

# IDV Learners Capstone Project - Board Game Analysis

JHANSI

2023-03-12

## Executive Summary

BoardGameGeek (<https://boardgamegeek.com>) is a website that collects user reviews of boardgames in order to rank them from best to worst. The purpose of this analysis is to determine if there is any pattern in the data collected by BoardGameGeek and if there are some attributes of different games that can predict how popular or well recieved the game is by the community. The data set used for this project contains approximately 5,000 records. Each record represents a single board game and includes the following information:

- Rank from most highly to least highly rated
- Game ID
- Name of the game
- Minimum and maximum number of players
- Minimum, maximum, and average play time
- Year of release
- Average rating of the user community
- Geek rating, which is a weighted average to normalize games with few ratings
- Number of times the game has been rated
- Minimum recommended age for players
- Game mechanics
- Number of people who have reported buying the game
- Category of the game
- Designer of the game
- Weight, which is the community's assessment of how complex the game is from simple (1) to complex (5)

To perform this analysis the dataset has been split into an approximately 85% / 15% partition with the larger set used to train the models and the smaller set used to validate the performance of the models. A linear model attempts to predict the average rating of a game based on how complex it is as there appears to be a preference for more complex games in this community. Next, a decision tree is used to attempt to predict whether or not a game will be among the top 10% most highly rated games based on features such as the game mechanics and category of the game.

```
## Warning: package 'corrplot' was built under R version 4.2.3
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3
```

```
## Warning: package 'caret' was built under R version 4.2.3
```

## Methods and Analysis

First we will check the data for missing values and also examine some of the variables to see if any cleaning needs to be done.

```
## [1] 0
```

```
##      rank      bgg_url      game_id      names min_players max_players
##        0          0          0         28        4991        4969
##   avg_time  min_time  max_time      year  avg_rating geek_rating
##    4926      4937      4926     4895         70         157
##  num_votes  image_url      age  mechanic      owned      category
##    2633          0     4979     2685     2030        2741
##   designer      weight
##    2536        2025
```

```
## [1] "Lord of the Rings: The Confrontation"
## [2] "Citadels"
## [3] "Cosmic Encounter"
## [4] "Axis & Allies"
## [5] "Cosmic Encounter"
## [6] "Tales of the Arabian Nights"
## [7] "Fortress America"
## [8] "Cosmic Encounter"
## [9] "Cry Havoc"
## [10] "Fresh Fish"
## [11] "Barbarossa"
## [12] "Wizard"
## [13] "Warrior Knights"
## [14] "Santorini"
## [15] "Mogul"
## [16] "Samurai"
## [17] "Conquest of the Empire"
## [18] "Elfenroads"
## [19] "Blackbeard"
## [20] "Arkham Horror"
## [21] "Hellas"
## [22] "Founding Fathers"
## [23] "Around the World in 80 Days"
## [24] "Samurai"
## [25] "Cartagena"
## [26] "Crimson Skies"
## [27] "Aladdin's Dragons"
## [28] "Saga"
```

```
##      game_id      names year
## 1       492  Aladdin's Dragons 2000
## 2     53103  Aladdin's Dragons 2009
## 3     15987   Arkham Horror 2005
## 4        34   Arkham Horror 1987
## 5     12005  Around the World in 80 Days 2004
## 6    204599  Around the World in 80 Days 2016
## 7     10093    Axis & Allies 2004
```

## 8	98	Axis & Allies	1981
## 9	550	Barbarossa	1988
## 10	72809	Barbarossa	2010
## 11	235	Blackbeard	1991
## 12	25685	Blackbeard	2008
## 13	826	Cartagena	2000
## 14	224031	Cartagena	2017
## 15	478	Citadels	2000
## 16	205398	Citadels	2016
## 17	17710	Conquest of the Empire	2005
## 18	97	Conquest of the Empire	1984
## 19	39463	Cosmic Encounter	2008
## 20	15	Cosmic Encounter	1977
## 21	40529	Cosmic Encounter	1991
## 22	40531	Cosmic Encounter	2000
## 23	3855	Crimson Skies	1998
## 24	6551	Crimson Skies	2003
## 25	192457	Cry Havoc	2016
## 26	1323	Cry Havoc	1981
## 27	180325	Elfenroads	2015
## 28	711	Elfenroads	1992
## 29	99	Fortress America	1986
## 30	115293	Fortress America	2012
## 31	37358	Founding Fathers	2010
## 32	35423	Founding Fathers	2007
## 33	1017	Fresh Fish	1997
## 34	164698	Fresh Fish	2014
## 35	4529	Hellas	2002
## 36	207330	Hellas	2016
## 37	18833	Lord of the Rings: The Confrontation	2005
## 38	3201	Lord of the Rings: The Confrontation	2002
## 39	174744	Mogul	2015
## 40	4562	Mogul	2002
## 41	101865	Saga	2011
## 42	9202	Saga	2004
## 43	3	Samurai	1998
## 44	3061	Samurai	1996
## 45	63	Samurai	1979
## 46	194655	Santorini	2016
## 47	9963	Santorini	2004
## 48	34119	Tales of the Arabian Nights	2009
## 49	788	Tales of the Arabian Nights	1985
## 50	22038	Warrior Knights	2006
## 51	1143	Warrior Knights	1985
## 52	1465	Wizard	1984
## 53	3463	Wizard	1978

```
## Warning in set.seed(5, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

At first it appears that some games have duplicate records, however upon closer inspection it is apparent that these correspond to the same game that was revised and re-released at a later date, so these records will be retained as unique game versions.

Next, we will examine a correlation matrix to see if there are any obvious relationships between some of these variables. Note, that for this plot only numeric variables can be examined:

