Life's a Party: Analyzing Party Compositions for Combat Encounters in Dungeons & Dragons Fifth Edition through Monte Carlo Methods

Michael Hauge and Mason Starr

December 7, 2021

Abstract

Dungeons & Dragons is a tabletop roleplaying game in which a party of adventurers explores and fights their way through a fantasy world of the players' creation. There are many different classes that players can choose to handle difficult turn-based combat encounters, including bulky melee-based characters to deal damage, more frail spellcasters with arcane abilities, healers, ranged combatants, and more. In this paper, we present the results of 1.485 million sets of combat encounters for different combinations of classes within an adventuring party, produced by a Python-based simulation of D&D combat. Overall, the most successful party was made up of a Cleric (a healer), a Fighter (a bulky melee build), a Paladin (a bulky healer and damage-dealer), and a Barbarian (the bulkiest class that deals the most damage). While the simulation is not a perfect representation of how human players approach the game, general trends for each class with regards to combat are clear. Bulkier classes, with lots of hit points, hold up better in combat, especially at the level and difficulty simulated; in addition, the total damage a character can deal is much more important to the success of the party than the damage on a per-round basis.

1 Background and Description of Problem

1.1 The d20 System

In 1974, Gary Gygax and Dave Arneson designed Dungeons & Dragons: a tabletop role-playing game that would define an entire genre of games for generations to come. In D&D, players engage in collaborative storytelling by controlling unique characters that slay monsters, go on quests, collect treasure, explore new frontiers, uncover ancient secrets, and more. While the rules of the game provide structure and balance to the adventure, the only limit on the game's content is your imagination.

One player, known as the Dungeon Master (DM), controls the world at large: all the monsters, kings, peasants, and animals. The DM describes the world to the other players, who each control a unique player character. These characters together form an adventuring party. In this analysis, we will simulate combat between an adventuring party and various monsters with Monte Carlo methods, to determine the most efficient party composition.

Dungeons & Dragons is a d20 system; players primarily roll a 20-sided die (d20) to attempt various actions such as attacking a monster, picking a lock, casting a spell, etc. [1]. A typical die roll goes as follows:

1. A player wants to do something - *e.g., attack a monster.*
2. The player rolls a d20 and adds the relevant modifier - *e.g., roll 1d20, add attack modifier*
3. The result is compared against a difficulty class to determine outcome - *e.g., the attack roll total is greater than the monster's armor class, so the attack hits.*

Our project uses the set of rules for the fifth edition of the game, or '5e.'

1.2 Rolling Dice

Many values in D&D are expressed in the form '$n\mathrm{d}k + m$,' which means "roll n k-sided dice and add m to the total." We can define a discrete random variable $R \sim \mathrm{Unif}(1, k)$, which can take values between 1 and k each with probability $1/k$, as the result of rolling a single die. R has mean $\mu_R = (k + 1)/2$ and variance $\sigma_R^2 = (k^2 - 1)/12$.

Let $R_1, \ldots, R_n \overset{\mathrm{iid}}{\sim} \mathrm{Unif}(1, k)$ be the results of rolling n k-sided dice. If we define

$$Y = \sum_{i=1}^{n} R_i \tag{1}$$

then the distribution of Y is given by

$$P(Y = y) = \frac{1}{k^n} \binom{n}{y}_k \tag{2}$$

where $\binom{n}{y}_k$ are the polynomial coefficients, which can be computed by a number of different recursive relations [2].

Because the rolls R_i are independent, the sum Y has mean $\mu_Y = n\mu_R$ and variance $\sigma_Y^2 = n\sigma_R^2$. Thus, if we define the random variable

$$X = Y + m = \sum_{i=1}^{n} R_i + m \tag{3}$$

to be the result of the total expression '$n\mathrm{d}k + m$,' then X has mean $\mu_X = \mu_Y + m = n\mu_R + m$ and variance $\sigma_X^2 = \sigma_Y^2 = n\sigma_R^2$. The Python code used to generate these distributions is given in Appendix A.1.

Overall, however, as the number of dice rolled n increases, the distribution of Y converges to $\mathrm{Nor}(\mu_Y, \sigma_Y^2)$ and the distribution of X converges to $\mathrm{Nor}(\mu_X, \sigma_X^2)$.

It is possible to generalize this analysis further, to account for an arbitrary number of dice with different numbers of faces - i.e., $n\mathrm{d}k + m\mathrm{d}j$. Individual die rolls, which have uniform probability, can be convolved with each other to yield the probability distribution for this arbitrary case [3]. The discrete convolution of two functions $f(x)$ and $g(x)$ is given by:

$$(f * g)(x) \equiv \sum_z f(z)g(x - z) \tag{4}$$

Since convolutions are commutative and associative, they can be combined in arbitrarily large sequences (i.e., $\{[(f * g) * h]...\}$), allowing for the computation of any desired probability distribution. Therefore, given a random variable $S = n\mathrm{d}k + m\mathrm{d}j$, the probability distribution is given by:

$$P(S = s \in n\mathrm{d}k + m\mathrm{d}j) = \sum_r P(r \in n\mathrm{dk})P(s - r \in m\mathrm{d}j) \tag{5}$$

where $P(r \in n\mathrm{d}k)$ and $P(s - r \in m\mathrm{d}j)$ can be computed from further convolutions, the basis of which are the uniform distributions $P(x \in \mathrm{d}k) = 1/k$ and $P(y \in \mathrm{d}j) = 1/j$. For example, if $S = 2\mathrm{d}8 + 1\mathrm{d}4$:

$$P(S = s) = \sum_r P(r \in 2\mathrm{d}8)P(s - r \in \mathrm{d}4) \tag{6}$$

$$P(r \in 2\mathrm{d}8) = \sum_z P(z \in \mathrm{d}8)P(r - z \in \mathrm{d}8) \tag{7}$$

$$\therefore \quad P(S = s) = \sum_r \left(\sum_z P(z \in \mathrm{d}8)P(r - z \in \mathrm{d}8) \right) P(s - r \in \mathrm{d}4) \tag{8}$$

It is therefore straightforward to compute these probability distributions, although the analytical expressions for these series of convolutions are difficult and tedious to obtain, and thus outside the scope of this paper.

1.3 Advantage

There are many scenarios in D&D 5e in which the player, or an enemy, is granted 'advantage' or 'disadvantage.' There are many mechanics that bestow advantage on a player (for example, a player has advantage on an attack roll when attacking a prone enemy), and the DM may choose to grant a player advantage on a skill check or saving throw for contextual or roleplay-based reasons.

The mechanic of advantage is simple: the player rolls their d20 twice, and uses the higher roll. Likewise, for disadvantage, they must use the lower roll. Intuitively, this advantage is undoubtedly good - but how good?

If we define a random variable $R = \mathrm{d}N_{\mathrm{adv}}$ as a roll of an N-sided die with advantage, and two additional random variables R_1 and R_2 as the individual dice rolls, we can represent the value of R as:

$$r = \max(r_1, r_2) = \begin{cases} r_1 & \text{if } r_2 \leq r_1 \\ r_2 & \text{otherwise} \end{cases} \tag{9}$$

We can write the probability distribution by doubling the probability that $r = r_1$ and $r_2 \leq r_1$ (since it does not matter which dice was rolled first), and subtracting off a correction term so we do not count the case when $r = r_1 = r_2$ [4]:

$$P(R = r \in \mathrm{d}N_{\mathrm{adv}}) = 2P(r_1 = r \in \mathrm{d}N)P(r_2 \leq r_1 \in \mathrm{d}N) - P(r_1 = r_2 = r \in \mathrm{d}N) \tag{10}$$

or, written in functional form [4]:

$$P(R = r \in \mathrm{d}N_{\mathrm{adv}}) = \begin{cases} \frac{2r-1}{N^2} & 1 \leq r \leq N \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Likewise, the probability distribution for rolling with disadvantage is given by [4]:

$$P(R = r \in \mathrm{d}N_{\mathrm{disadv}}) = \begin{cases} \frac{2(N-r)+1}{N^2} & 1 \leq r \leq N \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Figure 1 shows the cumulative distribution functions for rolling a d20 with no modifier, with advantage, and with disadvantage. Overall, while the expected value of a standard d20 role is 10.5, the expected value of a d20 roll with advantage is 13.825, while the expected value of a d20 roll with disadvantage is 7.175. Therefore, on average, having advantage on a roll is about equivalent to a +3 bonus, will disadvantage is approximately a -3 penalty. In addition, having advantage almost doubles the chance of a critical hit (rolling a 20, from 5% to 9.75%), and halves the chance of a critical miss (rolling a 1, from 5% to 2.5%).

