




DOTA 2 MATCH DATA METHODS DOCUMENT



Jacob Sorensen
219 Meadville St, Edinboro, PA 16444

Contents

Introduction	1
The Dataset	1
Main Data Files	2
Data Dictionary	3
Obtaining the Raw Data sheets	4
Sorting and Cleaning	4
Data Analysis	4
GG analysis.....	5
Feed analysis.....	5
Conclusions	6
Bibliography	10

Introduction

The purpose of this project is to explore how chat messages in the video game Dota 2 relate to average match length, as well as see what the time distribution of certain chat messages are compared to average match length.

The Dataset

The dataset was obtained from Kaggle.com (Anzelmo, 2019). It includes match data from 50,000 unique Dota 2 matches. It was derived from the Dota 2 data dump created by Opendota, a website that obtains large volumes of data from Dota 2. The whole Dataset totals about 1 GB in size.

Dota 2 is a Multiplayer Online Battle Arena in which there are 2 teams, each with 5 players. The name of the two teams is Radiant and Dire. The main objective of this game is to destroy the other teams "ancient", a building located at the center of each team's base. Towers and barracks are considered important objectives, as your team is rewarded when the enemy's buildings are destroyed. Towers are buildings that must be destroyed in order to reach the ancient. There are 3 lanes, each with 3 sets of towers, and there is a fourth set of towers directly in front of the ancient. There are also buildings called barracks which, when the enemy team's barracks are destroyed, makes your teams "creeps" more powerful. Creeps are units that are sent out frequently in each lane from each team, which fight enemy creeps, buildings or players.

There are 12 excel files included in this dataset, each with their own types of data. The Dataset totals about 1 GB in size. The two files titled "match" and "chat" are the ones likely to be examined in this project in order to see the relationship between certain chat messages, match length and game outcome. The file "chat" has 5 columns and 1,439,489 rows, while the file "match" has 13 columns and 50,001 rows.

Main Data Files

These file descriptions are taken from Kaggle (Anzelmo, 2019).

Main Data Files	
matches	contains top level information about each match.
players	Individual players are identified by account_id but there is an option to play anonymously and roughly one third of the account_id are not available. Anonymous users have the value of 0 for account_id. Contains totals for kills, deaths, denies, etc. Player action counts are available, and are indicated by variable names beginning with unit_order_. Counts for reasons for acquiring or losing gold, and gaining experience, have prefixes gold_, and xp_.
player_time	Contains last hits, experience, and gold sampled at one minute interval for all players in all matches. The column names indicate the player_slot. For instance xp_t_1 indicates that this column has experience sums for the player in slot one.
teamfights	Start and stop time of teamfights, as well as last death time. Teamfights appear to be all battles with three or more deaths. As such this does not include all battles for the entire match.
teamfights_players	Additional information provided for each player in each teamfight. player_slot can be used to link this back to players.csv
objectives	Gives information on all the objectives completed, by which player and at what time.
chat	All chat for the 50k matches. There is plenty of profanity, and good natured trolling.
test_labels	matchid and radiantwin(as integer 1 or 0)
test_player	full player and match table with heroid, playerslot, matchid, and accountid
player_ratings	contains match counts, win counts, and TrueSkill rating, calculated on 900k matches which occurred prior to other uploaded data. trueskill ratings have two components, mu, which can be interpreted as the skill, with higher value being better, and sigma which is the uncertainty of the rating.
match_outcomes	data for ~900k matches used to calculate player ratings.
purchase_log	item purchase times
ability_upgrade	ability upgrade times and levels
cluster_region	allows the mapping cluster found in match.csv to geographic region.
patch_dates	release dates for various patches, use start_time from match.csv to determine which patch a match was played in.
ability_ids	use with ability_upgrades.csv to get the names of upgraded abilities
item_ids	use with purchase_log.csv to get the names of purchased items