```
## Warning: Using an external vector in selections was deprecated in tidysselect 1.1.0.
## i Please use 'all_of()' or 'any_of()' instead.
## # Was:
## data %>% select(numeric_variables)
##
## # Now:
## data %>% select(all_of(numeric_variables))
##
## See <https://tidysselect.r-lib.org/reference/faq-external-vector.html>.
```

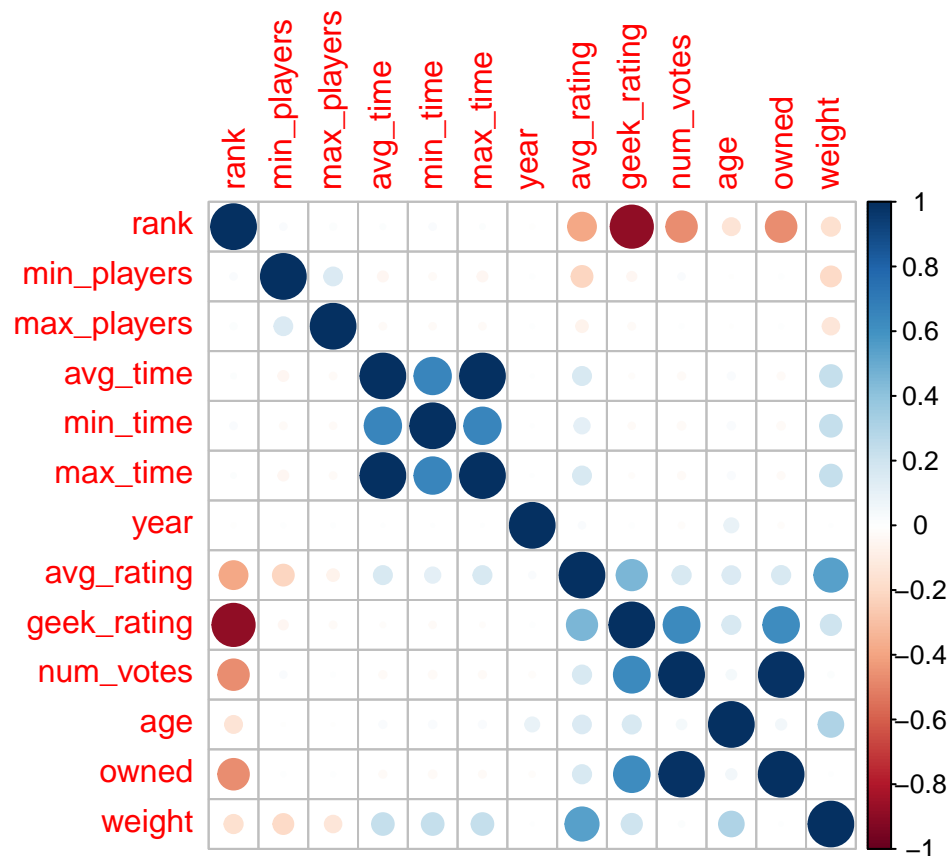
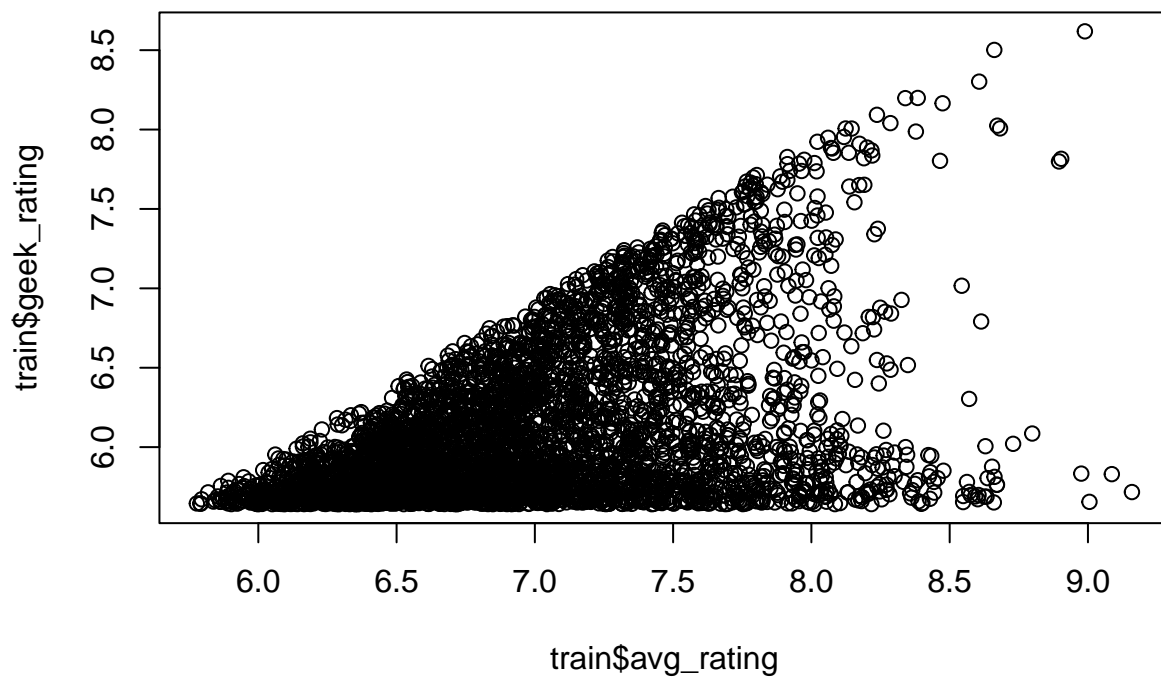


Figure 1: Figure 1: Correlation Matrix

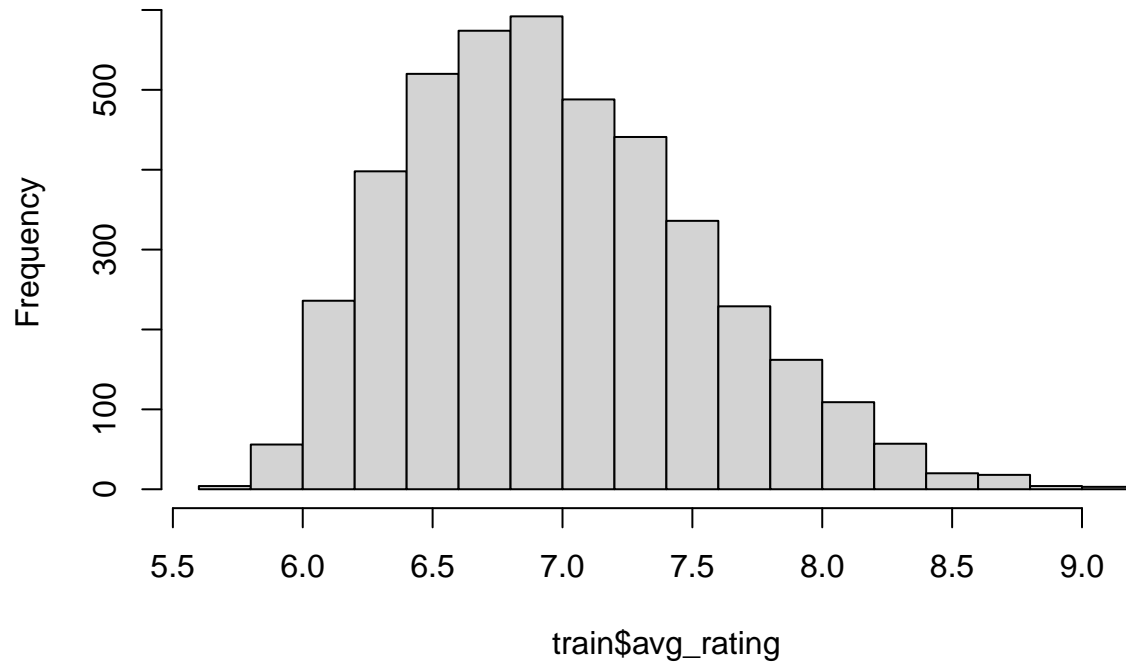