However, most of the time, the player is rolling a d20 in an attempt to get higher than a certain set number. If that number is around 10, where the difference between having advantage and the standard roll is greatest, having advantage is actually more equivalent to a +5 bonus, while if the player is rolling against a number closer to 1 or 20, the benefit of advantage decreases [5].

1.4 Stats and Modifiers

In Dungeons and Dragons, characters and monsters possess different ability scores, which represent their strengths and weaknesses in different areas. Consult Table 1 for an overview of each ability score explained with a tomato.

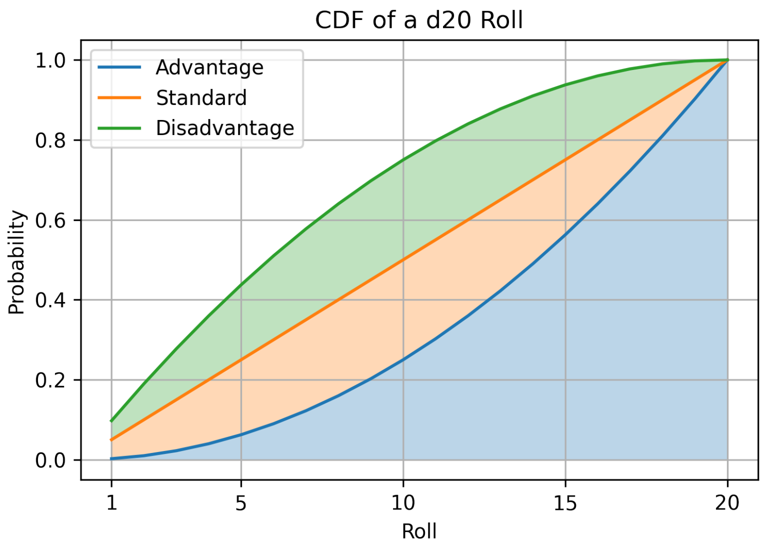


Figure 1: Plot of the cumulative distribution functions of a standard d20 roll, a d20 roll with advantage, and a d20 roll with disadvantage. (Discrete CDF shown as continuous for illustrative purposes.)

Ability Score	Relevant Statistics	Description
Strength	Melee attack rolls and damage	Your ability to crush a tomato
Dexterity	Ranged attack rolls, reflex saves, armor class	Your ability to dodge a tomato
Constitution	Hit points, fortitude saves	Your ability to eat a rotten tomato
Intelligence	Spellcasting (Wizard)	Your ability to know that a tomato is actually a fruit
Wisdom	Spellcasting (Cleric, Druid), will saves	Your ability to know that a tomato does not belong in a fruit salad
Charisma	Spellcasting (Warlock, Sorcerer)	Your ability to sell someone a tomato-based fruit salad

Table 1: Ability scores explained with tomatoes [6].

Ability scores, in conjunction with a character's class, define a unique set of statistics and modifiers. For the purposes of this analysis, the most important statistics are attack modifiers, damage modifiers, armor class (or AC, which determines how hard a character is to hit), hit points, and saving throws.

Modifiers are flat bonuses added to a given die roll. For example, if a character has a +5 attack modifier, they roll a 1d20 and add 5 before comparing to their target's armor class. In this sense, modifiers essentially shift the distribution of given die roll, changing the mean accordingly.

1.5 Party Composition

In this analysis, we will be evaluating parties composed of combinations of the 12 core classes: Barbarian, Bard, Cleric, Druid, Fighter, Monk, Paladin, Ranger, Rogue, Sorcerer, Warlock, and Wizard. When choosing 4 party members from 12 options without replacement, the number of possible party combinations is given by:

$$\binom{12}{4} = \frac{12!}{(4!)(8!)} = 495$$

So, we have 495 possible party compositions to evaluate. Traditional wisdom dictates that a balanced party should have a spell caster, a beefy melee fighter, a healer, and a ranged combatant. The "default" adventuring party is a Fighter, a Cleric, a Rogue, and a Wizard. Will this traditional wisdom hold? Let's find out.

1.6 Class Descriptions

Each player character has a unique class that grants them special abilities. Some classes can cast spells, some are experts with weapons, and some are skilled in other ways. For this simulation, we are using official level three character sheets, download from the Wizards of the Coast website [7]. Base statistics for each character are given in Table 2.

Class	Max HP	AC	Attack Modifiers	Attack Damage
Barbarian	32	14	+5	2d6+3
Bard	24	15	+5	1d8+3
Cleric	27	18	+4	1d8+2
Druid	25	15	+4	1d6+3
Fighter	28	18	+5	1d10+3
Monk	15	15	+5	1d6+4
Paladin	28	16	+5	2d6+3
Ranger	25	15	+7	2d8+3
Rogue	18	15	+5	1d6+3
Sorcerer	23	14	+4	1d4+1
Warlock	24	14	+5	1d10+3
Wizard	17	13	+5	1d10

Table 2: Base statistics for each class. Attack modifiers and damage are listed for only the base attack of each class. It does not account for damage from other sources, such as sneak attack, high level spells, barbarian rage, etc.

Barbarians live for the thrill of combat. They are able to fly into a battle rage, which grants them additional attacks and bonus damage. They also have the ability to be reckless; doing so gives them advantage on all attacks for a round, but gives enemies advantage when attacking the barbarian until their next turn. It's a dangerous tactic, but barbarian have the hit points to justify such a strategy. The barbarian uses their greatsword to attack the most powerful enemy in the room.

Bards are masters of music, and can use magically enhanced tunes to buff their allies. The bard can give their allies "inspiration dice," which can be rolled to add a bonus 1d6 onto any roll. Unfortunately, the bard can only use this ability three times a day, so deciding when to use an inspiration dice is key. Unfortunately, the combat algorithm is not too smart in deciding when to use this ability to maximum effect. Bards also have the soothing words ability, which allows their allies to recover additional health during short rests.

Clerics are devout worshipper of powerful deities, who grant them the ability to cast spells and heal their allies. Clerics are the best healers, and have the ability to restore hit points to their allies in a pinch. The cleric can also cast spiritual weapon against stronger opponents, summoning a magical hammer that pounds them into submission.

Druids are one with nature. They are able to heal their allies with natural magic, though not as well as the cleric. The druid attacks enemies with dual scimitars, potentially landing two hits in a turn. Against powerful enemies, the druid casts flame blade, increasing the damage of their scimitars.

Fighters are highly trained combatants that dominate the battlefield. The fighter attacks the strongest enemy in sight with their sword, doing a respectable amount of damage on a hit. In addition, the fighter can use their bodyguard ability to protect a nearby ally each turn, granting them advantage against all incoming attacks.

Monks are skilled hand-to-hand combatants, using martial arts and ki abilities to best their opponents. Our monk is a halfling, a short fellow who uses the size of opponents against them. The monk can use flurry of blows three times per combat to get two extra attacks, potentially tripping the target to grant all allies advantage against the downed enemy. As a halfling, the monk also is lucky, which allows them to reroll natural 1s on attack rolls.

Paladins are holy warriors that strive to defeat the enemies of their gods. They have a small number of spells that they can use to lay on hands or smite. Lay on hands can be used to heal allies, and smite is used to get a substantial bonus to damage as divine wrath is invoked against the enemy.

Rangers are hunters that stalk the wilderness, often feeling more at home in a tent than in an inn. Our ranger is a talented archer who uses their colossus feller ability to almost double their damage output. Notably, the ranger has the highest attack modifier of any class. The ranger also chooses to attack the weakest enemy in a battle, to pick off minions before attacking a stronger opponent.