Data Dictionary

Data dictionary	
Chat	
match_id	The ID number of the specific match in question, this will be different for every match.
key	The chat messages
slot	Which slot the player who said the chat message was in. 0-4 is someone from the radiant side and 5-9 is someone from the dire side.
time	The time in the game on which the chat message was said. This is in seconds in relation to time 0, which is about 2 minutes after players first enter the match.
unit	The username of the person who sent the chat message
Match	
match_id	The ID number of the specific match in question, this will be different for every match.
start_time	The time at which the match started
duration	how long the match went on for
tower_status_radiant	the status of the towers on the radiant side
tower_status_dire	the status of the towers on the dire side
barracks_status_dire	the status of the barracks on the radiant side
barracks_status_radiant	the status of the barracks on the dire side
first_blood_time	The time at which "first blood" happened, this is the first time a player was killed by the enemy team.
game_mode	the game mode of the match in question, this should be the same for every match
radiant_win	if the radiant team won, True if yes, False if no.
negative_votes	how many post-game negative votes the game has
positive_votes	how many post-game positive votes the game has
cluster	unknown

Obtaining the Raw Data sheets

In order to analyze the chat.csv file in excel it needed to be split, the way I did this was by opening it in notepad++ and splitting it into two csv files, the first with the first 25,000 matches and the second with the second 25,000 matches. The match.csv file was used to find the game duration for all 50,000 matches, this was placed into the raw general match data sheet. These were imported into excel by opening them with excel and dragging the sheets into one excel document. The first 3 sheets after the about page are those 3 raw data files.

Sorting and Cleaning

The next two sheets, Chat1 sorting and Chat2 sorting are the same as the raw chat files, but they have been sorted by the chat messages in alphabetical order. The only cleaning that was done was in the general match data sorting sheet, and an outlier match was removed. This match, match ID 9946, had a duration of 16037 seconds, surpassing the next longest game by almost 10000 seconds. When analyzing the chat messages for this match I noticed they were mostly nonsensical, and only started appearing after 5000 seconds into the match. This match was therefore removed in order to maintain legible histograms.

Data Analysis

Two message types were analyzed, messages containing “gg” that were related to the phrase “good game” and messages containing “feed”. Both message analyses were placed into the “Analysis of Messages” sheet.

GG analysis

The “gg related messages...” column was created by sorting the chat1 and chat2 columns by alphabetical order and extracting most GG related messages. Words that had gg inside of them (e.g. flogging) were not included in this. The sort and filter tool was used to obtain some categories like “gg”, “ggwp” and “gege”. After these were extracted, all messages were placed into the analysis of messages into the same column, in order to recombine the previously separated chat columns. After the “GG related messages...” column was created, the “gg occurrences per game” column could be created. This was made by taking all the match ID’s from the gg related messages column and removing all duplicates. The number of “gg” occurrences per game was then produced by counting the number of times each match ID appeared in the gg related messages column. The game duration was produced by looking up the match ID in the “general match data” sheet. Graphs using this data were then produced, and a histogram of gg related messages was overlaid with a histogram of average game length. The x-axis for these histograms is the time of either the games or the chat messages, separated into 60 second chunks. A scatterplot of gg message occurrences was also produced.