This plot demonstrates that there are some relationships between some of the variable; unfortunately, some of these correlations are not very useful. For example, average game time is correlated with both minimum and maximum time, which is a trivial observation. It is worth noting that how many people report owning a given game and the number of times that the game is rated does correlate with a more highly ranked game. The most interesting feature here is that it appears that the weight of the game (complexity) appears to correlate with good ratings.

Next we look at the average rating versus the “geek” rating. As we see from the following plot, there is a capping effect built into the “geek” rating such that a game with a small number of very high ratings does not overwhelm the overall ranking system.

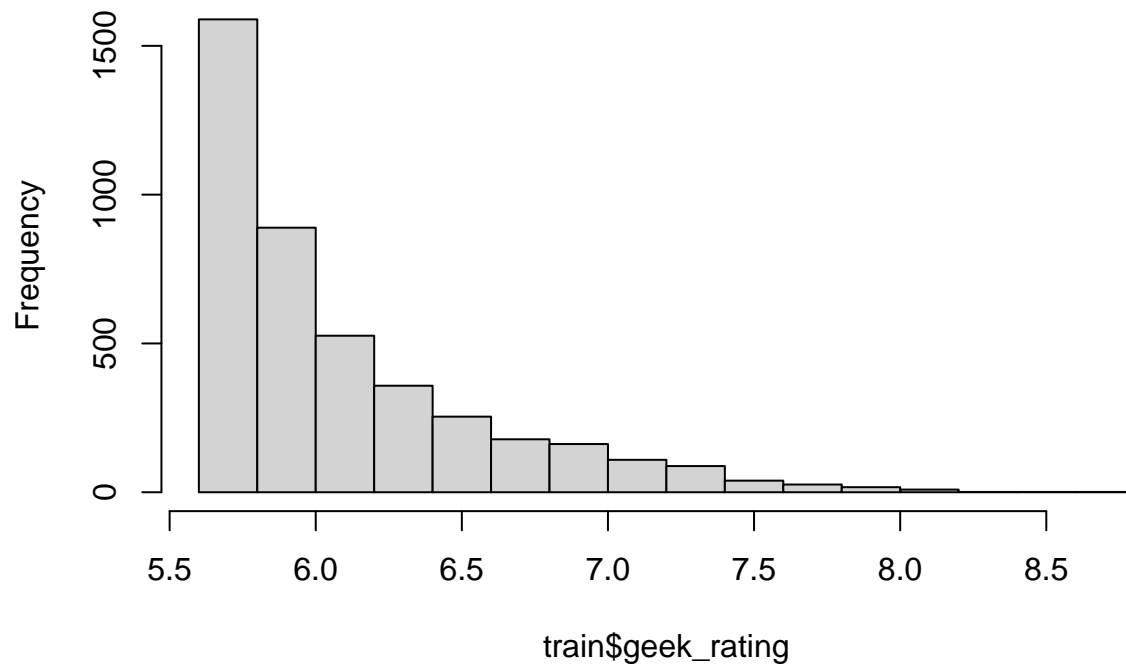


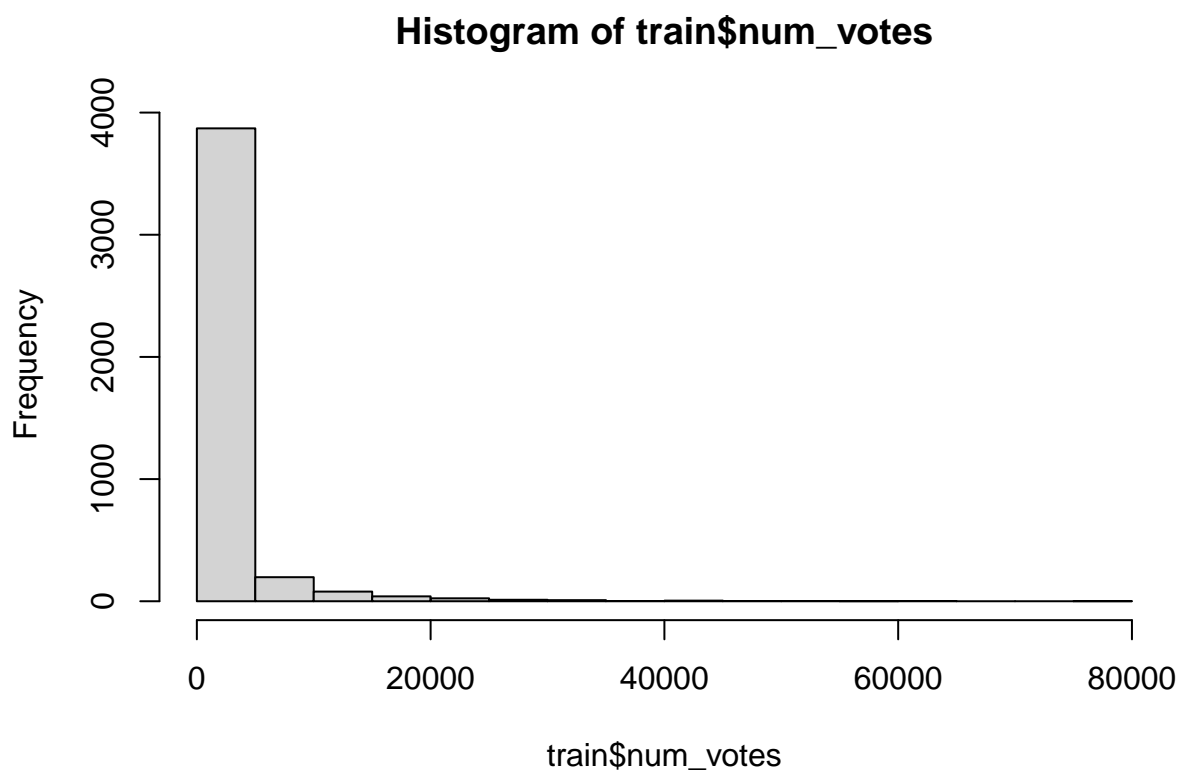