Rogues are sneaky tricksters who rely on their skills to make it out of sticky situations. Our rogue attacks with a shortsword and a dagger, and has the potential to land two attacks in one turn. The rogue also uses their sneak attack ability to do extra damage if they have advantage on the enemy. For this simulation, we assume that the rogue has advantage 50% of the time.

Sorcerers are born with magic coursing through their veins. Our sorcerer in particular gets their power through a connection to a frost dragon. They can use frost breath to attack multiple enemies at once, cast blur to grant allies advantage against all attacks, and cast magic missile for guaranteed damage. If all else fails, they can cast their trusty ray of frost.

Warlocks are granted their magic from a mysterious and powerful patron. They can cast scorching ray on powerful opponents to do a ton of damage, but spend most of their time casting eldritch blast, which does about as much damage as a fighter's sword.

Wizards learn magic through years of studying and practice. Our wizard is a pyromancer; they can cast flaming sphere to roll around the battlefield and do damage to enemies each turn, or cast burning hands to toast multiple enemies at once. Mostly, they cast fire bolt. Though powerful spellcasters, wizards are very frail and do not have a lot of hit points.

1.7 Monsters

D&D 5e classifies monsters by challenge rating (CR), or in general the difficulty a party will have when facing a particular creature. Ten total monsters were implemented, at the following CRs: 1/2 (Orc, Giant Wasp), 1 (Hippogriff, Specter), 2 (Quaggoth), 3 (Displacer Beast, Hell Hound), 4 (Shadow Demon, Lizard King), and 5 (Troll). All statistics and other information about D&D monsters were taken from the official Monster Manual [8].

An interesting note about monsters in D&D 5e is that their maximum HP is given as the result of a roll of the form $ndk + m$, like those discussed in section 1.2. Figure 2 shows the distributions for the maximum HP of each of the programmed monsters, illustrating how, if n is sufficiently high, these discrete distributions begin to look like normal distributions.

As an aside, it is possible to investigate these monsters further to quantify their difficulty beyond a simple CR, and to investigate the game's balance. Figure 3 shows the damage that each monster deals on a hit, plotted against the monster's maximum HP. The data are plotted as ellipses, with their centers aligned with the means of each variable and their major and minor axes given by three times the standard deviation in each direction.

Overall, there is a high positive correlation between a monster's damage dealt on a hit and its maximum HP (0.7572). This indicates that, as monsters get more difficult, their damage output scales with their health pool. In addition, the different CR groupings of the monsters are visible, but they do not match up perfectly with what might be expected. For example, the Shadow Demon does not seem to deserve its designation as CR 4, having a lower damage output and maximum HP than the Displacer Beast, which is CR 3. However, the Shadow Demon has additional mechanics and abilities (for example, invisibility) that are not easily represented in this simplified plot.

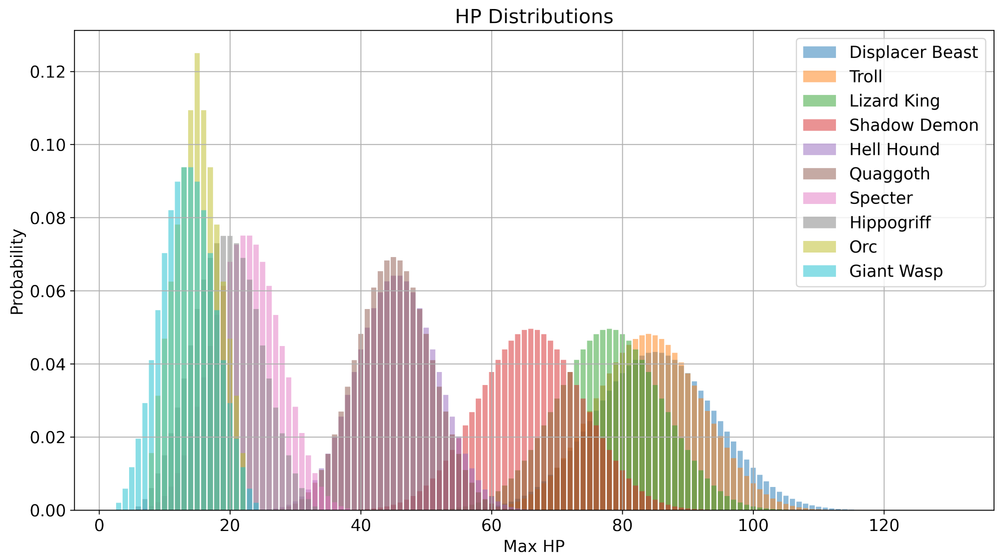


Figure 2: Maximum HP distributions for selected monsters.

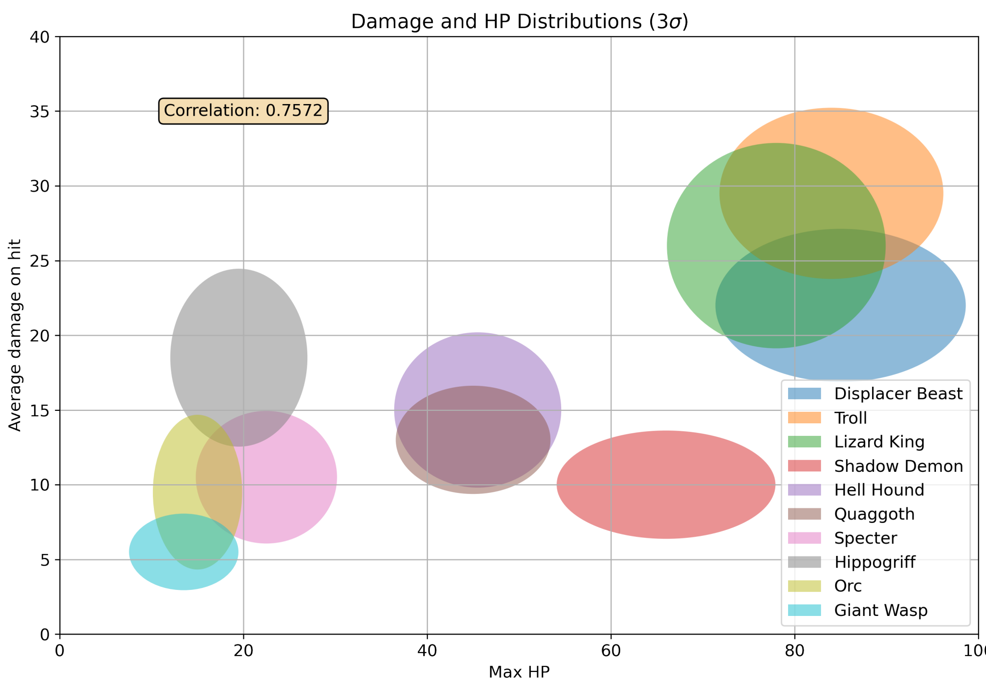


Figure 3: Maximum HP and average damage on hit distributions for selected monsters. The ellipses shown have major/minor axes equal to $\pm 3\sigma$.

2 Implementation

All 2,000+ lines of our code for implementation of classes, monsters, dungeons, and combat, as well as for additional analysis and data processing, can be found at the following GitLab link: https://gitlab.com/dillan1/dnd_5e_party_evaluator/ [9].