Feed analysis

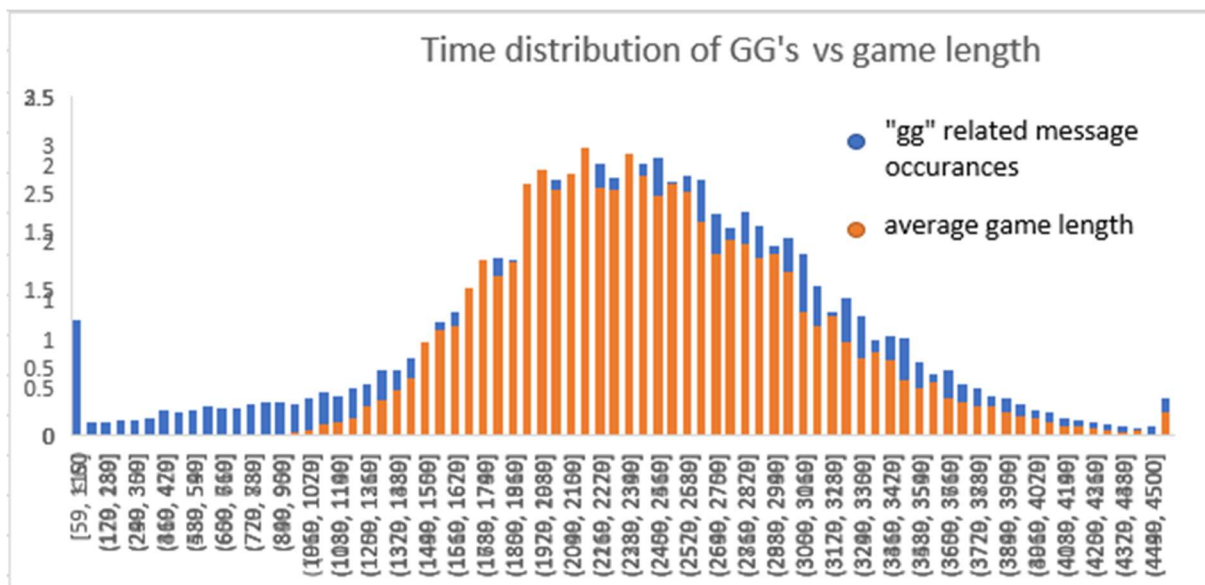
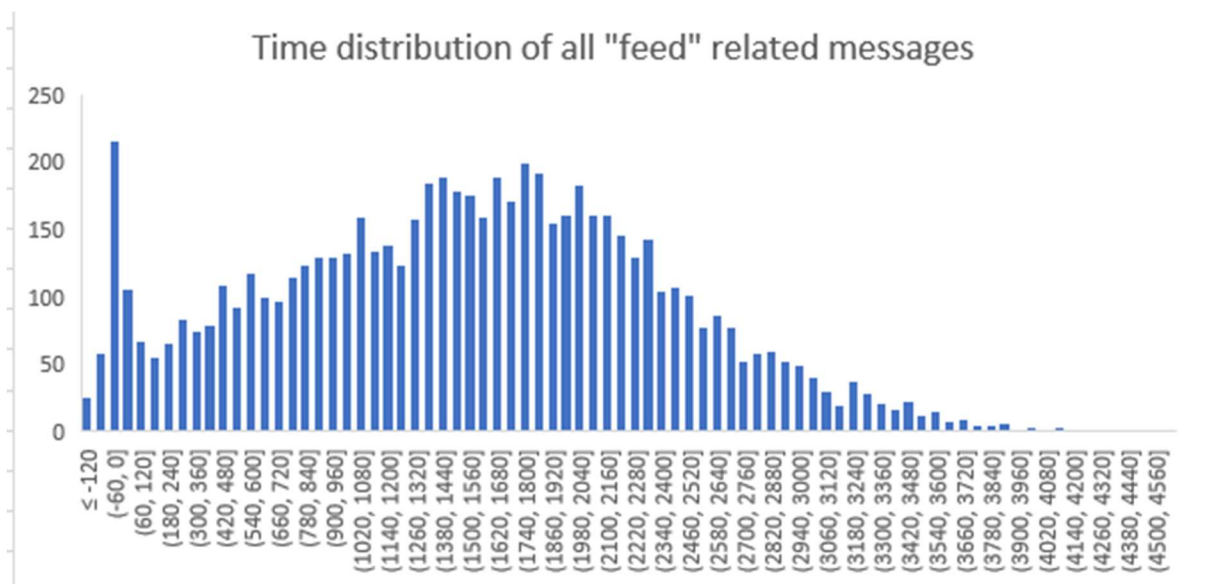
The “feed” related message column was produced in a similar manner to the “gg” related message column, except it was made using the filter tool, to filter by the string “feed”. So any message that contained “feed” anywhere in it, including words like “feeding” were included in this column. Like the gg column, it includes messages from all 50,000 games, so the chat1 and chat2 sheets were recombined after the “feed” messages were extracted. The feed occurrences per game column was produced in the exact same manner as the gg occurrences per game column. Some graphs were produced using this data. A histogram of the duration of games that contain feed related messages was

plotted against the histogram of average game length. A scatterplot of feed occurrences was also produced.

