, and the sum of those 5 dice’s pips are at the end of each row. You can click in any of the cells to see the formulas I used.

---

1 I’m relying on the meaning of probability from your first and second projects here: the percentage of time you will successfully observe the event in question over repeated trials.
In the box out farther, I’ve used the `countif` command to look at all of the sum cells and decide how many 5–dice sums are less than 15 (size of event space). From what follows from our 1000 trials, we can see the probability of rolling a sum less than 15 is about 21%. Replicating this experiment 9000 more times (that is, pressing [ nine more times to generate 9000 more 5–dice rolls), I got the following percentages: 0.222, 0.247, 0.197, 0.222, 0.206, 0.196, 0.22, and 0.23, and 0.206. Averaging these 9 with the previous 0.214, I arrive the experimental (Monte Carlo) probability $p$ (rolling a sum less than 15 when 5 dice are rolled) $\approx 22\%$.

Do you see how technology is absolutely huge with these Monte Carlo type simulations? When the absolutely exact answer is not needed (which is all the time outside of a classroom), simulations work perfectly. Let’s look at one more example to round out the project.

**Example 5:** In a bag are 3 green blocks and 7 blue ones. You reach in and select 4 at random, without replacement. What’s the chance you get two of each color?

**Answer 5:** Refer to the “blocks in a bag” sheet.

This one was a lot of fun to create. I created 1000 trials, each of which consisted of pulling 4 blocks out of a bag containing 10 blocks. As you can see in each “trial” row, there are 3 green (“G”) and 7 blue (“B”) blocks in the “bag” (the entire row). The four columns B through E represent the first four blocks you pulled (that’s why I grayed out the rest). Column L keeps track of how many green blocks I got; since only two possible colors can occur, if I didn’t get green, I must’ve gotten blue. So, from this iteration, the chance of getting two of each color was about 29.6%. If I run 9 more iterations, I arrive at an average (across all 10,000 individual trials) of about 29.8%

**Part 3. The Law of Large Numbers**

In the previous 3 examples, did anyone feel a little queasy? By that, I mean…did any of you say something like, “Wait a darn minute, Sean…how do we even know the answers we’re getting in those simulation re even close to the right ones?”

If so…good! You shouldn’t blindly believe that these methods work. So, here’s a little comparison of each question, with both the Monte Carlo approximate probability, alongside the exact probability (true probabilities are worked in the appendix, if you’re interested):

---

8 The randomization of this was way fun. I encourage you to check it out (the data I used are to the right in the Excel file), and ask if you have questions!

9 These are the values I got with my randomizations. You might get slightly different answers, but they’d still be darn close.
True probability | Monte Carlo probability | Error  
---|---|---
5 heads out of ten flips | 0.246 | 0.244 | 0.8%  
Sum less than 15 on 5 dice | 0.222 | 0.22 | 0.9%  
2 green and two blue blocks | 0.3 | 0.298 | 0.7%  

You see how small those errors are? That’s great! It means there is almost perfect agreement between the theoretical (and, sometimes, awfully hard to calculate) “true” probability values, and their approximations. This is called the “Law of Large Numbers”...if you run randomized simulations to find a probability, the will agree, over large numbers of trials, with the theoretical probabilities they are representing. Weather people use this all of the time, as do financial planners, engineers, and, yes, rocket scientists.

It should be noted, also, that there is nothing special about the 10,000 trials I ran; it just seemed to be a nice, large number. However, more random trials will always result in closer approximations to the true probabilities, since the law of large numbers will kick in with more and more trials.

Monte Carlo approximations are totally awesome. They allow us to attack potentially cumbersome problems in a fairly straightforward way. With the advent of accessible computing technology, I see Monte Carlo as the future of this kind of math. You wanna try some of your own? Sure you do!

***************************

Open the Excel file “simulationexercises.xlsx”. It’s a rather large file; be patient. For each of the following simulation problems, open the sheet mentioned in the problems, and press ten times, each time making note of the probability statements corresponding to whichever question you’re answering (make sure you write down the ten values you got for each on the work you hand in). Average these ten results to get your answer in each. That’s it, and you’re off, supercounting!