2.1 Classes (the Programming Kind)

We elected to program our simulation in Python, to take advantage of statistics-based Python libraries and make use of Python's simple object-oriented programming. A basic Unified Modeling Language class diagram is shown in Figure 4

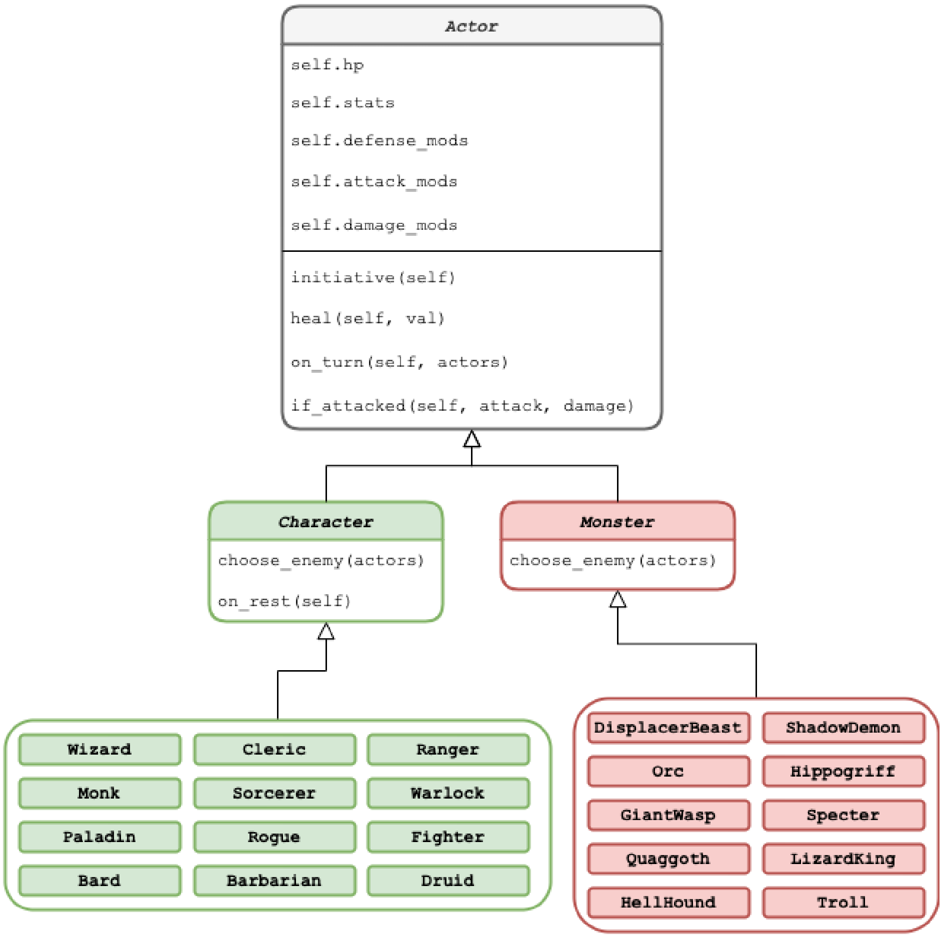


Figure 4: Unified Modeling Language class diagram showing the relationships between Characters, Monsters, and Actors.

All creatures, both Characters and Monsters, are Actors. They have statistics such as strength, dexterity, armor class, and more. They also have functions to handle rolling initiative, healing, attacking, taking damage, deciding what to do on their turn, etc. Both Characters and Monsters have coded strategies to determine their enemies, but only Characters have the ability to short rest outside of combat.

2.2 Combat Encounters

In this analysis, we simulate a series of combat encounters between an adventuring party and monsters. To begin a combat, all Actors roll initiative by rolling a 1d20 and adding their dexterity modifier. This determines the turn order of the combatants: highest initiative takes their turn first, followed by second highest, and so on.

On their turn, an actor follows a programmed pattern of behaviour. Although the exact implementation varies from actor to actor (see section 2.1), most actors pick an enemy to attack, then attempt to attack them. Some Actors have more advanced logic, such as Sorcerers only using their powerful spells on high CR opponents. If an attack hits (usually by an attack roll meeting or exceeding a target's armor class), damage is dealt and the target's hit points are reduced. If an actor reaches 0 hit points, they are assumed to be dead. Each actor takes their turn in initiative order, repeating until either all monsters are dead or the whole party is dead, at which point the combat ends. A sample of the verbose output of a combat simulation is shown below.

```
Roll for initiative! Turn order: Orc 2, Orc 3, Fighter, Orc 1, Wizard, Rogue, Cleric
Orc 2 attacks Wizard and deals 9 damage
Orc 3 attacks Wizard and deals 7 damage
Fighter protects Rogue
Fighter attacks Orc 3 with a big sword and deals 12 damage
Orc 1 attacks Rogue and deals 8 damage
Wizard attacks Orc 2 with burning hands and deals 10 damage
Wizard attacks Orc 3 with burning hands and deals 11 damage
Orc 3 is slain!
Wizard attacks Orc 1 with burning hands and deals 12 damage
Rogue attacks Orc 1 with shortsword and misses
Rogue attacks Orc 1 with dagger and deals 2 damage
Orc 1 is slain!
Cleric heals Wizard
Wizard heals 15 hit points
Orc 2 attacks Rogue and deals 13 damage
Rogue is slain!
Fighter protects Wizard
Fighter attacks Orc 2 with a big sword and misses
Wizard attacks Orc 2 with fire bolt and misses
Cleric attacks Orc 2 with hammer and misses
Orc 2 attacks Wizard and misses
Fighter protects Wizard
Fighter attacks Orc 2 with a big sword and deals 13 damage
Orc 2 is slain!
Combat is over!
Fighter takes a short rest
Wizard takes a short rest
Cleric takes a short rest
```

This model of combat is vastly simplified compared to actual combat in Dungeons and Dragons. When playing the game for real, there are many factors that affect combat: perhaps enemies are out of reach, dangerous hazards threaten the battlefield, or a time sensitive objective drives the players to change their combat strategy. There are hundreds of possible options for a given actor on their turn. In our simulation, things are simplified: any actor can be attacked at any time, and factors such as movement and objective are not considered. Despite these limitations, our simulation gives an accurate representation of general trends for each character class.

2.3 Dungeons and Rests

To simulate a variety of combat encounters, we ran each party through a series of simulated dungeons. Each dungeon contained multiple rooms, with each room containing some monsters for the party to fight. The party enters a room and fights the monsters inside. If the party fails, the dungeon run is over. We record statistics for each party member: damage dealt, turns alive, damage per round (DPR), etc. If the party is victorious, they take a short rest before moving on the next room, as shown in Figure 5. If the party successfully clears every room in the dungeon, statistics are recorded and the run end.

Short rests provide an opportunity for party members to heal. While taking a short rest, a character has the option to spend one or two of their three "hit dice" to recover some hit points. In this simulation, characters will spend one hit die if they are below 50% hit points, and will spend two hit dice if they are below 25% hit points. Some classes, such as the Bard, have abilities that effect short rests.

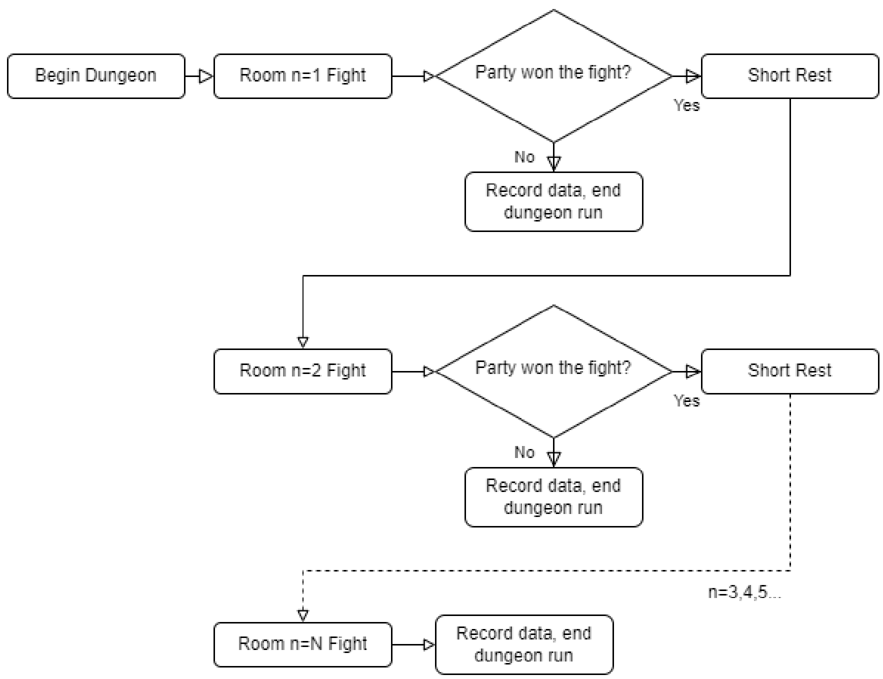


Figure 5: Flowchart depicting how a party progresses through a dungeon with N rooms.

