The Report of Steam Game Store

Jinyu Li BUID: U14978542

12/13/2021

Abstract

In this project, I would like to talk about the Steam game data. The questions come to my mind is: First, what is the correlation between the ratings of games and some properties for one game. Second, when fit the model, do these variable perform differently in different tags of games. The tag is the most common label players think for one game provided by Steam.

To make this project work smoothly, I firstly do some EDA to find what we can see through the data. Then, I conduct a multilevel regression model to get the relationship between the ratings and those variables. According to the model, I found variable a, b and c may correlate with ratings of games.

Introduction

video games are becoming more and more popular, especially during the pandemic and because of the RTX 30 series of Graphic Cards of Nvidia, which makes games run more smoothly and the graphics even more delicate. Many publishers take the advantage of this and provide their better service to attract people. Consequently, I would like to ask, except for the graphs, what aspects of games make them popular and attract people and I use the data from Steam.

As most of us known, The Steam platform is the largest digital distribution platform for PC gaming, holding around 75% of the market share in 2013. It is run by Valve, providing installations and automatic updates for Valve's games as well as games from third-party publishers and it's a good communities for players to discuss games, where we can find a lot of information whether one game is good or not from customers including their playtime and so on. Therefore, Steam games data are good resource for me to do some analysis.

To represent the degree of preference towards one game, I would like to use the log of rating ratio, which is the log(positive ratings/negative ratings) and will be explained why to make the transformation of the results in the model part. I will pick up some variables in the dataset, which can be important for one game and fit the model with them.

Method

Data Cleaning and Processing

The data is from Kaggle, named Steam Store Games (Clean dataset). Although the data seem to be clean, but it is not good enough for me to get some usefull information and some columns and words are not machine and human readable. Therefore I need to wrangle my data.

First, I extract some information in some columns and make some more dummy variables, for example in the "platforms" column it contains the systems(Linux, Windows, or Mac) that the game supports, I divide them into 3 dummy varibles to show which system is available for one game. Second, I change some variables

into values to make them readable. For example, the "owner" column represents the range of owners for one game like 500000-1000000, and I change them into rank value from 1 to 13 where 1 is the smallest amount of owners for one game. Third, I get another dataset on the same Kaggle page from the Steam community. It is the tags players give to each of the game. I got the most frequent tags of each game and left join them to my dataset. Last, I dropped some columns that I don't need like "appid", "english", etc.

After cleaning the data, I get the data I need for EDA and modeling, which is as follows:

Column	Description
name	Title of app (game)
release_date	Integers, release date for the game, in format YYYY-MM-DD
average_playtime	Integers, average user playtime, from SteamSpy
median_playtime	Integers, median user playtime, from SteamSpy
price	Float, Full price of title in GBP in 2019
platform	Integers,the number of systems the game supports, integer ranged 1 to 3
rating_ratio	Float, Positive ratings : Negative ratings
owner_rank	Integers, the size of owners, the maximus is 13
steamspy_tags	The most correlated genre of the game according to players' views
publisher	Name of the 1st publisher
developer	Name of the 1st developer
multiplayer	Dummy variables, if the game supports multiplayers
Windows	Dummy variables, if the game is available in Windows System
Mac	Dummy variables, if the game is available in Mac System
Linux	Dummy variables, if the game is available in Linux System
Steam_Cloud	Dummy variables, if the game surports Steam_Cloud
Anti_Cheat	Dummy variables, if the game surports Valve Anti-cheat
tags	The most requently used tags detected in another dataset

There are 6886 observations with 23 variables in the final version of my cleaned dataset.

Exploratory Data Analysis

To start with, I make one barplot to show how many games are there now for each tag and pick up the top 10 tag as follows:

Then, I ploted the distribution of log of rating ratio grouped by different tags. I picked the the log of rating ratio because when I draw the distribution of rating ratio, it looks not good and values are scattered and some values are too large. By using the log transformation, I avoid the problem and the distribution grouped by tags looks similar to normal distribution. The figure is as follows:

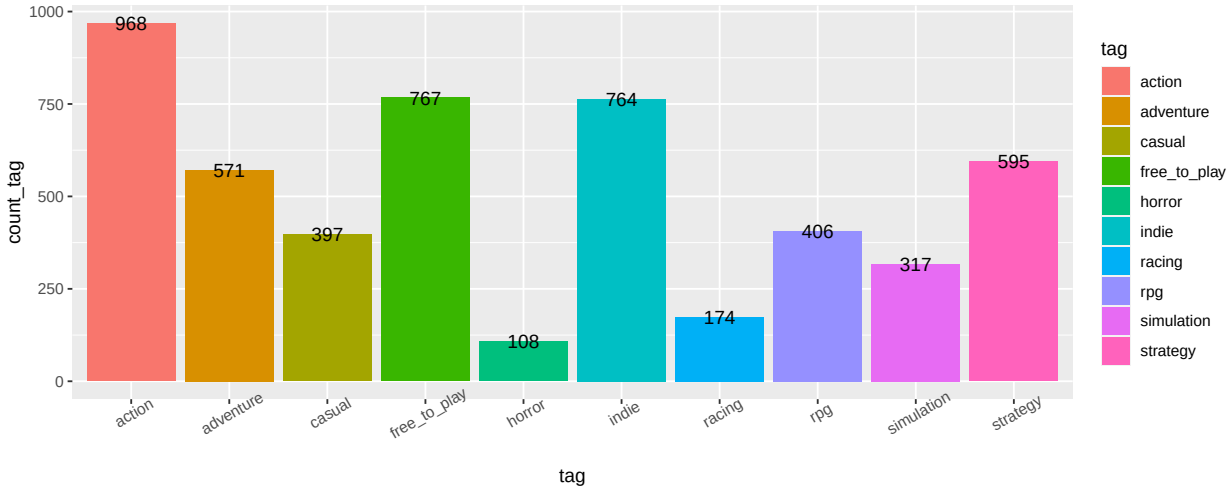
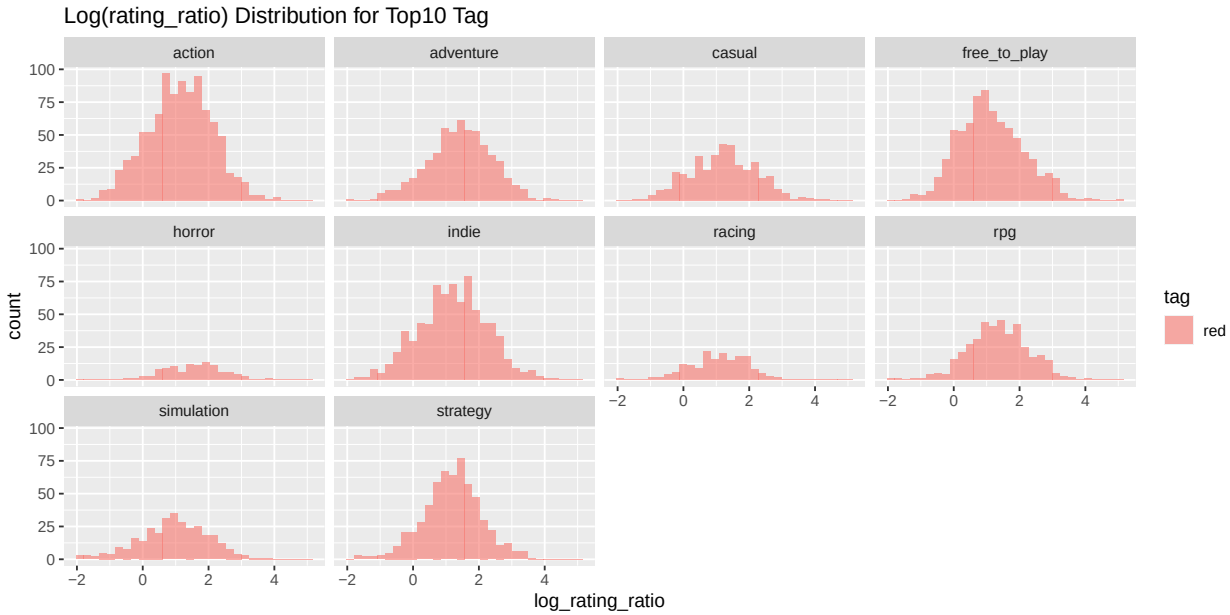
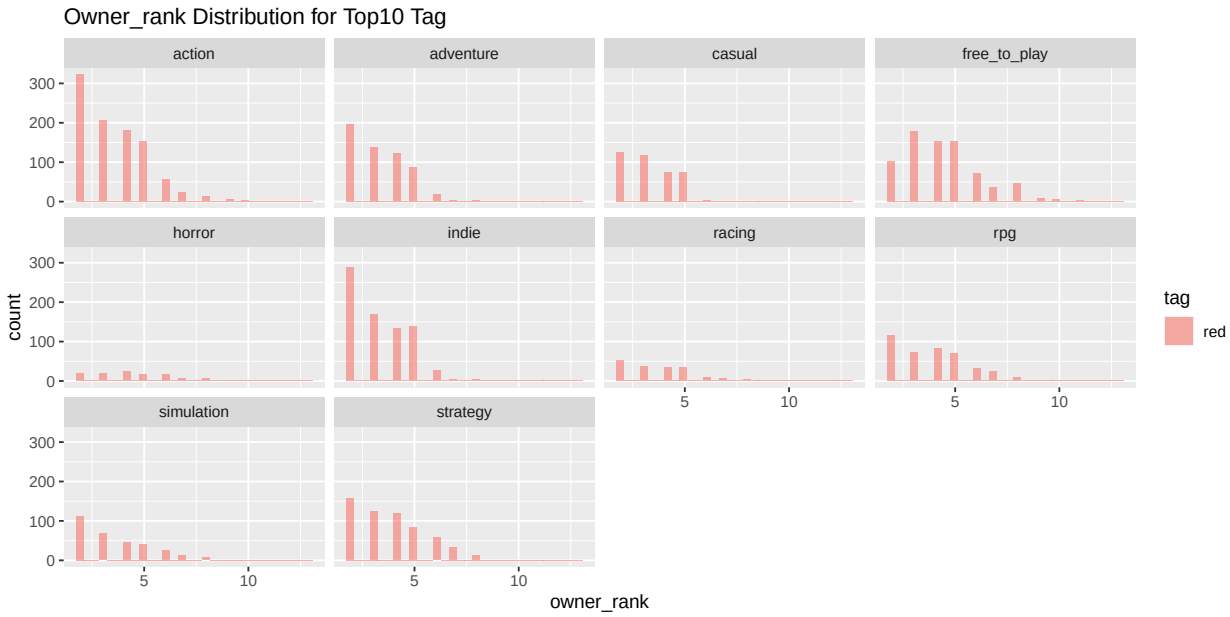


Figure 1: Top 10 steam game tags



For the same reason, I plot some other values like average playtime grouped by tags to check whether the distribution of those variables different in different tags. Here I will show one of them and put the rest(except for those dummy variables) into the appendix.



Model Fitting

According to the EDA, I picked up the variables below:

owner_rank	Anti_Cheat	multiplayer	Steam_Cloud
platform	achievements	price	

In this part, I will use a multilevel regression model. I will pickup the top 10 tags as the categories for the model, taking the "action" tag as an example, the model can be conducted as follows:

$$log(Rating\_Ratio) = 0.8410 + 0.0123 * (Owner\_Rank) + 0.0737 * (Anti\_Cheat) + 0.0124 * (Multiplayer) + 0.1621 * (Platform) - 0.00007 * (Achievements) + 0.0077 * (Price) + 0.2782 * (Steam\_Cloud)$$

where the random effects are "owner_rank", "Anti_Cheat", "multiplayer" and the intercept. I set my random effect based on the EDA and common knowledge. I set that variables like "multiplayer" as random effect, because it should be vary among different tags, and in some games, if they can compete with other, the game would be more fun.