This capping effect can also be visualized by comparing the next three histograms where we see that the average ratings are similar to a normal distribution whereas the “geek” ratings are skewed toward the bottom. This is because games with few ratings are normalized toward a lower rating so that the website will not rank a game very highly if it has not been reviewed by enough people.

**Histogram of train\$avg\_rating**



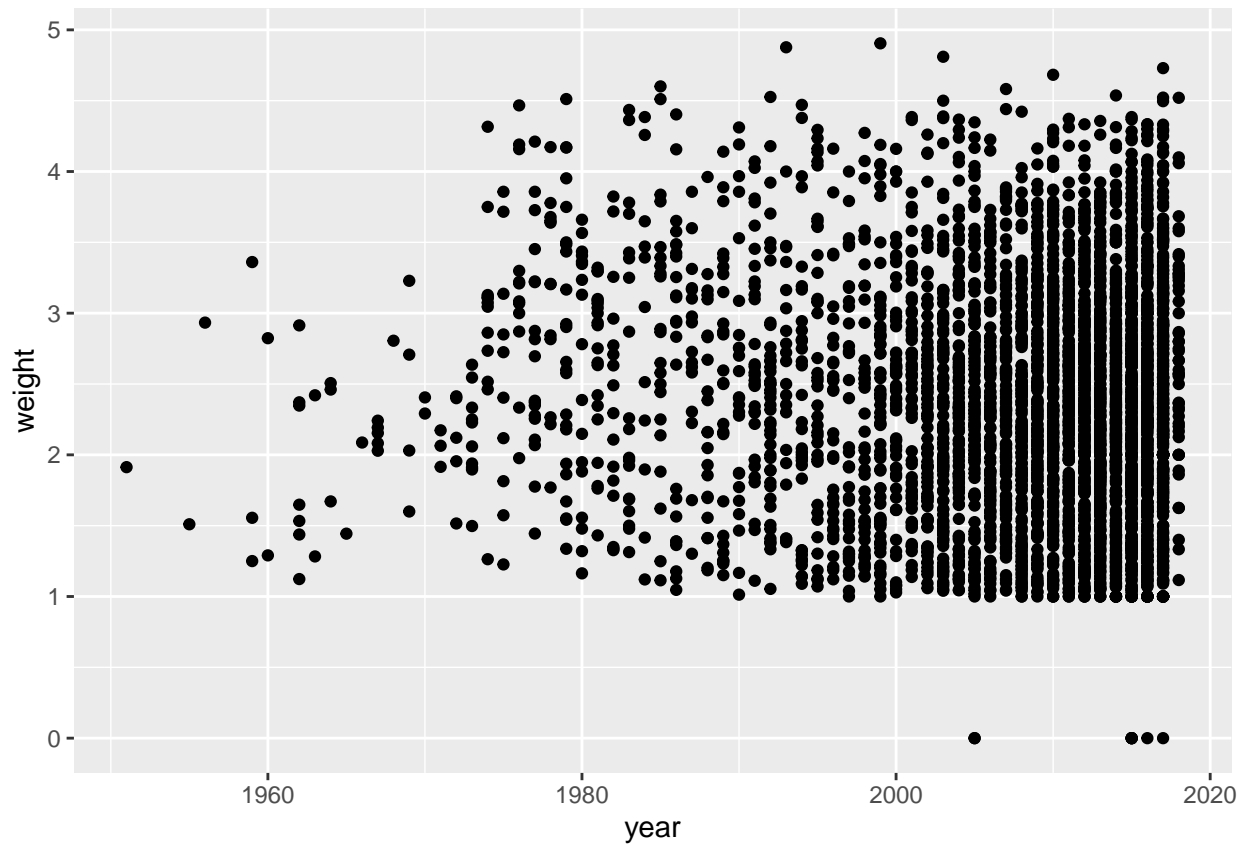