We elected to run parties through three separate dungeons, detailed in Figure 6. Each dungeon has three rooms filled with a variety of monsters. These dungeons are quite challenging and most parties will fall prey to the beasts within. However, some parties will be able to clear the dungeon if they are effective enough and have luck on their side.

2.4 Monte Carlo Simulation

A Monte Carlo algorithm employs repeated use of random sampling to obtain numerical results. In our case, we are sampling die rolls and modeling the results using the rules of D&D. While one dungeon run is enough to obtain some base statistics, we need more samples.

To assess the 495 different party combinations, the parties were sent through the dungeons described above 1000 times each, for a total of 1.485 million dungeon simulations. For each character class in each simulation, the total damage dealt, total damage taken, number of rounds lived, and the room in which they died (if any) was recorded, as well as the overall success or failure of the party to beat the dungeon. This data was then unpacked and analyzed by party composition, individual class, and pairings of classes.

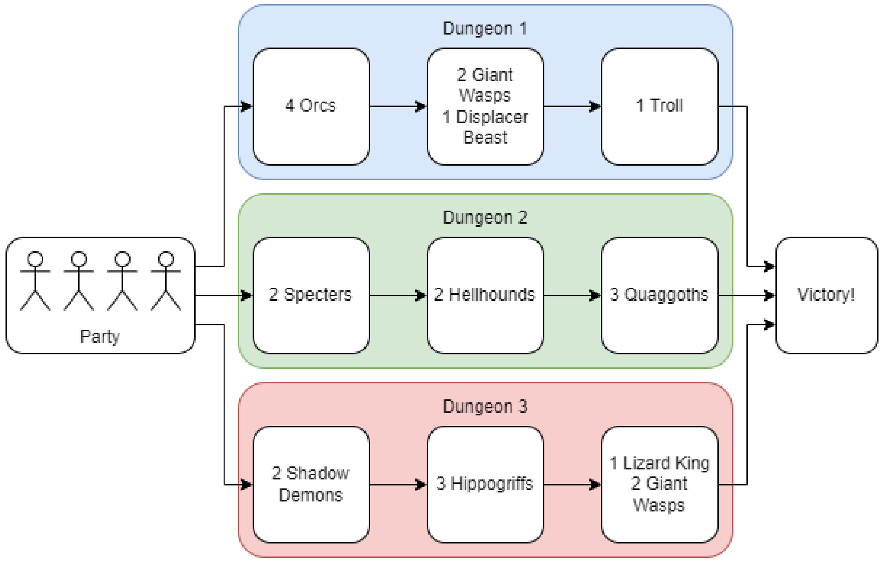


Figure 6: Dungeons and contents

3 Main Findings

3.1 Party Composition Analysis

Table 3 shows the average, standard deviation, minimum, and maximum success rates for each of the three dungeons, as well as the parties that achieved the minimum and maximum success rates.

Dungeon	Mean Success Rate	Std. Dev	Minimum	Maximum
1	0.082	0.070	0.001 (Wizard/Druid/Sorcerer/Monk)	0.305 (Fighter/Bard/Ranger/Barbarian)
2	0.114	0.062	0.010 (Cleric/Druid/Sorcerer/Monk)	0.338 (Cleric/Fighter/Ranger/Barbarian)
3	0.039	0.036	0.000 (Wizard/Druid/Sorcerer/Monk)	0.240 (Cleric/Fighter/Paladin/Barbarian)
All	0.078	0.047	0.013 (Cleric/Druid/Sorcerer/Monk)	0.279 (Cleric/Fighter/Paladin/Barbarian)

Table 3: Overall success rates for the three simulated dungeons.

There is clearly synergy between the Cleric, Fighter, and Barbarian, which is in line with the player's intuition: a healer, paired with two bulky party members that deal massive damage, is a recipe for success. On the other hand, when the Druid, Sorcerer, and Monk, all relatively frail classes, are grouped together, they do not do well at all. Spellcasters like the Wizard, Sorcerer, and Warlock, and ranged attackers like the Ranger, are at a disadvantage in the simulation, which does not account for the fact that they would mostly hang around the edges of the battlefield, away from the fray.

From Figure 7, which plots the distributions of the success rates, it is possible to see that, while the rate for Dungeon 1 exhibits a somewhat bimodal distribution, the distributions for Dungeon 2 and Dungeon 3 look more like negative binomial distributions, with tails stretching out to the right.

Table 4 shows the frequency of each class appearing in the top ten and bottom ten parties in terms of overall success rate.

From the table, a few trends are clear. For one, having a Barbarian and a Fighter, two bulky, physical, damage-dealing classes, is essential for having a top-performing party. On the other hand, the pairing of Druid and Sorcerer, two relatively frail caster classes that rely on magic and do not have good defense, is certain to doom the party. Another point of note is the Cleric, which appears with significant frequency in both the top ten and bottom ten. The Cleric is at its best when it is not necessarily being relied upon to do damage, but is instead using its healing abilities to support other classes that do significant damage (for

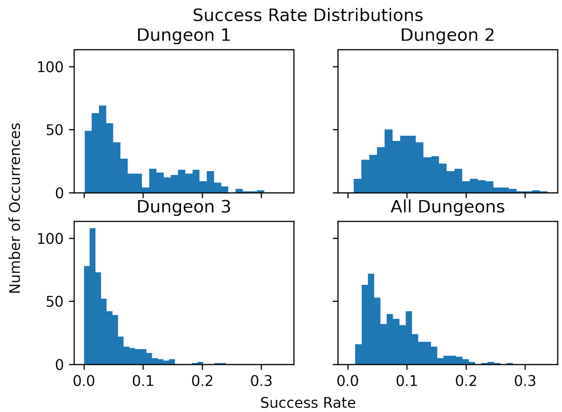


Figure 7: Layout for dungeons 1 through 4.

Class	Top 10	Bottom 10
Barbarian	1.0	0.0
Bard	0.3	0.4
Druid	0.2	1.0
Monk	0.0	0.4
Ranger	0.4	0.0
Warlock	0.0	0.3
Paladin	0.3	0.0
Sorcerer	0.0	1.0
Cleric	0.6	0.4
Fighter	1.0	0.0
Rogue	0.1	0.4
Wizard	0.1	0.1

Table 4: Frequency of each class appearing in the top ten or bottom ten performing parties.

example, when grouped with a Fighter, Paladin, and Barbarian). On the other hand, when grouped with frailer classes (for example, a Druid, Sorcerer, and Monk), the Cleric does not provide much help to the party, since it cannot deal the necessary damage and it cannot heal its fellow party members fast enough.

3.2 Damage per Round

As illustrated in Figure 8, there is a negative correlation between a class's damage per round (that is, the average damage the class inflicts on an enemy during each round of combat) and that class's maximum hit points. This makes sense, from a game balance perspective; classes that are able to do more damage are balanced out by being more frail, encouraging players to be careful with this high-risk, high-reward play style. The notable exception, found in the upper right corner of the plot, is the Barbarian. The Barbarian is a rather unbalanced class at the level considered in this paper, and its status as an outlier is a frequent theme of the findings.