To interpret the coefficient: In this case when the game is tagged as "Action", for the "owner_rank" and "platform", the larger they are, meaning the there are more owners and supporting systems for this game, the greater the log of rating ratio will be. When they grow every 1 unit respectively, the log of rating ratio increases by 0.0123 and 0.1621 respectively on average. For "Anti_Cheat", "Multiplayer" and "Steam_Cloud", when the game support these things respectively, the log of rating ratio increase by 0.0737, 0.0124, and 0.2782 respectively on average. And there are 2 variables not really making sense- "Achievements" and "Price", when the achievement increase 1 unit, the log of rating ration decrease 0.00007 on average; And when the price increase 1 unit, the log of rating ration increase 0.0077 on average.

After modeling, I conduct the residual analysis. The details of residual plots are in the Appendix. In the plot, the residual lies around 0 with out a clear pattern, which indicates that the regression looks good. Moreover, I use QQ-plot(see in the Appendix as well) to make another test, and in the plot most of the points lies on the line. In Resisuals vs Leverage Plot, we can find some outliers there but hard to find.

Result

According to the fitted model and the coefficients table in the Appendix, I can tell that the variable most of the variables correlated with the ratings of games, but surprisingly the price and achievements variables are not two of them. For those fixed variables, I find the platform and Steam_Cloud variables correlated to the rating a lot and may be 2 main factors of ratings, which makes sense. If one game support a lot systems and it is available in Steam Cloud so I can play wherever I go, it really means a lot and I could play them all the time.

Moreover, for the random effect part, it kind of answers my question that the ratings of different tag may correlated to different elements. The intercepts of each tags are different from one another. In this case, some are positive while other are not. And I find the multiplayer variable and anti_cheat variable vary a lot in different tags as well. It's good to know that talking about ratings, if games with tag of action, strategy and simulation support multiplayer, the ratings of these game can be higher than those don't. And if the games support anti-cheat system, people may feel better about the game (not causally), with strategy, simulation and racing game as exceptions.

So here are some conclusion, variables like owner_rank, Anti_Cheat, multiplayer, platform, Steam_Cloud correlated to the ratings of games more than other variables do. Also, games with different tags vary in ratings and people may judge these game with different grading systems, but we still don't know if there is any causal relationship and we need to look into it in the future.

Discussion

There are several limitations and I would like to issue or overcome in the future.

1. There are some confounding and colinearity issues. For example, actually for the games, they don't belong to one single tags or categories but several, so we need to promote the model to reduce these issues.
2. Some relationship among variables are still unknown. When we come to the conclusion that 2 variables are linear correlated or not correlated, we have no idea if there is other relationship and we don't know if the correlation is causal. So we need to make other models to estimate them if we do it in the future.
3. The data of each game are limited. For now, we only have some data for each game in general and we didn't get the information for each customer, which limit the analysis of customer. Meanwhile, if we need to get the sentiment analysis or some other related NLP analysis, we need to dig the comments and process them carefully, which takes a lot of time.

Reference

1.Steam wikipedia

2.Steam Store Games (Clean dataset)

Appendix

Coefficients of Random effect:

tag	owner_rank	Anti_Cheat	multiplayer	(Intercept)
action	-0.01230737	0.07371410	0.012399476	0.8410823
adventure	0.02215393	0.12781971	-0.307245284	0.5726270
casual	-0.11445635	0.01278460	-0.305467867	2.4679519
free_to_play	0.04329076	0.19825849	-0.571897566	0.2850026
horror	0.07302017	0.14483948	-0.064821286	-0.1564996
indie	-0.12100202	0.02322936	-0.197472447	2.5185063
racing	0.06237748	-0.09407389	-0.047948136	-0.1955601
rpg	0.14021900	0.09490795	-0.273847925	-1.2459433
simulation	0.14594236	-0.18467832	0.002282447	-1.5176855
strategy	0.09094057	-0.08288278	0.023332254	-0.6073685

Coefficients of fixed part:

platform1	achievements	price	Steam_Cloud
0.1620983	-6.921915e-05	0.007699993	0.2781679

The summary of fit2 information

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log_rating_ratio ~ platform1 + (owner_rank | tag) + achievements +
##     price + Steam_Cloud + (Anti_Cheat | tag) + (multiplayer |
##     tag) + (1 | tag)
##    Data: df1_select_tag
##
## REML criterion at convergence: 13719.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.5506 -0.6538 -0.0091  0.6633  4.2035
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  tag      (Intercept) 0.137230 0.37045
##           owner_rank  0.009818 0.09909  -0.98
##  tag.1    (Intercept) 0.012537 0.11197
##           Anti_Cheat  0.037897 0.19467  0.89
##  tag.2    (Intercept) 0.000000 0.00000
##           multiplayer 0.079678 0.28227   NaN
##  tag.3    (Intercept) 0.009351 0.09670
##  Residual             0.858768 0.92670
## Number of obs: 5067, groups:  tag, 10
##
## Fixed effects:
##                Estimate Std. Error t value
## (Intercept)   7.819e-01  6.056e-02  12.912
## platform1     1.621e-01  1.598e-02  10.145
```

```
## achievements -6.922e-05  4.583e-05  -1.510
## price         7.700e-03  1.951e-03   3.946
## Steam_Cloud   2.782e-01  2.900e-02   9.591
##
## Correlation of Fixed Effects:
##             (Intr) pltfr1 achvmn price
## platform1   -0.400
## achievemnts -0.030  0.008
## price       -0.191  0.010  0.005
## Steam_Cloud -0.041 -0.189 -0.022 -0.220
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

