

Business Problem

Can I provide a tool to determine what percentile a RuneScape player is in when they finish collecting all the rare drops from a specific boss? Can this tool be used to plot out a bossing order to maximize duplicated drops to minimize the number of kills needed at certain bosses? Can this tool be used to generate a drop log for a specified number of kills to help players decide what bosses are worth killing for money?

Background/History

RuneScape is a game developed in the late 1990s and still boasts over 100,000 players logged on at any time. As the game continues to age, many players turn to completing the in-game collection log as a reason to continue playing, or as a vector check on what content to complete to generate income for their characters. Many creators on Twitch and YouTube are basing entire series on how fast they can complete the collection log. The community has built a passion around “green logging” every boss they enjoy. The named piece of content in the in-game collection log turns green when every item has been collected.



General Graardor
Giant Mole
Grotesque Guardians
Hespori
The Inferno
Kalphite Queen
King Black Dragon
Kraken
Kree'arra
K'ril Tsutseroth
Nex
The Nightmare
Obor

Data Explanation

Using the Drop Generator library, I was able to assign probabilities to drops that reflect the confirmed drop rates from the developers of RuneScape; Jagex. With this library, I was able to loop through the drop table as many times as necessary to fill up Data Frames that held a list of every drop that the boss possessed. I recorded the boss's name, average kills until the log was completed, the fastest a log was completed as well as the slowest a log was completed.

Boss Name – The name of the boss. One raid uses a difficulty system called “Invocation” which is used to delineate the level of the raid, as well as an increased potential of rare items.

Average Kills – After 10,000 logs are completed, this number is the average amount of kills you can expect to complete the collection log.

Lowest Kills – After 10,000 logs are completed, this was the fastest log completion in the simulation. These are typically outliers in the dataset.

Highest Kills – After 10,000 logs are completed, this was the slowest log completion in the simulation. These are typically outliers in the dataset.

After completing a log, the counter would reset and run the simulation again until all 10,000 simulations were run and recorded. After the final simulation, the important data was appended to a complete data frame that housed a collection of all the boss' information.

Methods

I created a histogram for each variable to see if there were any noticeable distributions. From there, I noticed that all but one boss was positively skewed. The only boss that had no skew, was "Tombs of Amascut – 500 Invocation". I primarily used the Drop Generator library to create drop tables to loop over. one size fits all approach was not feasible, so large-scale functions were not truly an option. Each raid has milestone items that can only be obtained at certain kill thresholds, and as such needed specific checks for the kill variable. Some bosses have minions that drop rare items as well, so including them into the calculation was also pertinent. This was a thought exercise on Python, as the Tombs of Amascut has 4 drops that can only be dropped once, and having to account for those checks throughout each simulation was needed.

For each piece of content, I created a data frame that would be used to track the progress of the log throughout the simulations. Whenever a unique item was identified in a kill's drop list, the data frame would update to increase the Amount Dropped column by 1. At the end of the tally, a check would occur to see if the minimum value in that column was at least 1. If so, that simulation was complete.

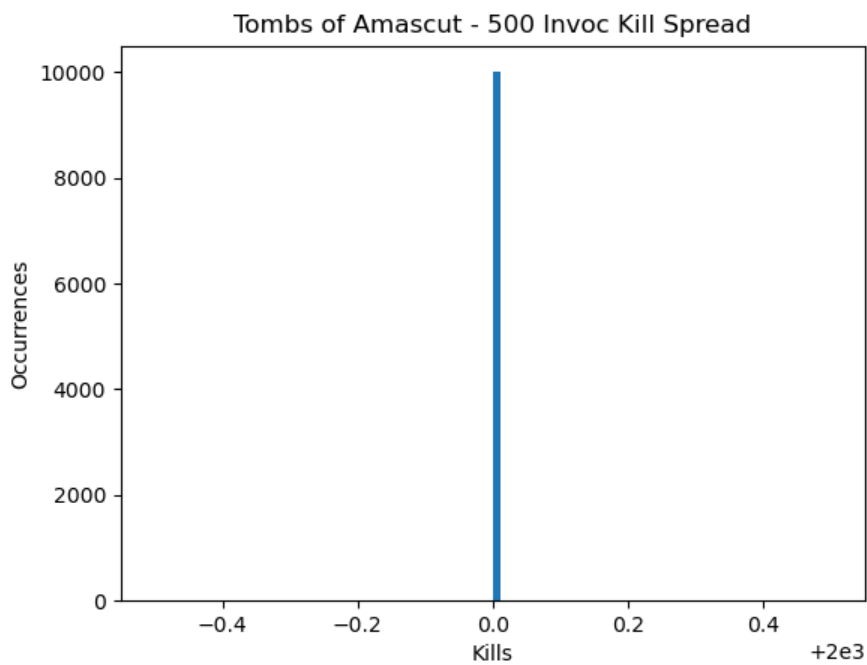
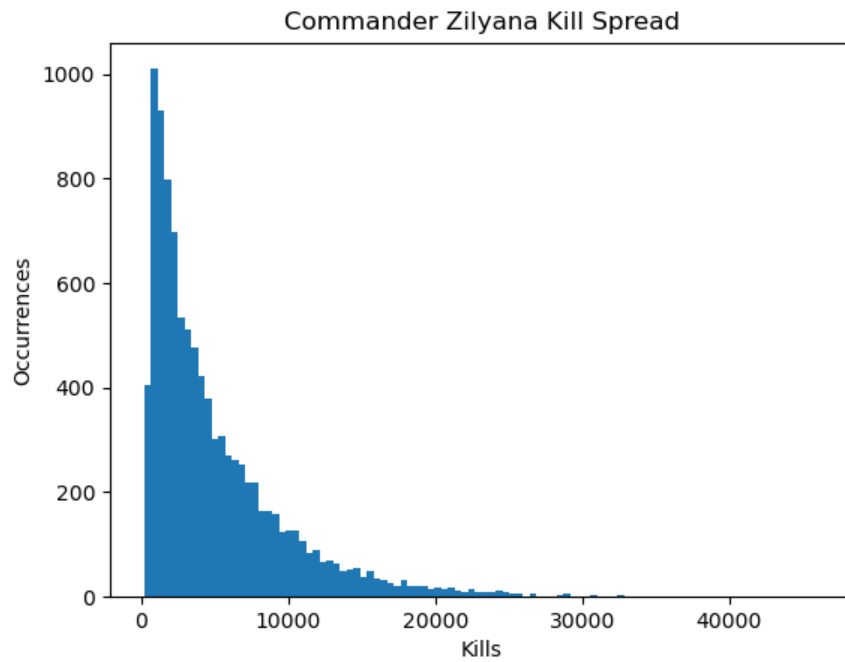
	Items	Amount Dropped
0	Olmlet	0
1	Metamorphic Dust	0
2	Twisted bow	0
3	Elder maul	0
4	Kodai insignia	0
5	Dragon claws	0
6	Ancestral hat	0
7	Ancestral robe top	0
8	Ancestral robe bottom	0
9	Dinh's bulwark	0
10	Dexterous prayer scroll	0
11	Arcane prayer scroll	0
12	Dragon hunter crossbow	0
13	Twisted buckler	0
14	Torn prayer scroll	0
15	Dark relic	0
16	Onyx	0
17	Twisted ancestral colour kit	0
18	Xeric's guard	0
19	Xeric's warrior	0
20	Xeric's sentinel	0
21	Xeric's general	0
22	Xeric's champion	0