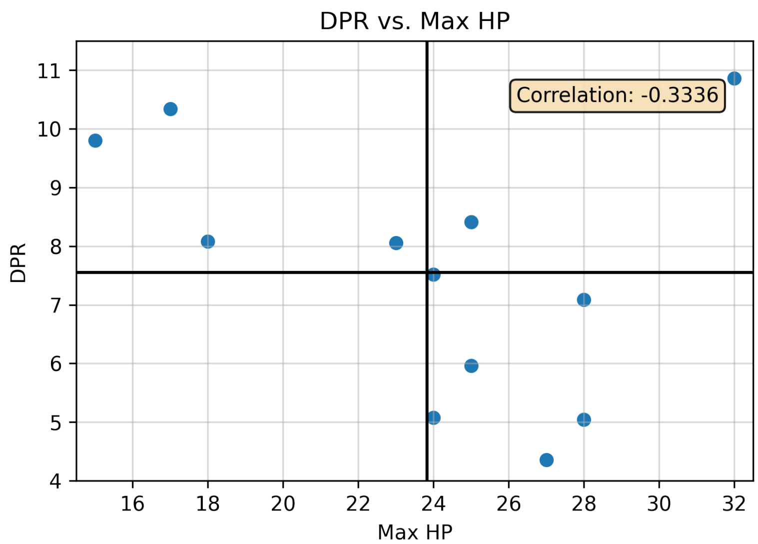


Figure 8: Correlation between each class's damage per round and maximum hit points. Black lines are plotted at the mean of each variable.

The overall correlation between class damage per round (DPR) and maximum hit points (HP) is -0.3336; if the outlier of the Barbarian is not considered, it is significantly larger in magnitude, at -0.7812.

3.3 Individual Class Performance

For a more detailed look at how each individual class contributes to party success, it is necessary to break the data down by class. Figure 9 shows the overall success and survival rates for each class, as well as the ratio of these rates. This ratio is, in a way, a measure of how important each class is to the success of parties containing that class. For example, the Barbarian and Cleric have the highest ratio of survival rate to success rate; for each of them, they survive to the end of about 4/5 dungeons they beat, indicating that they play very important roles in the success of their parties. On the other hand, the Wizard survives to the end of only around 50% of the dungeons its parties beat.

Out of the classes with the top five survival rate to success rate ratios, four deal heavy damage (Barbarian, Ranger, Paladin, and Fighter), three are bulky (Barbarian, Paladin, and Fighter), and two are effective healers (Paladin and Cleric), information which can be used to inform party composition.

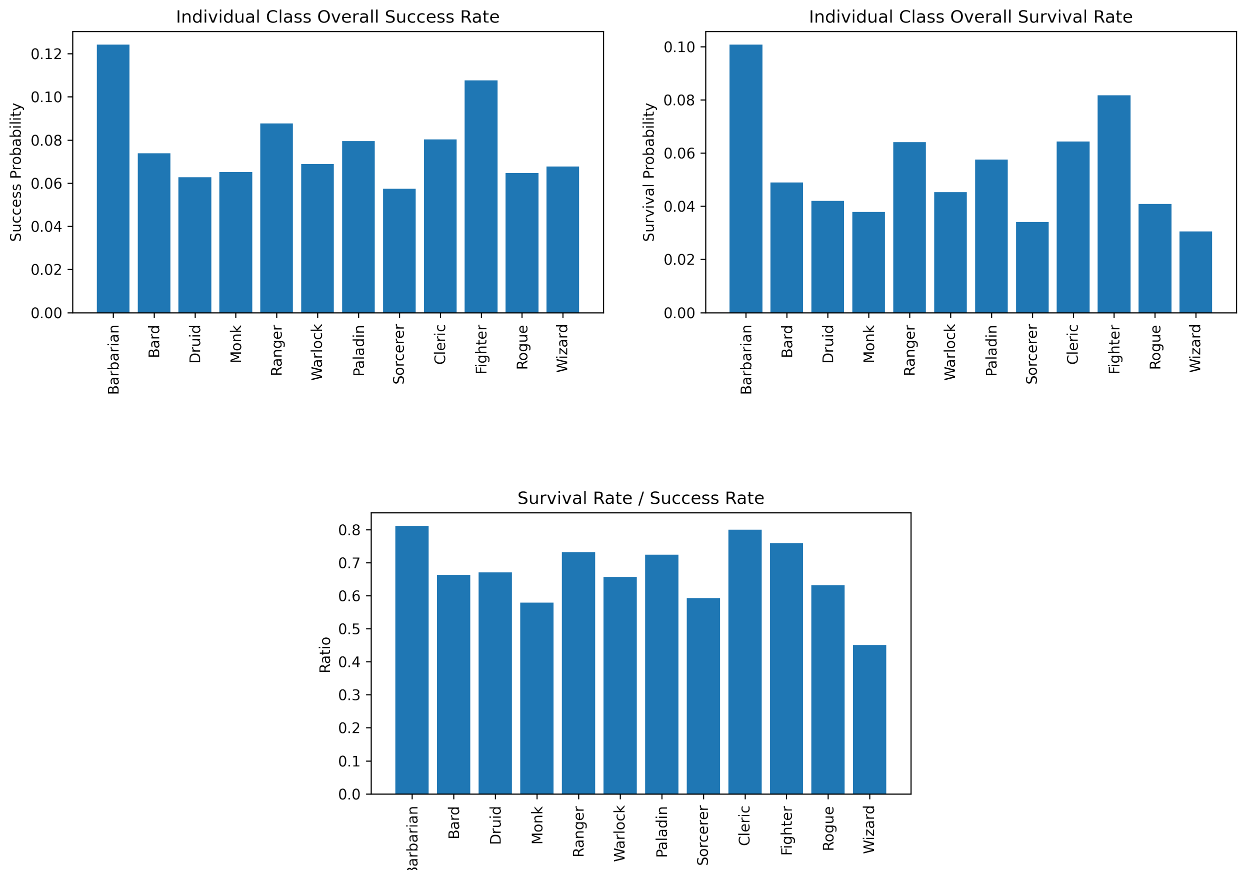
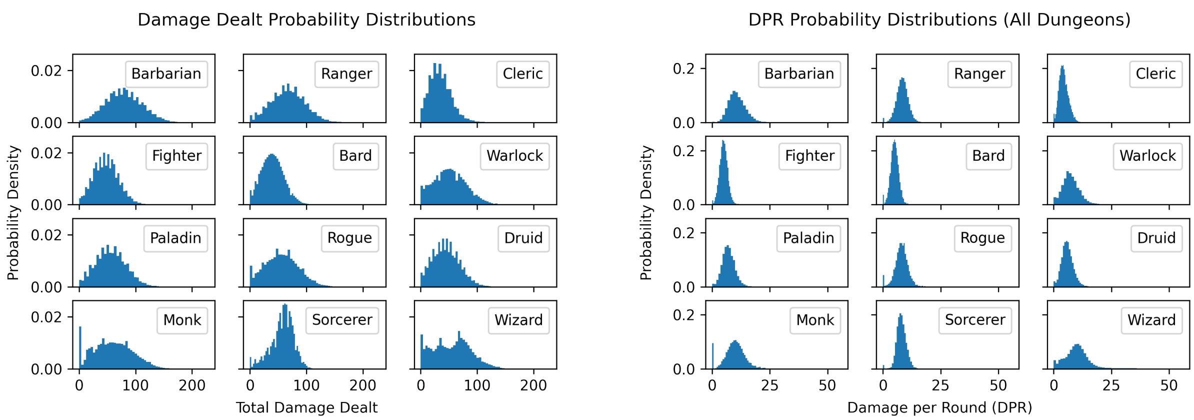


Figure 9: Data on overall success rate, survival rate, and ratio of survival rate to success rate, broken down by class.

Delving deeper into the specific data recorded during each simulation (damage dealt, damage taken, and rounds survived), it is possible to gain more insight. (Tables summarizing the mean and standard deviation of each of these metrics, as well as damage per round, may be found in Appendix B.)

From an overall perspective, the Barbarian deals the most damage, and takes the most damage, out of all the classes, by a significant margin. This makes it incredible useful for a party, both by dealing damage to enemies and taking hits from enemies that otherwise its fellow party members would take. The Barbarian also has a consistently high damage per round (DPR), although surprisingly the Monk and Wizard also have high DPRs. This may be, counter-intuitively, due to the fact that the Monk and Wizard do not tend to survive for very long; additionally, both of these classes have the capacity for high damage output (the Monk through its Flurry of Blows ability, and the Wizard through its higher-level spell slots). The Cleric dealt the least damage overall, but it provides different benefits to the party through healing.