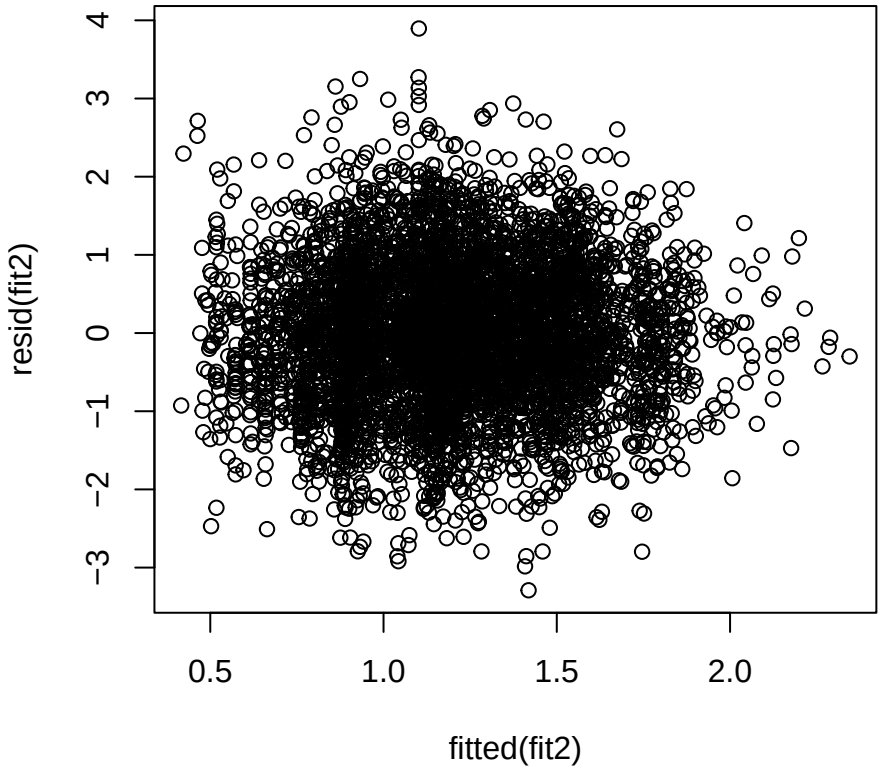


Figure 2: Residuals Plot

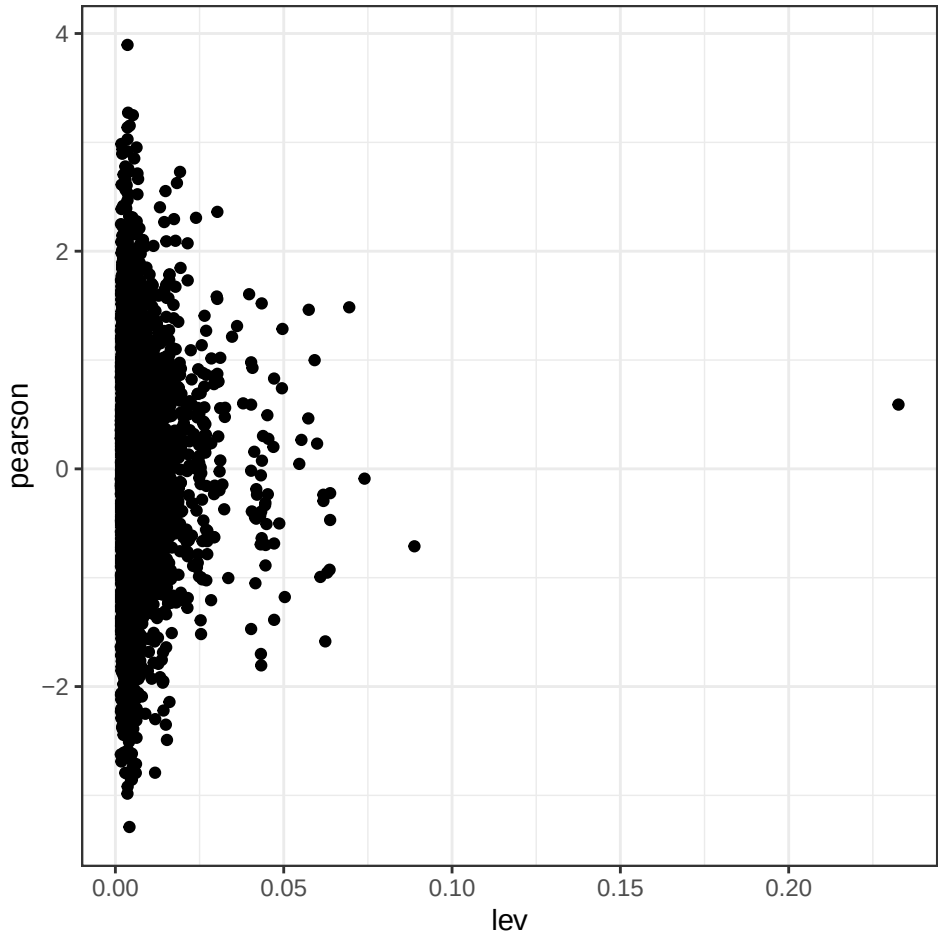
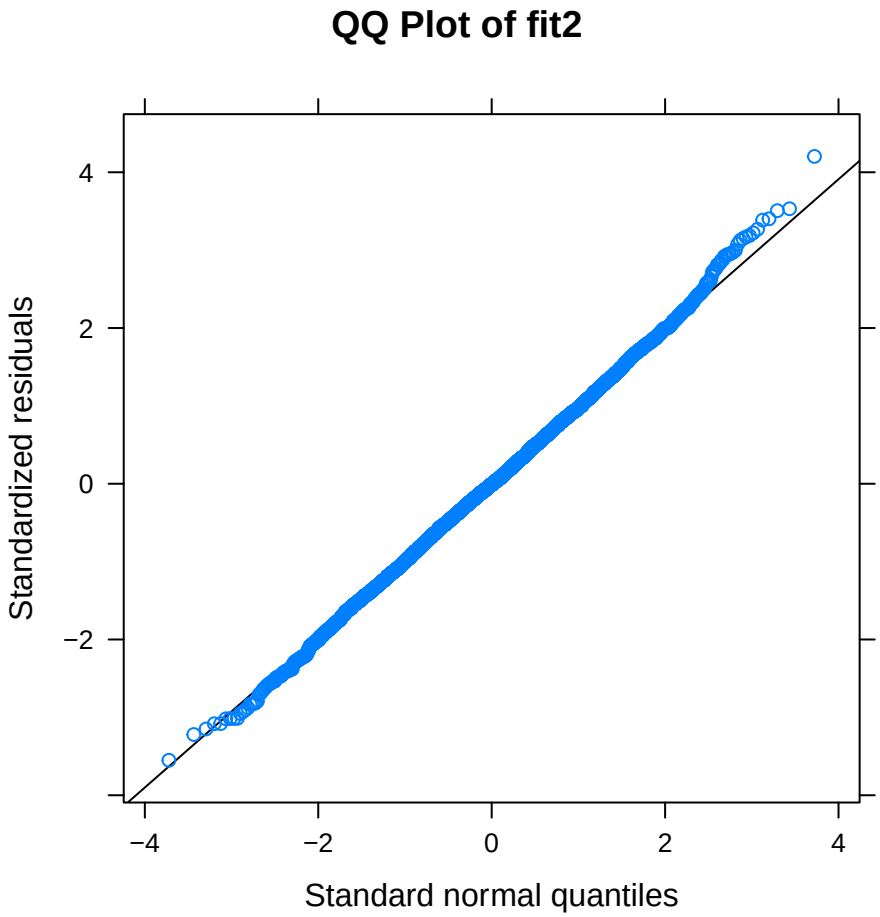