Point values for these: 2 points for each...here’s what you have to do! List all ten values you got when you randomized the sheets, and then the average of those ten values. Screenshots are fine, if you prefer!

8. Around 8.15.17, a good buddy texted me the picture at right. He was blown away by the palindromic nature of the 5 – card hand he had dealt himself. I love his question: “Seriously, what could the odds of this be?” Although I texted him the calculation of the probability he desired, I decided it would also be a cool way to use a computer simulation solution. So, to rephrase, let’s figure out the chances of dealing 5 cards to yourself (without replacement) and having them come up in the following order:

```
Matches 5th card
Matches 4th card
Any card
Matches 2nd card
Matches 1st card
```

---

1 A palindrome is a word or phrase that reads the same forwards as backwards. “Racecar”, “Radar” and “Never odd or even.” Numbers can be palindromic, too! 12321 and 9456549 are examples.
9. (2 points) *(use Excel sheet “more blocks”)* Refer back to example 5. Instead of getting 2 of each color, what is the probability of getting 3 blue blocks and 1 green block?

10. (2 points) *(use Excel sheet “shuffle”)* Many of you own an mp3 player (or some device that can play mp3’s). Way more convenient than that clunky Walkman I used to lug around. One feature of my mp3 player (and, I’m sure, yours) is the “shuffle” command; which plays the songs on your player in a “random” order. Suppose (for simplicity) your mp3 player can only hold 10 songs. If your mp3 player is set on its “shuffle” command, and you let it play exactly 10 songs, what is the probability that you will hear each song on your player exactly one time?

11. (2 points) *(use Excel sheet “hold ‘em flush”)* A very popular poker variant is Texas Hold ‘Em. In this game, you are dealt two cards face down, as are all of your opponents. Then, in turn, 5 more cards are turned over by the dealer: the first of the three is called the “flop”, the fourth is the “turn”, and the fifth, the “river”. From those 7 cards, you form a hand from the best five. Use this sheet (which simulates a 4 – person game, where you sit to the left of the dealer) to approximate the chance of a flush (5 cards, all in same suit, but not in rank order) “after the flop”; that is, ignore the potential turn and river cards (I’ve grayed them out).

12. (2 points) *(use Excel sheet “birthday”)* This is the classic birthday problem at which I hinted in MTH 105 (if you took that journey) and maybe in here as well. If you have a room of 25 people, what is the probability that at least two of them have the same birthday? I’ve allowed for 366 days per year, and I’m assuming that all birthdays are equally likely (this probably isn’t true, though). 1 is January first, 2 January second, and so on. You’ll have to scroll over a bit to actually see the results on this one.

---

1 Actually, the term “shuffle” is misleading. When you think of “shuffling”, I bet you think of a deck of cards, in which case, after shuffling, you deal. This implies that you get each card (or in the case of the mp3 player, a song) once. I don’t like getting repeats on shuffle until I’ve heard all songs at least once. Unfortunately, as the teeny-weeniness of this answer shows you, that almost never happens (and this is only with 10 songs! Imagine with the thousands they usually have). However, it wouldn’t be that hard for the mp3 companies to create a shuffler that would enable you to 1) hear your songs in a pseudo – random order, and 2) not get any repeats (Apple already does this). We’ll study this in MTH 244.

2 This probability goes up very fast as you get more and more people in the room.
Example 3: Find $p$(exactly 5 heads in 10 flips of a fair coin)

Exact Answer 3: Above, we learned that the sample space for this problem was 1024. Now...how about the event space? Imagine 10 blank spaces (these represent the 10 flips of the coin):

To find the event space, we need to reckon in how many ways we can place 5 H’s in those spaces (the other 5 spaces will have T’s). Here’s one way:

Here’s another:

Perhaps an obvious one?

We could brute force this, but I (and you) don’t want to. Some of you might recognize this as (sort of) a permutation. For example, let’s build the first outcome from the blank row:

Step 1: Decide where the first “H” will go:

Steps 2 through 5: Decide where the other 5 H’s go.

