D&D Character Exploratory Data Analysis

John Pruitt for GA Python class 127 February 2021

Description of the data set:

Github user oganm, as part of the the project dnddata, shared a large data set of Dungeons and dragons characters. They run a website that prints character sheets and they've published these as a result of that process.

According to the author, the data set is still slightly too small to be fully representative. So there could be selection bias based on the culture of the players who saw this site advertised on reddit. Also some characters are written as a joke or thought experiment so they may not be intended for actual play.

Data dictionary:

A data dictionary is provided by the author, oganm, on the project README file

- **ip:** A shortened hash of the IP address of the submitter
- **finger:** A shortened hash of the browser fingerprint of the submitter
- **name:** A shortened hash of character names
- **race:** Race of the character as coded by the app. May be unclear as the app inconsistently codes race/subrace information. See processedRace
- **background:** Background as it comes out of the application.
- **date:** Time & date of input. Dates before 2018-04-16 are unreliable as some has accidentally changed while moving files around.
- **class:** Class and level. Different classes are separated by | when needed.
- **justClass:** Class without level. Different classes are separated by | when needed.
- **subclass:** Subclass. Might be missing if the character is low level. Different classes are separated by | when needed.
- **level:** Total level
- **feats:** Feats chosen. Mutliple feats are separated by | when needed
- **HP:** Total HP
- **AC:** AC score
- **Str, Dex, Con, Int, Wis, Cha:** Ability score modifiers
- **alignment:** Alignment free text field. Since it's a free text field, it includes alignments written in many forms. See processedAlignment, good and lawful to get the standardized alignment data.
- **skills:** List of proficient skills. Skills are separated by |.
- **weapons:** List of weapons, separated by |. This is a free text field. See processedWeapons for the standardized version

- **spells:** List of spells, separated by |. Each spell has its level next to it separated by *s. This is a free text field. See processedSpells for the standardized version
- **castingStat:** Casting stat as entered by the user. The format allows one casting stat so this is likely wrong if the character has different spellcasting classes. Also every character has a casting stat even if they are not casters due to the data format.
- **choices:** Character building choices. This field information about character properties such as fighting styles and skills chosen for expertise. Different choice types are separated by | when needed. The choice data is written as name of choice followed by a / followed by the choices that are separated by *s
- **country:** The origin of the submitter's IP
- **countryCode:** 2 letter country code
- **processedAlignment:** Standardized version of the alignment column. I have manually matched each non standard spelling of alignment to its correct form. First character represents lawfulness (L, N, C), second one goodness (G,N,E). An empty string means alignment wasn't written or unclear.
- **good, lawful:** Isolated columns for goodness and lawfulness
- **processedRace:** I have gone through the way race column is filled by the app and asigned them to correct races. Also includes some common races that are not natively supported such as warforged and changelings. If empty, indiciates a homebrew race not natively supported by the app.
- **processedSpells:** Formatting is same as spells. Standardized version of the spells column. Spells are matched to an official list using string similarity and some hardcoded rules.
- **processedWeapons:** Formatting is same as weapons. Standardized version of the weapons column. Created like the processedSpells column.
- **levelGroup:** Splits levels into groups. The groups represent the common ASI levels
- **alias:** A friendly alias that correspond to each uniqe name

The list version of this dataset contains all of these fields but they are organised a little differently, keeping fields like spells and processedSpells together.

How and why I chose which variables to use:

The full data set includes some information very specific to an individual character, down to their inventory of gear. Since my goal was to look at survivability choices made early on, a complete list of spells or magic items would not be very illustrative. I chose to look first at how many characters in the set were in each level, and then their classes for high level characters.

Once I had that subset, I thought I could show other decisions made very early in play, subclass and background. They confer some short term benefits (like starting amounts of gold) but also long term benefits that could make a difference over the course of a campaign.

A brief description of any data wrangling and cleaning steps you took:

This data set is very wide and as you can see in the data dictionary there are some “processed” columns with some cleaning and formatting done by the author. The author of the original data set also points out that due to the way the site was set up and maintained, data from before a certain date was not reliable.

There are also some inherent problems with the data set, such as free text fields where custom spells and other character attributes can be entered with misspellings or joke values added making views of the entire set have some odd outliers.

The biggest problem I encountered was that the site was logging character transactions for every print, not the current state of every character. Also Character names were hashed instead of listed verbatim and I found that there was a significant number of hash collisions in this field.

There were even characters with levels far above 20. This is not allowed in the base rules for the game so I also cleaned those out before doing analysis.

Challenges encountered:

Cleaning up the older dates and splitting out outlier values used by only one or two unplayable characters (make as a joke or homebrew) was relatively easy because what I was looking for in my analysis were the most common values for high level characters. Outliers could easily be ignored or would not show in most results.

I tried to comment above these cleaning steps why a certain action was being taken.

The class value for a character was ‘|’ delimited, for example ‘Fighter|Wizard’. Since multiclassing is an advanced technique I was able to map the field to single or multi class, and then make some general points about how multi-class characters compared to single.

The biggest challenge I faced was the problem of identifying unique characters at their highest level. As stated above, there were multiple transactions for each character but name alone had too many collisions to correctly identify an individual. I chose to remove duplicates across ‘name’ and ‘justClass’. That way, I could assume someone with the same name and job is the same person, but increasing levels are progress for the person.

Key insights:

1. The number of level one characters is much higher than expected, showing that many of the characters made see little to no play.
2. The characters that survive to high level play have a very different class distribution than the entire set. So we can conclude that there are aspects of the class choice that improve survivability.
3. Looking at the distribution of subclasses and backgrounds within the wizard class (one of the longest-lived classes in the dataset), we found some leaders in value count.
4. By considering the overrepresentation of these values in the long living set, we can see that they offer better benefits for character survivability.
5. Other aspects like character race confer narrow benefits. They might increase ability scores important to a particular class' abilities, but the humans (who are generally good at all things but master of none) are massively overrepresented at low level play and high level play.

Conclusions:

We could explore more to see which subclass choices lead to survivability, but we have made some interesting conclusions so far. I heard an interview with a designer, Mike Mearls, that there are subclasses and backgrounds for each class designed to be easy to pick up, but not as deep or powerful at high levels. So my major goal in this project was to see which choices are more common for long-lived characters, demonstrating better usefulness across a broad range of situations.