Figure 3: Residuals vs Leverage.


QQ Plot of fit2
Standardized residuals
Standard normal quantiles
4
2
0
−2
−4
−2
0
2
4

more plot

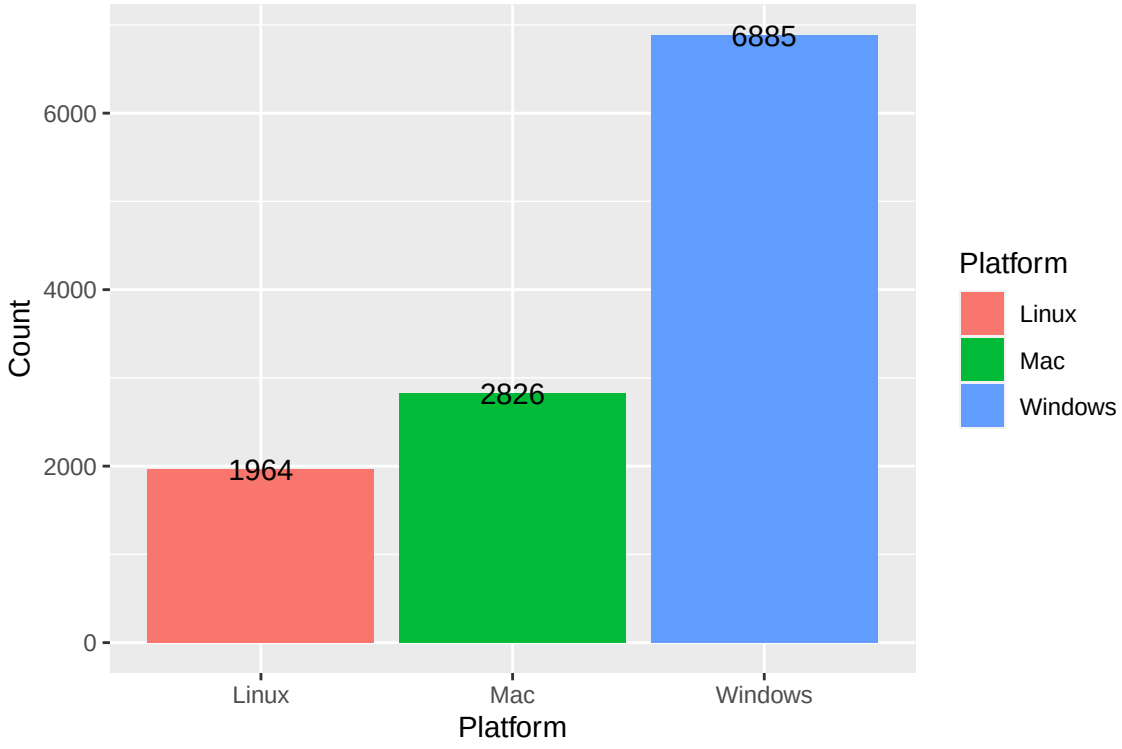


Figure 4: The Count of Games for different Platforms