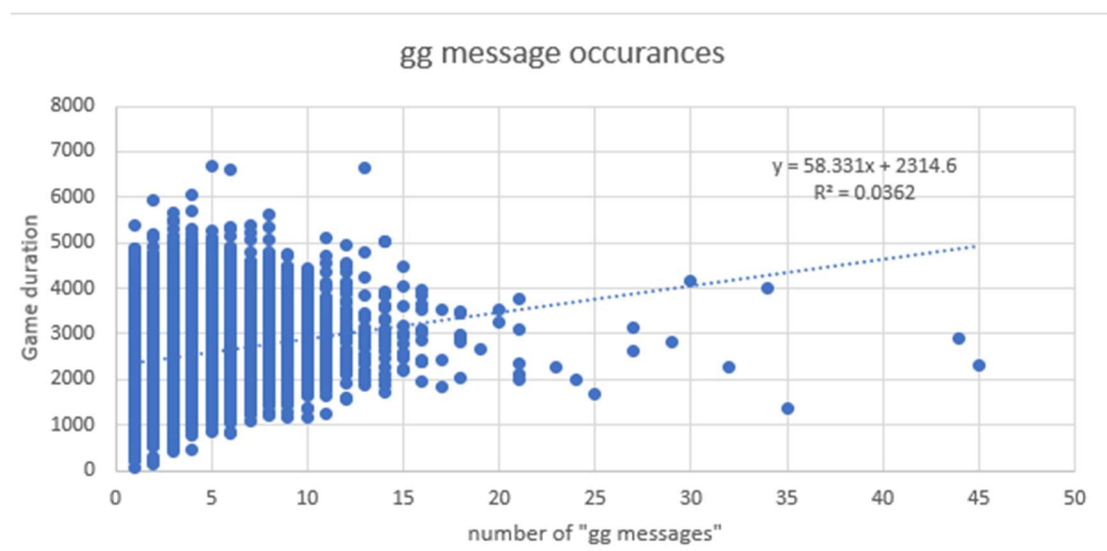
Conclusions

There are several interesting relationships that were produced from the analysis of the chat messages.

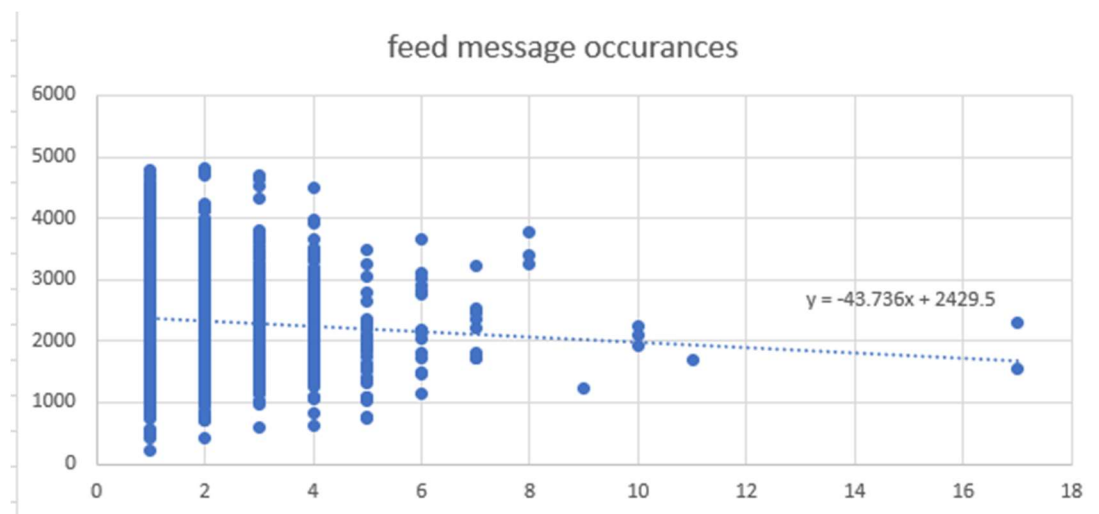
All histograms have time as the x value, and the y-value isn't as important with them because we are comparing the relative distributions and not the specific count numbers.