Now, there were 10 places for the first H, then 9 for the second, and so forth, so the total number of ways to get the 5 H’s in there would be $10 \times 9 \times 8 \times 7 \times 6$, or 30,240 ways.

Of course, that’s not right, now is it? The sample space is only 1024...how could the event space be larger?

Well, it can’t, for a simple reason: we double counted some events. Big time. Here’s why: Imagine building this outcome again:
...but in this order:

You see, you get the same outcome, but in a slightly different order. So, we really can’t count each of these permutations separately, since they both arrive at the same outcome. That’s why we call this something different: it’s called a “combination” (I mentioned those up in part A). But, how many of them are there? Well, clearly, every combination is also a permutation, but some permutations are repeated. All we have to do is remove those repetitions, and here’s a slick way to do it.

In the previous outcome, we just need to figure out how many different ways we could’ve created it, in order, as a permutation. Well, that must be 5*4*3*2*1, right? 5 ways to pick the first H, then 4, then 3, and so on.

So, if we take the number of permutations (30,240) and divide that by 120 (5*4*3*2*1), we’ll get rid of all of the repeated permutations. That leaves us with \( \frac{3024}{120} = 252 \). That’s our event space.

So, \( p(\text{exactly 5 heads in 10 flips of a fair coin}) = \frac{252}{1024} \approx 0.246 \), which agrees very nicely with the 0.244 we got via Monte Carlo simulation.

FYI: combinations like this have a neat shorthand that you might learn at some point. This one’s called “10 choose 5” (because you’re choosing 5 spots from 10), and it looks like this: \( \binom{10}{5} \). Its numerical value is also what you got in the 5th question you did for points.

**Example 5:** In a bag are 3 green blocks and 7 blue ones. You reach in and select 4 at random, without replacement. What’s the chance you get two of each color?

**Exact Answer 5:** I know I skipped 4. I’ll do that one last.

Using the definition of probability, we’ll first find the sample space. This is a combination, since we don’t care what order the blocks come out of the bag. So, there are \( \binom{10}{4} = 210 \) ways to pull 4 different blocks out of this bag of ten.
How many ways to get two green and two blue? Well, there are $\binom{3}{2} = 3$ ways of getting two green blocks and $\binom{7}{2} = 21$ ways of getting two blue ones, so the number of ways to get two greens and two blues is $\binom{3}{2} \times \binom{7}{2}$, or 63.

Thus, the probability of getting two of each color is $\frac{\binom{3}{2} \times \binom{7}{2}}{10 \times 210} = 0.3$, or 30%. Again, we got “close enough for government work”, as my dad says, with 29.8%.

**Example 4:** Roll 5 dice. What’s the chance their pip sum is less than 15?

**Exact Answer 4:** This is a nightmare, plain and simple. Unless you see the pattern, the only way is enumeration. Here goes:

- **Number of ways to roll 5 dice:** $6^5 = 7776$. OK so far...
- **Number of ways to get a sum less than 15** (hold on, folks...it’s a bumpy ride): You start with the number of ways to roll a 5 (1 way...all 1’s), then move to the number of ways to roll 6 (5 ways...one 2 and the rest 1’s), and keep working up to the number of ways to roll 14 (which is awful, but equal to 540). In total, you have:

$$
\begin{align*}
&\binom{4}{0} + \binom{5}{1} + \binom{6}{2} + \binom{7}{3} + \binom{8}{4} + \binom{9}{5} + \binom{10}{6} + \binom{11}{7} + \binom{12}{8} + \binom{13}{9} - 4 \binom{5}{1} - 12 \binom{5}{2} - 2 \binom{5}{3} - 5 \binom{5}{4} - 5 \binom{4}{3} - 1 \binom{4}{1} \\
&= 1722
\end{align*}
$$

which equals 1722 possible unique arrangements. Dividing this by the sample space 7776, we arrive at a probability of about 22.2%. And then we puke. But then we also feel better³, because we realize we didn’t need to find the exact solution to get close enough...our Monte Carlo 22% was just fine, thank you.

³Which often happens after puking.