Universitat
de les Illes Balears

21746 - Data Mining

Final Project

Steam Successful Indie Games Study

Iván Pulgar Rodas
Jordi Sevilla Marí
Nahuel Vazquez
Yelyzaveta Denysova
Xiaozhe Cheng
Gabriel Oliver Artigues
Arturo Mus Mejías

2026-01-10

Contents

1 Introduction

1.1 Explanation of the Attributes

The dataset we will be working with contains a total number of 94948 observations and 47 columns or variables. The columns that will be used are described bellow:

- `appid`: Unique idetnifier of the game on Steam. [num]
- `name`: Name of the game. [text]
- **'released_date"**: Represents the date where the game was released. [time]
- **'required_age"**: Corresponds to the minimum age required to play the game. [num]
- `price`: How much the game costs. If its 0 it means that the game is Free to Play. [num]
- `dlc_count`: Ammount of DLCs (Downloadable Contents) the game has. [num]
- `suppport_url`: URL to the support page of the game. [text]
- `windows`: Determines if the game runs in windows. [categorical]
- `mac`: Determines if the game runs in mac. [categorical]
- `linux`: Determines if the game runs in linux. [categorical]
- `metacritic_score`: Metacritic score based on critcical reviews (reviews from professionalss). By performing an investigation we think that an score of 0 means that when the scraping of the data was done there where no reviews for that game yet. [num]
- `achievements`: Number of achievements the game has. [num]
- **'recommendations"**: Ammount of user recommendations. [num]
- `supported_languages`: List of languages that the game supports.[**NO SÉ QUE CATEGORÍA DARLE**]
- `packages`: Avaliable packages for the game. It contains the name and a description of the package and the names, descriptions and subprices of the subpackages. [**LO MISMO QUE ARRIBA**]
- `Developers`: List of developers associated with the game. [**LO MISMO QUE ARRIBA**]
- `publishers`: List of publishers associated with the game. [**LO MISMO QUE ARRIBA**]
- `categories`: List of categories that the game has. [**LO MISMO QUE ARRIBA**]
- `genres`: List of genres that the game belongs to. [**LO MISMO QUE ARRIBA**]
- `positive`: Ammount of positive votes the game has. [num]
- `negative`: Ammount of negative votes the game has. [num]
- `estimated_owners`: Estimated owners of the game. [text]
- `average_playtime_forever`: Average playtime since March 2009 measured in minutes. [num]
- `average_playtime_2weeks`: Average playtime in the last two weeks measured in minutes. [num]
- `median_playtime_forever`: Median playtime since March 2009 measured in minutes. [num]
- `median_playtime_2weeks`: Median playtime in the last two weeks measured in minutes.[num]
- `peak_ccu`: Number of current users playing the day before the data was scrapped. [num]
- `tags`: List of tags the game has with its name and its key. [**NO SE QUE CATEGORIA DARLE**]
- `pct_pos_total`: Percentage of all reviews that are positive. [num]
- `num_reviews_total`: Nummber of the total reviews the game has. [num]