**Histogram of train\$geek\_rating**



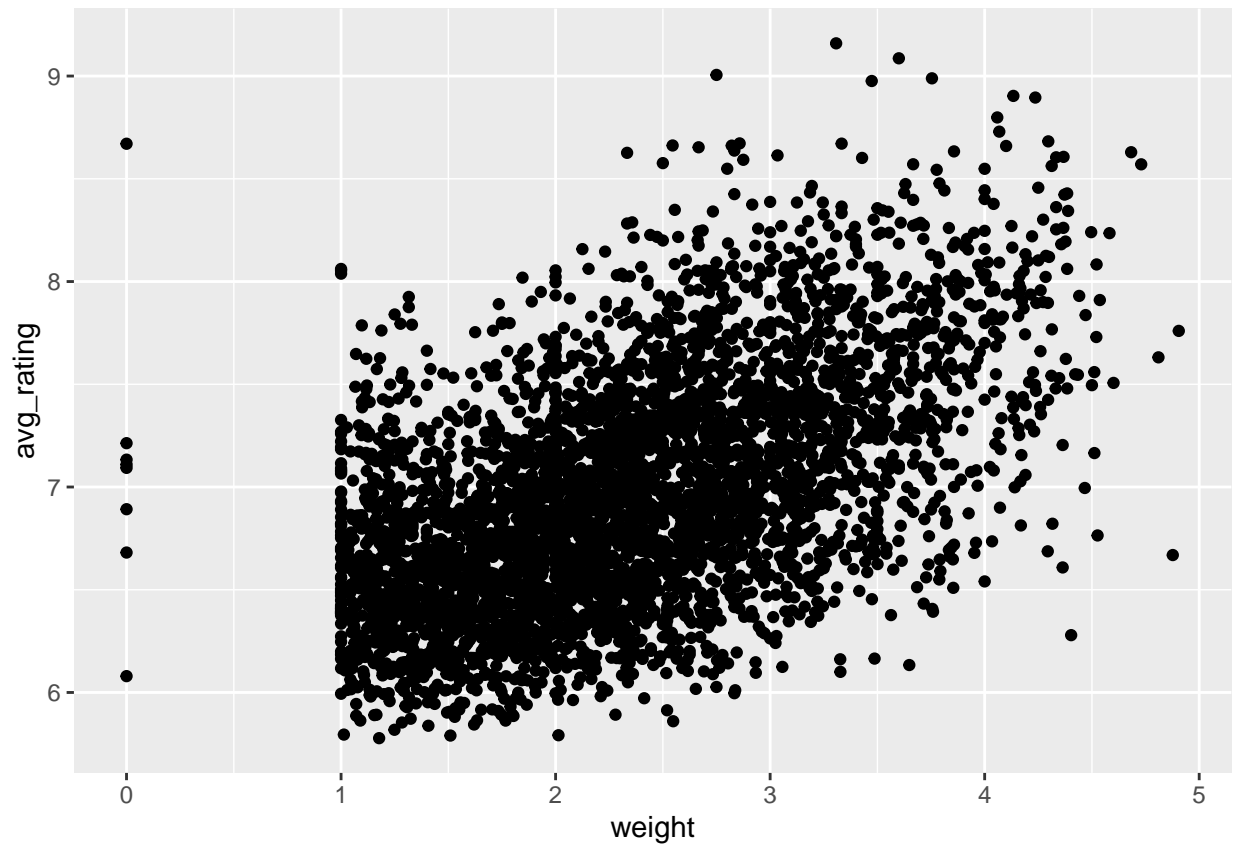


Next we look at the complexity of the game by the year of release. There might be a small effect here where newer games are more complex, but this relationship cannot be firmly established because there are not enough games released before the year 2000 to make a strong correlatioon possible.

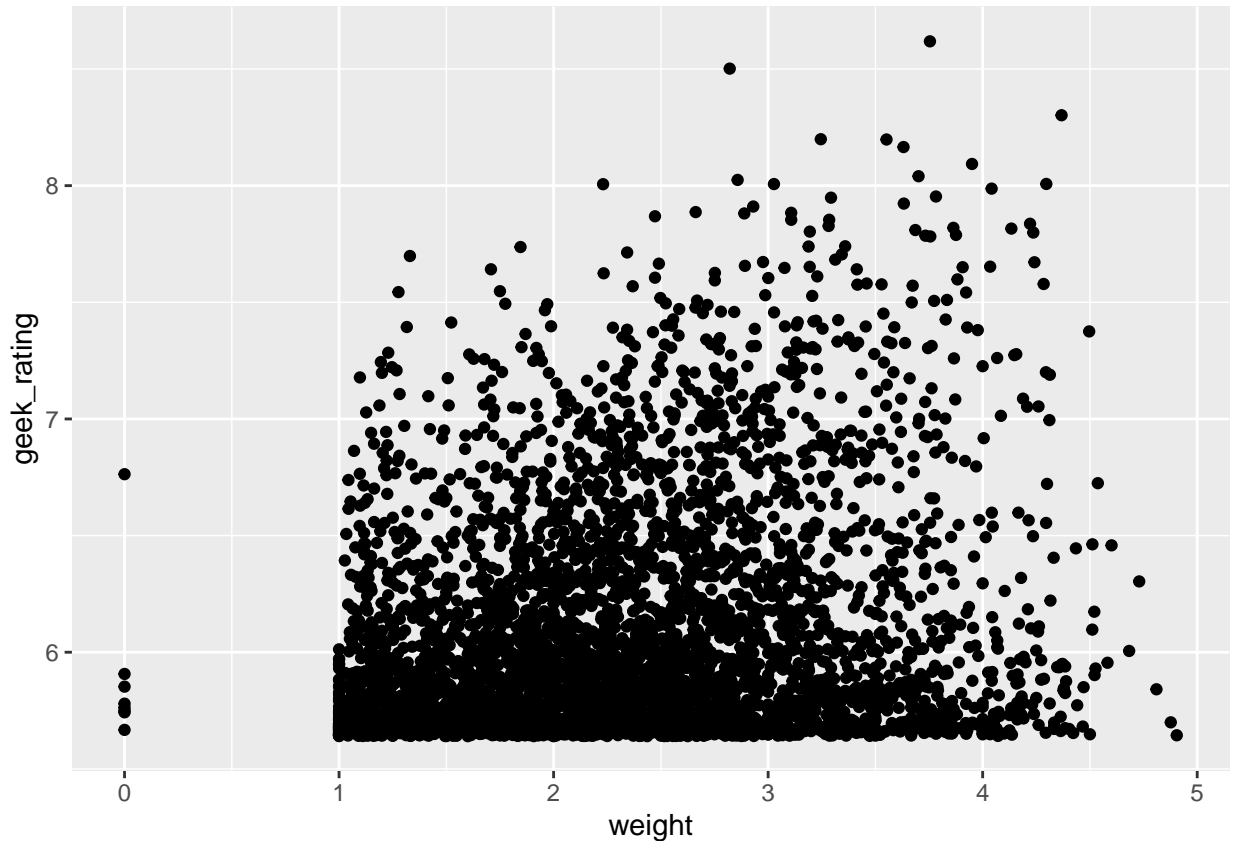




The next plot shows a promising and seemingly linear relationship between the weight (or complexity) of the game is positively correlated to the community's favorable ratings.



The next plot examines the “geek” rating versus the weight of the games. Here the relationship is less strong, again most likely due to the capping system that BoardGameGeek uses.



Using this apparent relationship between weight and the community's ratings we will build the following linear model and compare its predictive power to that of simply assigning the mean rating to all games.