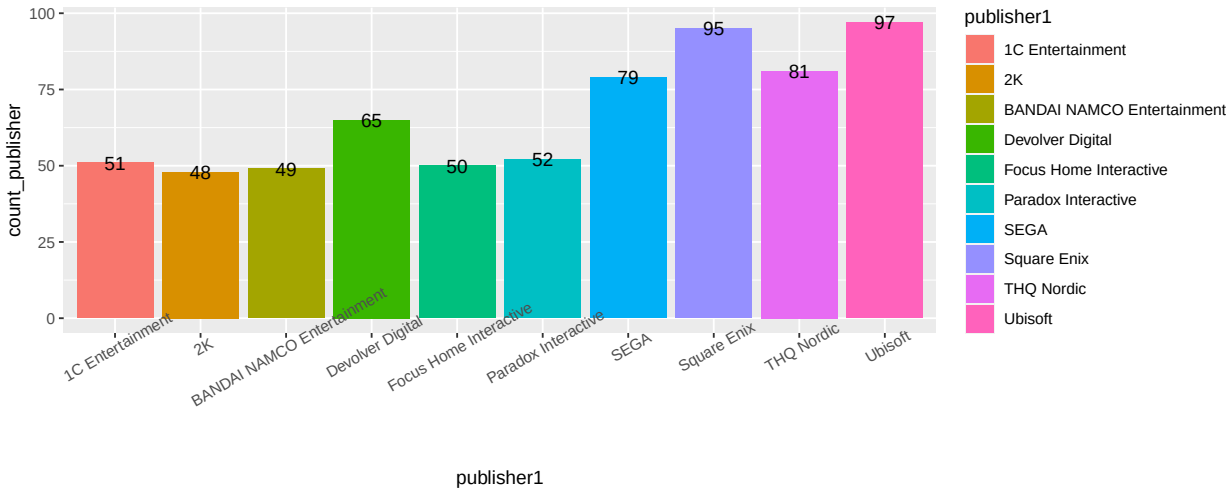


Figure 5: Top 10 Steam Publishers

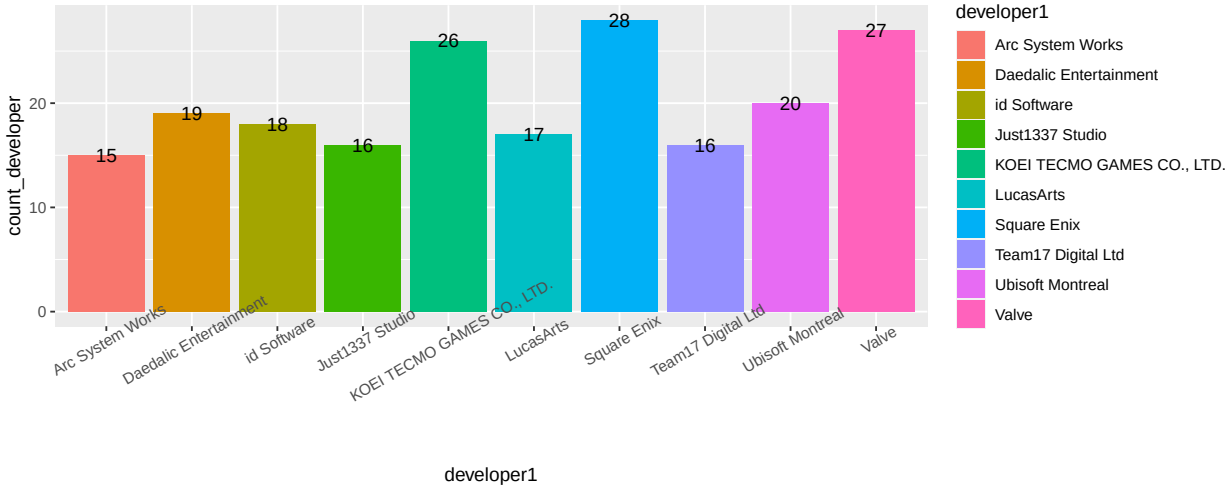


Figure 6: Top 10 Steam developers

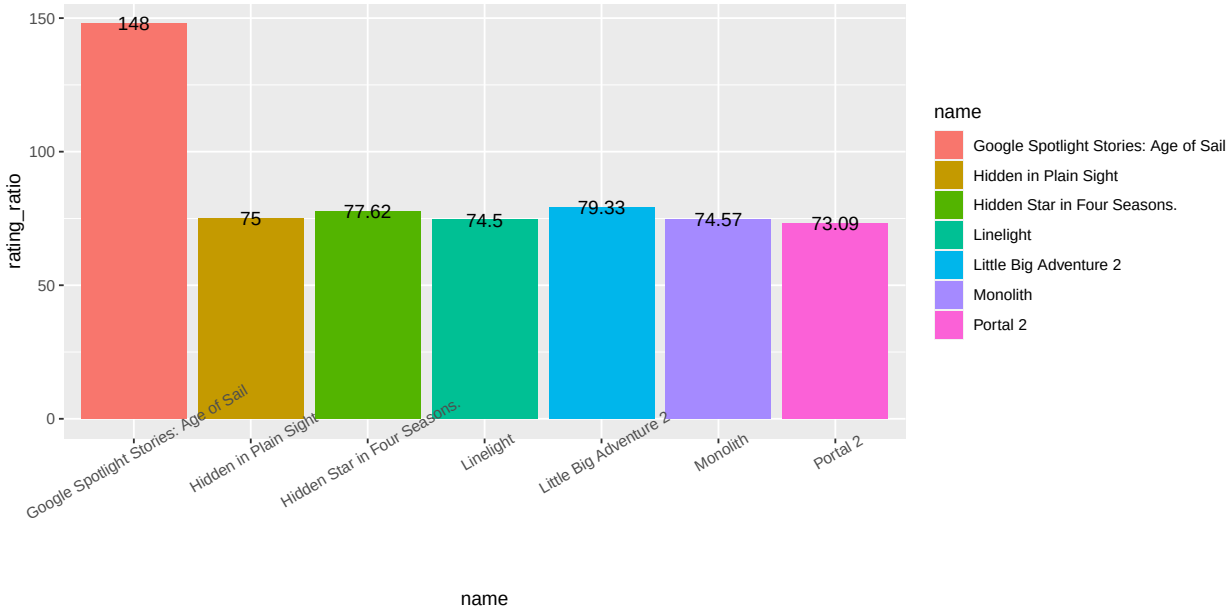


Figure 7: Top 7 rating ratio for Steam Games(app)