1.2 Objectives

TODO: ADD OBJECTIVES OF STUDY

2 Procesing de data

General look of the dataset

```
summary(steam)
```

```
##      appid              name           release_date        required_age
##  Min.   :     20   Length:94948       Length:94948       Min.   :-1.0000
##  1st Qu.: 887338   Class :character   Class :character   1st Qu.: 0.0000
```

```
##  Median :1591145   Mode  :character   Mode  :character   Median : 0.0000
##  Mean   :1707531                                         Mean   : 0.1783
##  3rd Qu.:2491703                                         3rd Qu.: 0.0000
##  Max.   :3570420                                         Max.   :21.0000
##
##      price            dlc_count         detailed_description about_the_game
##  Min.   :   0.000   Min.   :   0.0000   Length:94948         Length:94948
##  1st Qu.:   0.990   1st Qu.:   0.0000   Class :character     Class :character
##  Median :   3.990   Median :   0.0000   Mode  :character     Mode  :character
##  Mean   :   6.911   Mean   :   0.5632
##  3rd Qu.:   9.990   3rd Qu.:   0.0000
##  Max.   : 999.980   Max.   :3427.0000
##
##  short_description    reviews          header_image         website
##  Length:94948       Length:94948       Length:94948       Length:94948
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  support_url        support_email        windows             mac
##  Length:94948       Length:94948       Length:94948       Length:94948
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     linux           metacritic_score metacritic_url      achievements
##  Length:94948       Min.   : 0.000   Length:94948       Min.   :   0.00
##  Class :character   1st Qu.: 0.000   Class :character   1st Qu.:   0.00
##  Mode  :character   Median : 0.000   Mode  :character   Median :   2.00
##                     Mean   : 2.764                      Mean   :  19.54
##                     3rd Qu.: 0.000                      3rd Qu.:  19.00
##                     Max.   :97.000                      Max.   :9821.00
##
##  recommendations      notes           supported_languages full_audio_languages
##  Min.   :      0   Length:94948       Length:94948        Length:94948
##  1st Qu.:      0   Class :character   Class :character    Class :character
##  Median :      0   Mode  :character   Mode  :character    Mode  :character
##  Mean   :   1022
##  3rd Qu.:      0
##  Max.   :4401572
##
##    packages          developers         publishers         categories
##  Length:94948       Length:94948       Length:94948       Length:94948
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     genres          screenshots           movies            user_score
```

```
##  Length:94948       Length:94948       Length:94948       Min.   :  0.00000
##  Class :character   Class :character   Class :character   1st Qu.:  0.00000
##  Mode  :character   Mode  :character   Mode  :character   Median :  0.00000
##                                                           Mean   :  0.03097
##                                                           3rd Qu.:  0.00000
##                                                           Max.   :100.00000
##
##    score_rank        positive          negative         estimated_owners
##  Min.   : 98.00   Min.   :      0   Min.   :      0.0   Length:94948
##  1st Qu.: 99.00   1st Qu.:      0   1st Qu.:      0.0   Class :character
##  Median : 99.00   Median :      8   Median :      2.0   Mode  :character
##  Mean   : 99.13   Mean   :   1218   Mean   :    202.1
##  3rd Qu.:100.00   3rd Qu.:     51   3rd Qu.:     15.0
##  Max.   :100.00   Max.   :7480813   Max.   :1135108.0
##  NA's   :94909
##  average_playtime_forever average_playtime_2weeks median_playtime_forever
##  Min.   :      0.0        Min.   :    0.000       Min.   :      0.0
##  1st Qu.:      0.0        1st Qu.:    0.000       1st Qu.:      0.0
##  Median :      0.0        Median :    0.000       Median :      0.0
##  Mean   :    108.6        Mean   :    4.757       Mean   :    108.4
##  3rd Qu.:      0.0        3rd Qu.:    0.000       3rd Qu.:      0.0
##  Max.   :1462997.0        Max.   :18568.000       Max.   :1462997.0
##
##  median_playtime_2weeks    discount          peak_ccu
##  Min.   :    0.000      Min.   :  0.000   Min.   :0.000e+00
##  1st Qu.:    0.000      1st Qu.:  0.000   1st Qu.:0.000e+00
##  Median :    0.000      Median :  0.000   Median :0.000e+00
##  Mean   :    5.018      Mean   :  4.307   Mean   :9.285e+01
##  3rd Qu.:    0.000      3rd Qu.:  0.000   3rd Qu.:0.000e+00
##  Max.   :18568.000      Max.   :100.000   Max.   :1.212e+06
##
##      tags           pct_pos_total    num_reviews_total pct_pos_recent
##  Length:94948       Min.   : -1.00   Min.   :     -1   Min.   : -1.000
##  Class :character   1st Qu.: -1.00   1st Qu.:     -1   1st Qu.: -1.000
##  Mode  :character   Median : 58.00   Median :     15   Median : -1.000
##                     Mean   : 44.63   Mean   :   1448   Mean   :  5.328
##                     3rd Qu.: 84.00   3rd Qu.:     80   3rd Qu.: -1.000
##                     Max.   :100.00   Max.   :8632939   Max.   :100.000
##
##  num_reviews_recent
##  Min.   :   -1.00
##  1st Qu.:   -1.00
##  Median :   -1.00
##  Mean   :   16.88
##  3rd Qu.:   -1.00
##  Max.   :96473.00
##
```

Explanation of first looks of it, bad formatting, NA's, negative values...