```
## (Intercept)    weight
##  6.0716725    0.3826026

## [1] 0.3115047

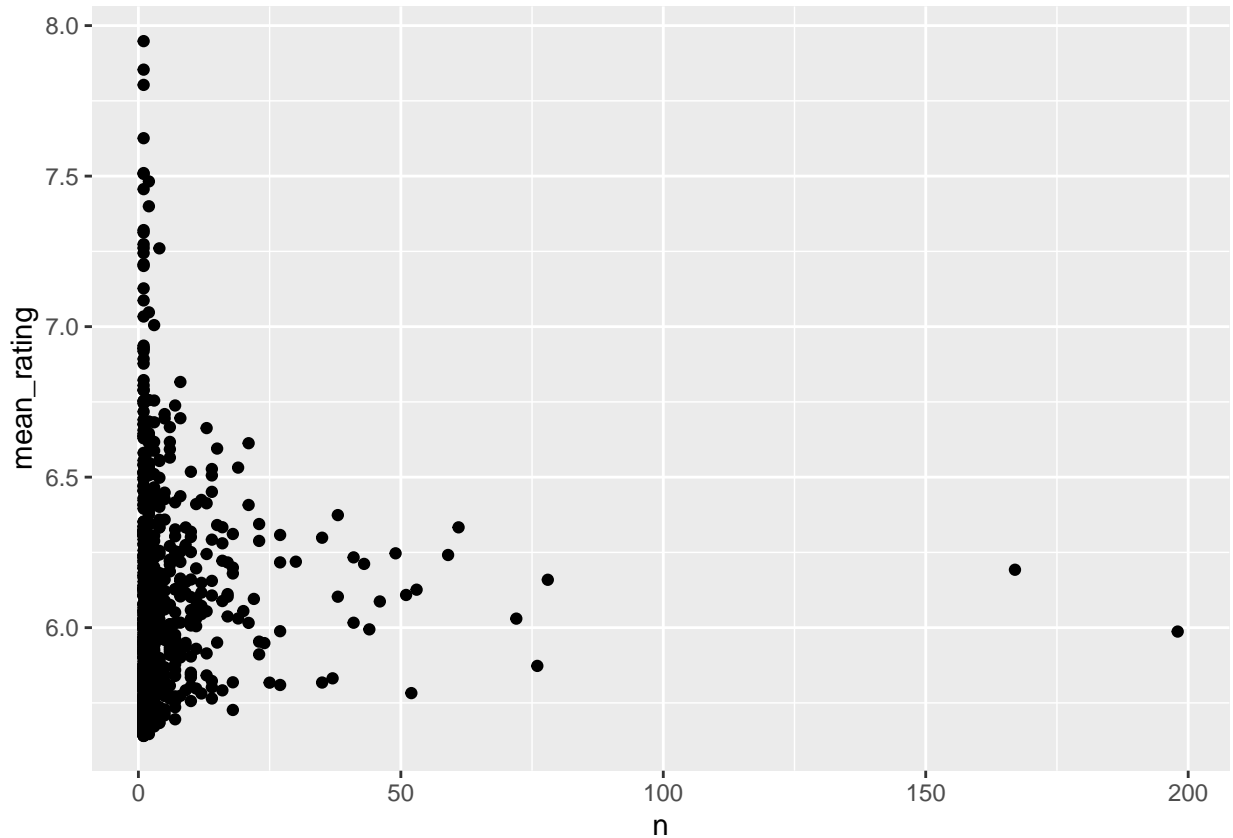
## [1] 0.2211273
```

As we see there is about a 29% improvement over the simple average by using the weight of the game to predict the ratings. This suggests that you are more likely to get a good game if you select a more complex game, but this result is not particularly robust. Next we will use a decision tree that will incorporate the differing game mechanics and “themes” (or categories) to improve these results.

## Results

To prepare the decision tree we will first need to clean the data a little bit. Both the mechanics field and the category field in the original data includes a lot of information. For example, the game “Gloomhaven” is described as an Adventure, Exploration, Fantasy, Fighting, Miniatures game that uses the Action / Movement Programming, Co-operative Play, Grid Movement, Hand Management, Modular Board, Role Playing, Simultaneous Action Selection, Storytelling, Variable Player Powers mechanics. This is so much information as to not be very useful. However, the first descriptors in both of these fields are primary and likely the most important when evaluating which game to buy. Therefore, we will extract just the information before the first comma in each of these fields and use it to build a decision tree.

Having cleaned the mechanic and category fields we will now plot the mean rating of each unique combination of mechanic and category. The next plot shows that more common mechanic/category combinations tend to be poorly rated. This is not to say that all unique combinations are winners, just that it appears that sticking to an old formula tends to lead to a worse game.



With this knowledge in hand, we can expect that the mechanics and categories will help us identify good games. The following decision tree uses this information to try to predict whether or not a given game will be in the top 10% of all games. First we build a simple tree using the information gleaned for the correlation matrix and use only the number of votes and the number of owners to predict good games. Then we will see how much the prediction can be improved by incorporating the mechanics and categories.

```
## [1] 0.9069149
```

```
## [1] 0.9428191
```

Here we see that using only the number of votes and owners the model correctly classifies a game 90.6%, which is not very good as simply predicting that every game is not in the top 10% will, by definition, be correct 90% of the time. By using mechanics and categories this is improved to 94.3%.

## Conclusions and Future Work

Overall it was difficult to discern any particularly useful patterns in this data other than trivial ones. For example, the fact that highly rated games are owned by more people is a strong pattern in the data, but also not very interesting as we would, of course, expect good games to be purchased by more people. BoardGameGeek’s “geek” rating is also perhaps a misleading variable because it might have the unintended consequence of hiding very good games that suffer from low distribution.

We were able to create a decision tree model which helps to predict top 10% games, but the accuracy was only 94.3%. This is significantly better than random guessing, but would need to be improved in order to be particularly useful. There is also a potential bias in this data as the ratings on BoardGameGeek are likely to skew toward serious board game enthusiasts and so might not be a good reflection of the opinions of the general public. It may be more useful to add additional data gathered from other sources to rectify this bias.