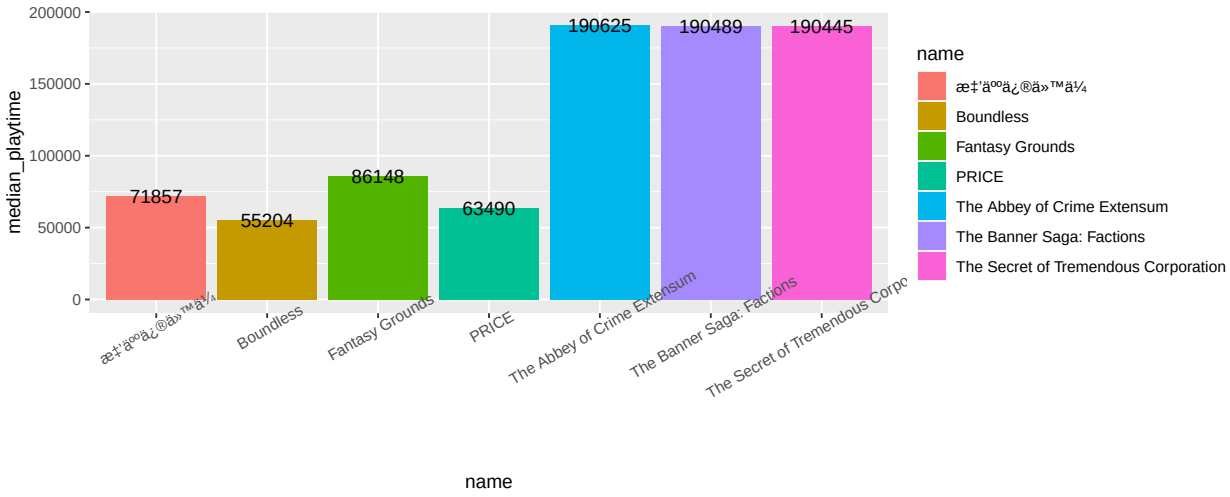


Figure 8: Top 7 median playtime for Steam Games

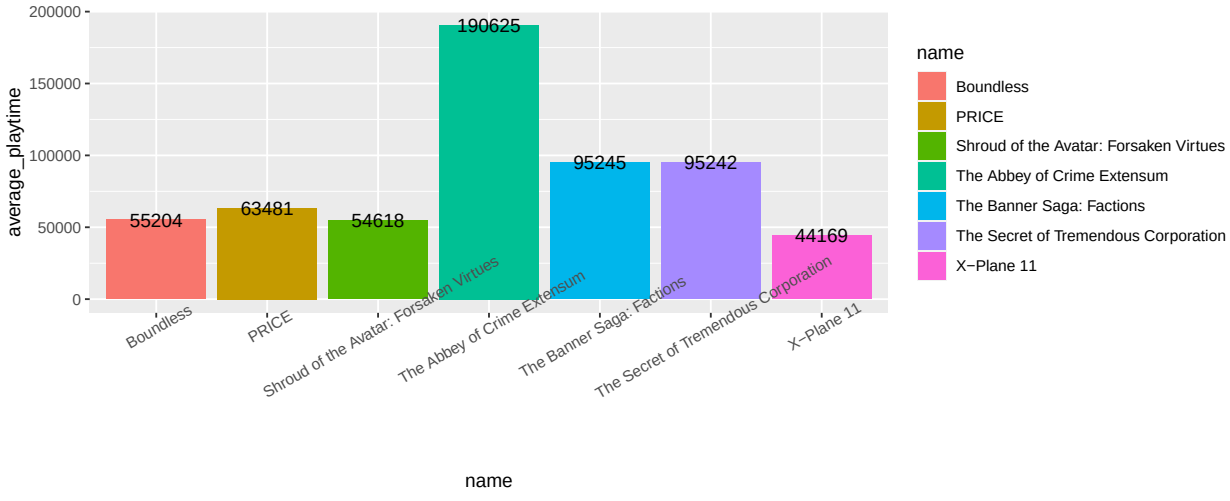


Figure 9: Top 7 average playtime for Steam Games

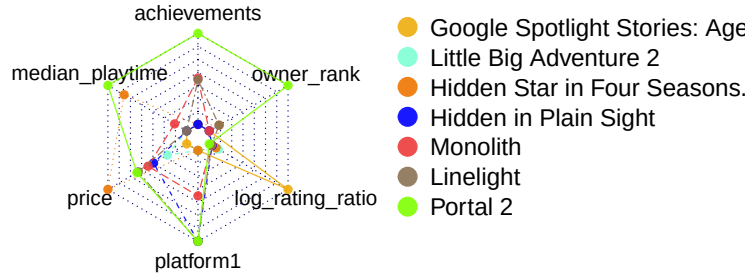


Figure 10: Radar charge for Top 7 rating ratio games