The distributions of total damage dealt and DPR are plotted in Figure 10. With their consistent round-to-round damage output, not dependent on abilities or spells, the Barbarian, Ranger, Fighter, and Paladin show total damage dealt distributions that look close to Gaussian (see Figure 10a). The other distributions tend to be skewed right, indicating that sometimes an ability or spell allows that class to deal significantly more than their average damage (for example, doing double damage on a critical hit), except that of the Sorcerer, which is skewed left. The Wizard is another exception - its damage dealt distribution looks like the sum of three right-skewed distributions, which may be conjectured to correspond to simulations, from left to right, in which the Wizard is: killed before using any spell slots, killed after using only first-level spell slots, and able to use its second-level spell slots. Figure 10b tells broadly the same story, again with the Wizard having a higher variance of DPR, since DPR depends heavily on what spells the Wizard is able to cast.



(a) Total damage dealt distributions.

(b) DPR distributions.

Figure 10: Class-specific damage data distributions.

The data on rounds alive is less useful, since it is very different for each of the three dungeons, which vary in difficulty. However, Figure 11 shows a representative example of what the rounds alive distribution looks like for a given class in a given dungeon (in this case, the Wizard in Dungeon 1). The distribution looks very much like a negative binomial distribution, which makes intuitive sense: in each round, there is a certain probability (i.e., p) that a character takes damage, and the character dies after taking damage a certain number of times (i.e., r), corresponding to something like a $\text{NegBin}(r, p)$ distribution.

It is also possible to examine the correlations between class-specific statistics and the overall success rates of parties containing that class, as shown in Figure 12. (On each plot, the mean value of each variable is plotted as a black line.)

Unsurprisingly, there is a high positive correlation between overall success rate and maximum HP. At a low level, hit points are a valuable resource, and the bulkiest class (Barbarian) has more than twice as many as the frailest (Wizard). However, it is interesting to note that there is no significant correlation between

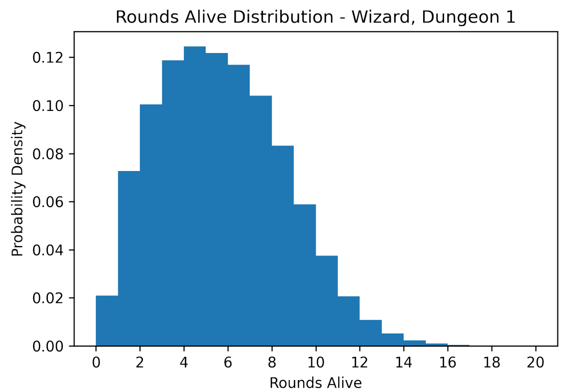
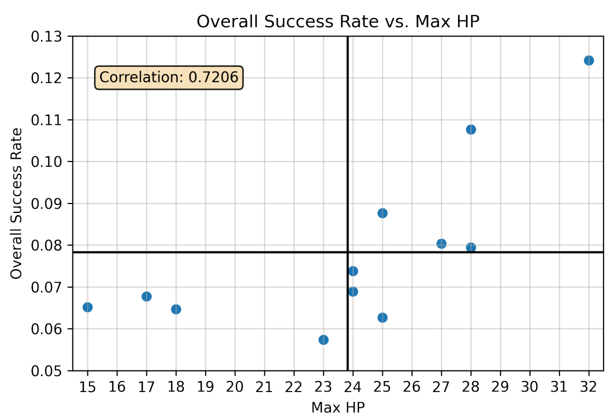
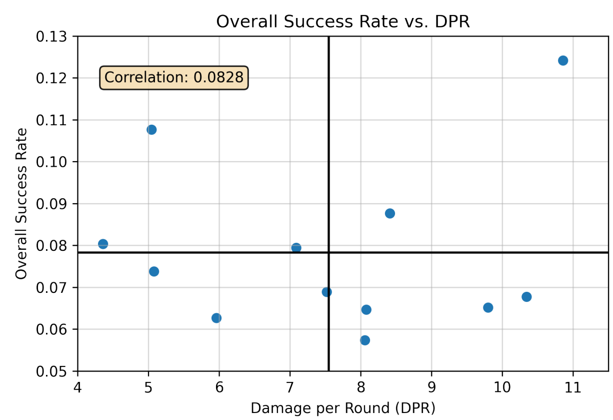


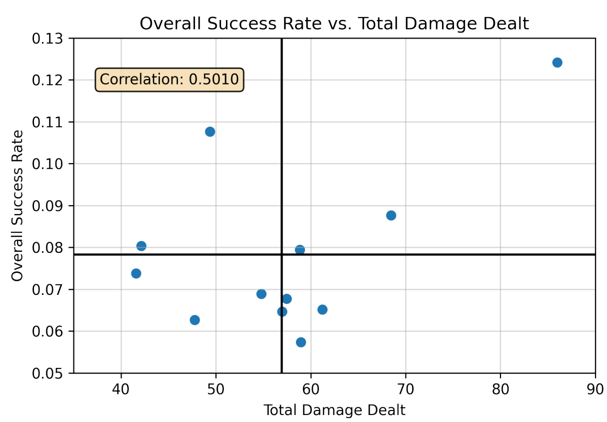
Figure 11: Representative rounds alive distribution: Wizard, Dungeon 1.



(a) Correlation between success rate and maximum HP.



(b) Correlation between success rate and DPR.



(c) Correlation between success rate and total damage dealt.

Figure 12: Correlations between overall success rates and class statistics.

success rate and DPR. Even dismissing the outliers (the Wizard in the upper left and the Barbarian in the upper right), there is no real trend. Figure 12c provides the answer to this conundrum. Rather than DPR, it is total damage dealt that has a positive correlation with success rate. Essentially, it is more important to stay alive and continue doing consistent damage than to do greater damage per round but be more likely to die.

4 Conclusions

4.1 Party Composition and Gameplay Balance

Though certain classes outperformed others in our simulations, we cannot draw the conclusion that some classes are definitively better than others. The character classes we modeled are unique characters; for example the Druid we modeled cannot be representative of every possible Druid in the game.

Additionally, Our simulation does not properly capture the complexity that Dungeons and Dragons offers, and thus cannot account for strategies that would increase or decrease a character's success rate. For example, we found that characters with low hit points tend to fail more than those with high hit points. Our model assumes that any actor can attack another actor at any time; a property that favors melee combatants and penalizes ranged combatants such as Wizards that in practice would be keeping their distance to take less damage when in combat with melee-based enemies. However, having a party with high hit points is very valuable at low level, as it makes a party more resilient.

When extrapolating results from this paper to an actual game of D&D, it is important to remember that combat is not the only thing that matters. There are countless obstacles that cannot be solved by a barbarian swinging a sword. Rogues can pick locks to bypass doors, druids can calm a wild animal by speaking directly to it, rangers can track their quarry for miles, and wizards can cast any number of useful spells. When creating a D&D character, maximizing DPR should not be the main concern.

4.2 Future Work

The current simulation architecture is able to model encounters between any number of combatants and any party size. Future work could include analyzing parties of different sizes, or even modeling an all out war between hundreds of fighters and hundreds of monsters!

More content could be added to the simulation. There are hundreds of monsters, dozens of character options, and mechanics like traps and legendary actions that could be implemented with more time.

Dungeons & Dragons is an immensely complicated game, filled with constant decision making: something that humans are good at, computers less so. With more time and resources, we could improve the AI of monsters, introduce more game mechanics such as feinting or death saves, and more. One of the biggest but most complicated improvements we could make is introducing grid-based combat, a key element of combat in D&D. Doing so would allow for the actor AI to execute more complicated strategies: rogues could flank to trigger sneak attack, wizard could keep back to stay safer in combat, and paladins could interpose themselves between their allies and the enemy. However, implementing these changes is incredibly complicated, and at that point we would basically be developing a video game with a grid display, pathfinding algorithms, advanced actor AI, etc. - all things that are out of the scope of this paper.

Finally, future work could include using different metrics to model party success. For example, the party could be run through a dungeon of infinite rooms, with their success based on how far they progress before perishing.