2.1 Handling of NA values

The attributes with missing values are:

```r
na_counts <- steam %>% summarise_all(~ sum(is.na(.)))

print(na_counts)
```

```
##   appid name release_date required_age price dlc_count detailed_description
## 1     0    0            0            0     0         0                     0
##   about_the_game short_description reviews header_image website support_url
## 1              0                 0       0            0       0           0
##   support_email windows mac linux metacritic_score metacritic_url achievements
## 1             0       0   0     0                0              0           0
##   recommendations notes supported_languages full_audio_languages packages
## 1               0     0                   0                   0        0
##   developers publishers categories genres screenshots movies user_score
## 1          0          0          0      0           0      0          0
##   score_rank positive negative estimated_owners average_playtime_forever
## 1      94909        0        0               0                        0
##   average_playtime_2weeks median_playtime_forever median_playtime_2weeks
## 1                       0                       0                      0
##   discount peak_ccu tags pct_pos_total num_reviews_total pct_pos_recent
## 1        0        0    0             0                 0              0
##   num_reviews_recent
## 1                  0
```

The columns with missing values are as follows:

LOs juegos con número de reseñas = -1 creemos que son porque el scrappear ha fallado durante su ejecución. Si vamos a trabajar con las reseñas podríamos decir en la presentación que vamos a probar de volver a intentar scrappear la información

```
## tibble [88,982 x 30] (S3: tbl_df/tbl/data.frame)
##  $ appid                   : int [1:88982] 2556940 449940 1287250 866510 870990 439260 388390 224356
##  $ name                    : chr [1:88982] "! Shakabula *" "! That Bastard Is Trying To Steal Our Go
##  $ release_date            : Date[1:88982], format: "2023-10-13" "2016-03-03" ...
##  $ required_age            : int [1:88982] 0 0 0 0 0 0 0 0 0 0 ...
##  $ price                   : num [1:88982] 14.99 2.99 19.99 1.99 0.99 ...
##  $ dlc_count               : int [1:88982] 0 0 0 39 0 0 0 1 3 0 ...
##  $ windows                 : chr [1:88982] "True" "True" "True" "True" ...
##  $ mac                     : chr [1:88982] "False" "False" "False" "False" ...
##  $ linux                   : chr [1:88982] "False" "True" "False" "False" ...
##  $ metacritic_score        : int [1:88982] 0 0 0 0 0 0 0 0 0 0 ...
##  $ achievements            : int [1:88982] 0 0 9 4997 2021 0 0 19 13 5 ...
##  $ recommendations         : int [1:88982] 0 0 0 495 0 0 0 108 0 0 ...
##  $ supported_languages     : chr [1:88982] "['English']" "['English']" "['English', 'Simplified Chin
##  $ packages                : chr [1:88982] "[{'title': 'Buy ! Shakabula *', 'description': '', 'subs
##  $ developers              : chr [1:88982] "['Skermunkel']" "['WTFOMGames']" "['Andreev Worlds']" "[
##  $ publishers              : chr [1:88982] "['Skermunkel']" "['WTFOMGames']" "['Andreev Worlds']" "[
##  $ categories              : chr [1:88982] "['Single-player', 'Full controller support', 'Steam Clou
##  $ genres                  : chr [1:88982] "['Action', 'Indie', 'RPG', 'Early Access']" "['Action',
##  $ positive                : int [1:88982] 0 57 45 410 25 83 37 126 0 0 ...
##  $ negative                : int [1:88982] 4 78 34 180 32 18 102 10 0 0 ...
##  $ average_playtime_forever: int [1:88982] 0 312 0 360 0 0 244 0 0 0 ...
##  $ average_playtime_2weeks : int [1:88982] 0 0 0 0 0 0 0 0 0 0 ...
##  $ median_playtime_forever : int [1:88982] 0 391 0 378 0 0 244 0 0 0 ...
##  $ median_playtime_2weeks  : int [1:88982] 0 0 0 0 0 0 0 0 0 0 ...
##  $ peak_ccu                : int [1:88982] 0 0 0 6 0 0 0 0 0 0 ...
```

```