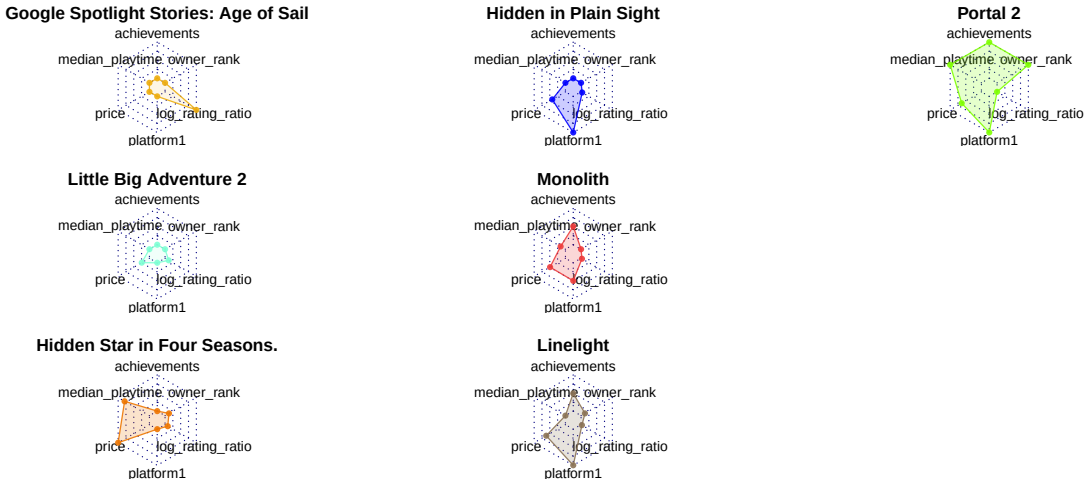


Figure 11: Radar charge for Top 7 rating ratio games seperatively

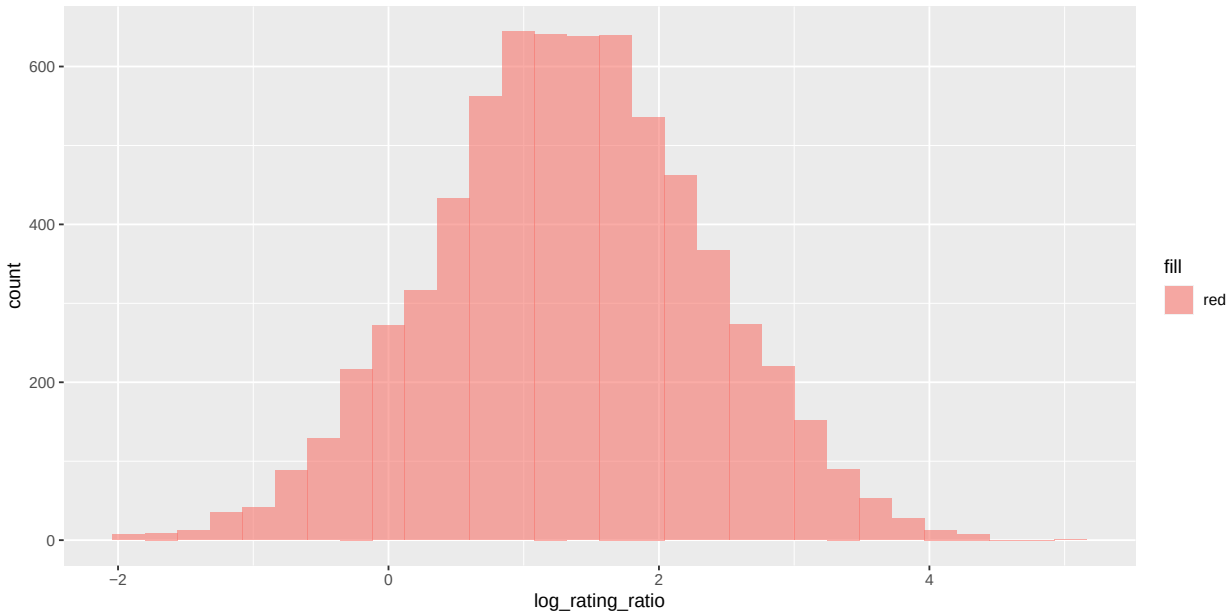


Figure 12: Radar charge for Top 7 rating ratio games

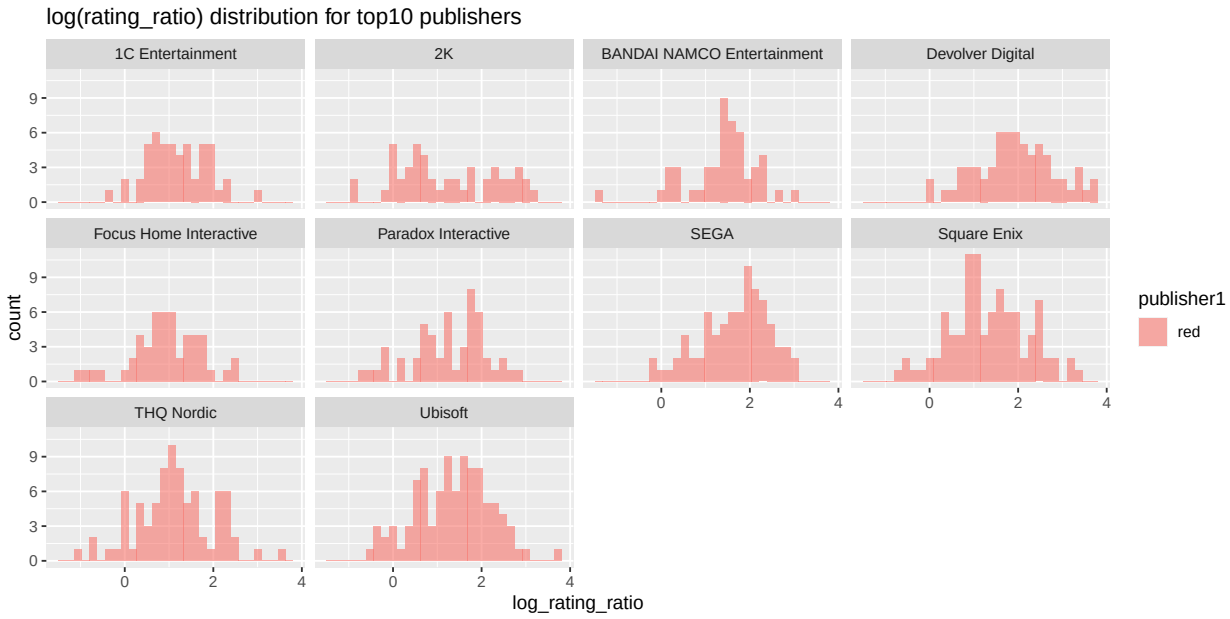


Figure 13: Radar charge for Top 7 rating ratio games