Analysis

After creating the histograms, I noticed that the average kills of most bosses fell within +/- 5% of their most unique drop.

	Boss Name	Average Kills	Lowest Kills	Highest Kills
0	King Black Dragon	6200.8247	122	53843
1	Chambers of Xeric: Challenge Mode	2112.1830	2000	6640
2	Hard Mode Theatre of Blood	2025.6928	2000	5744
3	Tombs of Amascut - 150 Invocation	2270.0679	2000	12702
4	Tombs of Amascut - 300 Invocation	2015.6065	2000	4797
5	Tombs of Amascut - 500 Invocation	2000.0000	2000	2000
6	General Graardor	5144.4013	198	52068
7	K'rill Tsutsaroth	5090.7788	156	38644
8	Kree'arra	4980.1719	197	46531
9	Commander Zilyana	5118.2934	204	46175

Using Matplotlib, I was able to graph the number of kills simulations to see the distribution curves. Most bosses held a strong positive skew. This was mostly due to only needing 1 of each item for log completion. If they got lucky on the rarer items, this would heavily push that simulation to a lower kill number. Raids on the other hand, had a minimum of 2,000 kills due to the longest milestone being the 2,000th kill. Interestingly, the Tombs of Amascut – 500 Invocation simulations all ended at 2,000 kills due to their drastically higher drop rates than the other invocation levels.



I also created a means to build a data frame of all drops from a boss, to show the expected loot for all drops when going for a completed log at any boss.

Conclusion

Statistically, raids held a much lower standard deviation due to their kill floors. On the other hand, bosses like the King Black Dragon had a standard deviation of over 4,800 kills due to the outliers with extremely poor luck. The means for non-raid bosses ended up within the 60-65th percentile due to the wide variations. These simulations are an excellent way to plan out a specific bossing route to take advantage of duplicated drops to minimize time when going for log completion.

Assumptions

The biggest assumption with the current simulation is that players will continue to do the raid content until they receive all milestone items. As they are only cosmetics, they provide no value outside of the color of the boss log turning from orange to green.

Limitations

Currently, these simulations do not reference other logs. If this functionality were added, a proper analysis and bossing order could be derived.

Challenges

The biggest challenge was figuring out the appropriate data size. Creating 10,000 completed logs provided a good balance between the time needed to generate the data and comprehensive results. The other challenge is, there are so many bosses in RuneScape, that a full collection will take time to generate fully.

Future Uses

These simulations will be used in future YouTube projects for outlining the progression of a 5-person group chasing the whole collection log 5 times between each other.

Recommendations

It is strongly recommended to flesh out the rest of the boss content, and then add simulations for non-boss-related collection logs. Separate the functionality for the simulated drops to provide tool for everyone to plan out resource gathering through killing bosses.

Implementation Plan

This program will be used in conjunction with script writing to plan out levelling routes and kill orders to minimize the time spent between each piece of content.

Ethical Assessment

Ethically, the biggest concern is the emotional impact on being unluckier than the average player. This can lead people to being burnt out, as there are no bad luck mitigators. Seeing the highest kill count for a completed log can also turn people off from bothering to try in the first place. With the King Black Dragon's average of 6,200 kills, having somebody play a video game for that long could negatively impact their health. Promoting healthy gaming habits will be key before promoting this project.

Appendix

Kumar, S. (2021, September 21). Here is the most efficient way to iterate through your pandas data frame. Medium. Retrieved April 30, 2023, from <https://towardsdatascience.com/heres-the-most-efficient-way-to-iterate-through-your-pandas-dataframe-4dad88ac92ee>

Ramencatz / python-loot · GITLAB. GitLab. (n.d.). Retrieved April 30, 2023, from <https://gitlab.com/ramencatz/python-loot>

Santos, B. (2020, March 10). Lord of the notebooks: Optimizing Jupyter. Medium. Retrieved April 30, 2023, from <https://medium.com/@bryan.santos/lord-of-the-notebooks-optimizing-jupyter-9cc168debcc7>