The distribution of GG messages seemed to be much closer to the game duration distribution than the feed distribution. This also gives a trend of more gg messages appearing later, with later games.



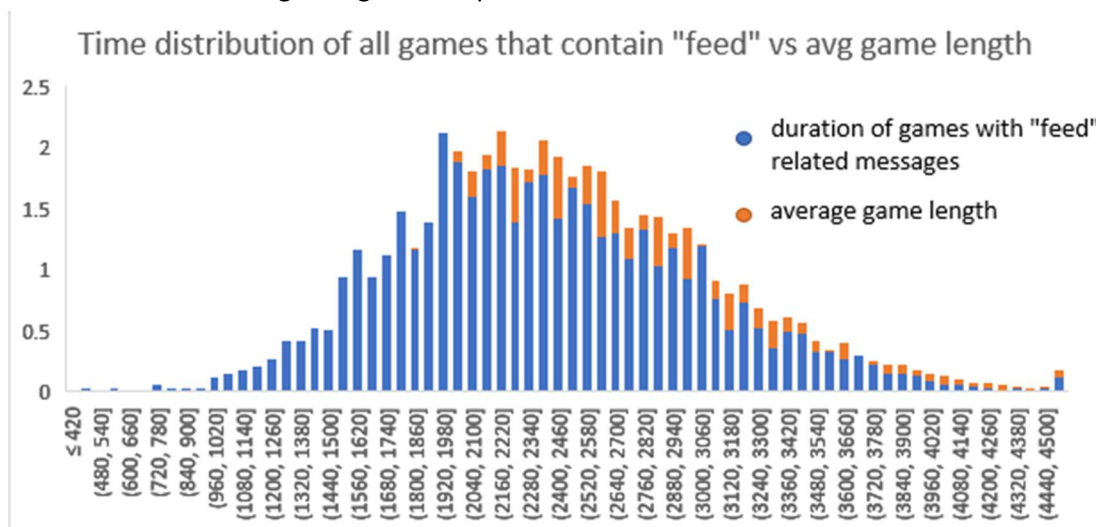
The GG message occurrences scatterplot gave a trendline, and while it wasn't a linear relationship, it showed a trend where for about every 58 seconds of in-game time that passed another "gg" related message appeared.



The feed message occurrences scatterplot trendline wasn't a linear relationship either, but it showed a trend where for every feed message the game length duration decreased by about 44 seconds.

# data points			
All messages	1439488		
"feed" messages	6701		
"GG" messages	140757		
All games	50000		
"feed" games	4593		
"gg" games	43950		
length	all games	"feed" game	"GG" game
average	2476	2366	2501
difference from all		-110	25
Quartiles			
longest duration	6673	4805	6673
Q4	2872	2788	2896
Mode	2415	2313	2439
Q2	2029	1924	2048
Shortest duration	59	214	59

Finally, the average of all games was taken and compared with the average of all “gg” games, and all “feed” games. The average game containing feed messages was about 110 seconds shorter than the average game, and the average game containing gg messages was about 25 seconds longer than the average game. The relationship between games with feed messaged and average game length can also be seen in the following histogram comparison.



Future work

I would have liked to analyze where gg messages came up in relation to that games end and see a distribution of the difference there compared to the game length. I did not have time, nor do I have the expertise to do this exactly.

Bibliography

Anzelmo, D. (2019, November 14). *Dota 2 Matches*. Retrieved from kaggle:

<https://www.kaggle.com/devinanzelmo/dota-2-matches>

Opendota. (n.d.). Retrieved from opendota: <https://www.opendota.com/>