References

[1] Mike Mearls and Jeremy Crawford. *Dungeons & Dragons Player's Handbook*. Wizards of the Coast, Renton, WA, 2014.

[2] Camila Caiado and Pushpa Rathie. Polynomial coefficients and distribution of the sum of discrete uniform variables. January 2007. Available at https://www.researchgate.net/publication/228457326_Polynomial_coefficients_and_distribution_of_the_sum_of_discrete_uniform_variables. Accessed 2021-11-09.

[3] Justin Willmert. Basic D&D Statistics: Sums of Dice Rolls. https://justinwillmert.com/articles/2020/basic-dd-statistics-sums-of-dice-rolls/, 2020. Accessed 2021-12-01.

[4] Justin Willmert. Basic D&D Statistics: Dice Rolling with Advantage. https://justinwillmert.com/articles/2020/basic-dd-statistics-dice-rolling-with-advantage/, 2020. Accessed 2021-12-01.

[5] Zero Hit Points. Advantage in D&D 5. http://zerohitpoints.com/Articles/Advantage-in-DnD-5. Accessed 2021-12-01.

[6] u/tan620. D&D Stats Explained with Tomatoes. https://www.reddit.com/r/DnD/comments/1s9l2g/dd_stats_explained_with_tomatoes/. Accessed 2021-12-07.

[7] Wizards of the Coast. Official Character Sheets. https://dnd.wizards.com/charactersheets. Accessed 2021-12-01.

[8] Wizards RPG Team. *Dungeons & Dragons Monster Manual*. Wizards of the Coast, Renton, WA, 2014.

[9] Mason Starr, Michael Hauge, and Dillan McDonald. DND_5E_Party_Evaluator. https://gitlab.com/dillan1/dnd_5e_party_evaluator/, 2021.

A Code

A.1 Rolling dice in the form '$nd k + m$'

```
from itertools import product
from collections import defaultdict

# PMF of a sum of n discrete RV's distributed Unif(a,b)
# Derived from https://stackoverflow.com/questions/69791287/how-to-get-the-probability-of-
                                                        each-possible-outcome-for-a-multinomial-case
def discrete_uniform_sum_pmf(a,b,n):
    du_pmf = {i: 1/(b-a+1) for i in range(a,b+1)}
    du_sum_pmf = {0: 1}

    for i in range(n):
        new_sum_pmf = defaultdict(float)
        for prev_sum, dice in product(du_sum_pmf,du_pmf):
            new_sum_pmf[prev_sum + dice] += du_sum_pmf[prev_sum]*du_pmf[dice]
        du_sum_pmf = new_sum_pmf

    return list(du_sum_pmf.items())

# PMF of a roll of XdY+Z
def roll_dist(X,Y,Z):
    pmf = discrete_uniform_sum_pmf(1,Y,X)
    rolls = []
    for pair in pmf:
        ls = list(pair)
        ls[0] += Z
        rolls.append(ls)
    return rolls
```

A.2 Rolling a d20 with advantage

```
import numpy as np

# PMF of a roll of an N-sided die with advantage
def adv_pmf(r,N):
    return (2*r-1)/(N*N)

# PMF of a roll of an N-sided die with disadvantage
def dis_pmf(r,N):
    return (2*(N-r)+1)/(N*N)

pmf_a = np.zeros(20)
cdf_a = np.zeros(20)
pmf_d = np.zeros(20)
cdf_d = np.zeros(20)
cdf_reg = np.zeros(20)

# Compute CDFs
for i in range(1,21):
    pmf_a[i-1] = adv_pmf(i,20)
    cdf_a[i-1] = sum(pmf_a[:i])
    pmf_d[i-1] = dis_pmf(i,20)
    cdf_d[i-1] = sum(pmf_d[:i])
    cdf_reg[i-1] = 0.05*i
```

B Summary of Class-Specific Data

Class	Damage Dealt (Avg.)	Damage Dealt (Std. Dev.)	Damage Taken (Avg.)	Damage Taken (Std. Dev.)	Rounds Survived (Avg.)	Rounds Survived (Std. Dev.)	DPR (Avg.)	DPR (Std. Dev.)
Barbarian	80.275	38.443	43.693	14.391	7.645	2.664	10.404	3.758
Bard	36.991	19.844	40.048	13.095	7.898	3.406	4.695	1.820
Druid	38.792	23.688	41.156	12.815	7.694	3.295	5.043	2.433
Monk	52.694	34.204	25.913	9.527	5.615	3.191	9.221	4.792
Ranger	63.387	31.016	40.163	12.426	7.933	3.211	7.954	2.665
Warlock	47.241	30.635	38.273	11.606	6.908	2.971	6.781	3.814
Paladin	50.931	27.450	38.869	11.468	7.901	3.107	6.509	2.881
Sorcerer	55.938	23.873	36.646	10.865	7.118	3.219	8.103	2.657
Cleric	32.815	22.165	39.496	12.899	9.328	3.786	3.422	1.636
Fighter	42.545	21.436	42.974	12.367	9.514	3.396	4.473	1.765
Rogue	50.930	29.795	30.934	10.520	6.529	3.314	7.665	3.027
Wizard	56.871	25.199	26.847	8.943	5.217	2.875	13.215	8.142

Table 5: Summary of class-specific data for Dungeon 1.

Class	Damage Dealt (Avg.)	Damage Dealt (Std. Dev.)	Damage Taken (Avg.)	Damage Taken (Std. Dev.)	Rounds Survived (Avg.)	Rounds Survived (Std. Dev.)	DPR (Avg.)	DPR (Std. Dev.)
Barbarian	78.510	24.016	45.338	12.452	9.067	2.358	8.794	2.221
Bard	43.594	19.051	37.502	11.778	9.007	3.076	4.853	1.637
Druid	51.506	21.460	38.876	11.832	8.947	2.946	5.864	2.114
Monk	66.300	33.713	24.055	8.625	6.799	2.979	9.697	4.099
Ranger	71.683	28.117	36.569	10.940	8.636	2.698	8.325	2.464
Warlock	56.541	25.065	36.293	10.018	8.301	2.791	7.007	3.112
Paladin	62.312	23.934	36.918	10.712	9.281	2.734	6.847	2.386
Sorcerer	58.915	15.707	35.101	9.452	8.078	2.767	7.630	1.850
Cleric	37.772	16.451	37.402	11.990	10.545	3.322	3.613	1.303
Fighter	53.393	20.792	39.798	12.269	10.447	2.933	5.141	1.640
Rogue	61.789	28.006	28.485	9.300	7.602	2.927	8.078	2.629
Wizard	53.932	39.511	25.204	8.610	6.801	2.996	7.151	4.198

Table 6: Summary of class-specific data for Dungeon 2.

Class	Damage Dealt (Avg.)	Damage Dealt (Std. Dev.)	Damage Taken (Avg.)	Damage Taken (Std. Dev.)	Rounds Survived (Avg.)	Rounds Survived (Std. Dev.)	DPR (Avg.)	DPR (Std. Dev.)
Barbarian	99.222	30.515	47.604	12.688	7.715	2.445	13.382	3.625
Bard	44.100	19.087	40.244	12.118	7.790	2.843	5.688	1.676
Druid	52.968	23.745	40.863	11.666	7.779	2.749	6.970	2.689
Monk	64.687	36.885	27.009	9.045	6.033	2.900	10.485	4.406
Ranger	70.289	30.925	40.383	11.405	7.813	2.813	8.954	2.514
Warlock	60.529	28.177	38.859	10.698	7.195	2.595	8.775	4.020
Paladin	63.253	26.523	39.836	10.753	8.183	2.756	7.900	2.772
Sorcerer	62.046	17.881	36.397	9.645	7.614	2.692	8.440	1.835
Cleric	55.831	25.198	40.253	12.299	9.267	3.278	6.029	1.832
Fighter	52.210	21.491	43.019	11.520	9.509	2.979	5.509	1.698
Rogue	58.216	30.286	31.277	9.813	6.712	2.913	8.488	2.830
Wizard	61.507	34.829	26.651	8.559	5.490	2.723	10.661	3.238

Table 7: Summary of class-specific data for Dungeon 3.