##  $ tags                    : chr [1:88982] "{'Early Access': 213, 'Action': 193, 'RPG': 187, 'JRPG':
##  $ pct_pos_total           : int [1:88982] -1 55 61 71 55 82 55 91 95 66 ...
##  $ num_reviews_total       : int [1:88982] -1 68 62 495 18 101 20 108 281 12 ...
##  $ estimated_owners_min    : int [1:88982] 0 50000 0 100000 0 0 100000 20000 0 0 ...
##  $ estimated_owners_max    : int [1:88982] 20000 100000 20000 200000 20000 20000 200000 50000 0 0 ..
```

```
##      appid              name           release_date         required_age
##  Min.   :     20   Length:88982       Min.   :1997-06-30   Min.   :-1.0000
##  1st Qu.: 852783   Class :character   1st Qu.:2018-12-05   1st Qu.: 0.0000
##  Median :1522535   Mode  :character   Median :2021-10-29   Median : 0.0000
##  Mean   :1655079                      Mean   :2021-04-04   Mean   : 0.1826
##  3rd Qu.:2429338                      3rd Qu.:2023-12-14   3rd Qu.: 0.0000
##  Max.   :3542350                      Max.   :2025-03-10   Max.   :21.0000
##      price            dlc_count           windows              mac
##  Min.   :   0.000   Min.   :   0.0000   Length:88982       Length:88982
##  1st Qu.:   0.990   1st Qu.:   0.0000   Class :character   Class :character
##  Median :   4.990   Median :   0.0000   Mode  :character   Mode  :character
##  Mean   :   7.349   Mean   :   0.5957
##  3rd Qu.:   9.990   3rd Qu.:   0.0000
##  Max.   : 999.980   Max.   :3427.0000
##     linux           metacritic_score  achievements     recommendations
##  Length:88982       Min.   : 0.00    Min.   :   0.00   Min.   :      0
##  Class :character   1st Qu.: 0.00    1st Qu.:   0.00   1st Qu.:      0
##  Mode  :character   Median : 0.00    Median :   5.00   Median :      0
##                     Mean   : 2.91    Mean   :  20.67   Mean   :   1013
##                     3rd Qu.: 0.00    3rd Qu.:  20.00   3rd Qu.:      0
##                     Max.   :97.00    Max.   :9821.00   Max.   :4401572
##  supported_languages   packages          developers         publishers
##  Length:88982        Length:88982       Length:88982       Length:88982
##  Class :character    Class :character   Class :character   Class :character
##  Mode  :character    Mode  :character   Mode  :character   Mode  :character
##
##
##
##   categories           genres             positive          negative
##  Length:88982       Length:88982       Min.   :      0   Min.   :      0.0
##  Class :character   Class :character   1st Qu.:      1   1st Qu.:      0.0
##  Mode  :character   Mode  :character   Median :     10   Median :      2.0
##                                        Mean   :   1267   Mean   :    208.3
##                                        3rd Qu.:     59   3rd Qu.:     17.0
##                                        Max.   :7480813   Max.   :1135108.0
##  average_playtime_forever average_playtime_2weeks median_playtime_forever
##  Min.   :      0.0        Min.   :    0.000       Min.   :      0.0
##  1st Qu.:      0.0        1st Qu.:    0.000       1st Qu.:      0.0
##  Median :      0.0        Median :    0.000       Median :      0.0
##  Mean   :    115.6        Mean   :    4.998       Mean   :    115.5
##  3rd Qu.:      0.0        3rd Qu.:    0.000       3rd Qu.:      0.0
##  Max.   :1462997.0        Max.   :18568.000       Max.   :1462997.0
##  median_playtime_2weeks    peak_ccu            tags           pct_pos_total
##  Min.   :    0.000      Min.   :0.000e+00   Length:88982       Min.   : -1.00
##  1st Qu.:    0.000      1st Qu.:0.000e+00   Class :character   1st Qu.: -1.00
##  Median :    0.000      Median :0.000e+00   Mode  :character   Median : 60.00
##  Mean   :    5.277      Mean   :9.778e+01                      Mean   : 45.39
##  3rd Qu.:    0.000      3rd Qu.:0.000e+00                      3rd Qu.: 84.00
```

```
##  Max.   :18568.000      Max.   :1.212e+06                      Max.   :100.00
##  num_reviews_total estimated_owners_min estimated_owners_max
##  Min.   :     -1   Min.   :        0    Min.   :        0
##  1st Qu.:     -1   1st Qu.:        0    1st Qu.:    20000
##  Median :     15   Median :        0    Median :    20000
##  Mean   :   1320   Mean   :    59038    Mean   :   143056
##  3rd Qu.:     81   3rd Qu.:        0    3rd Qu.:    20000
##  Max.   :8632939   Max.   :200000000    Max.   :500000000
```

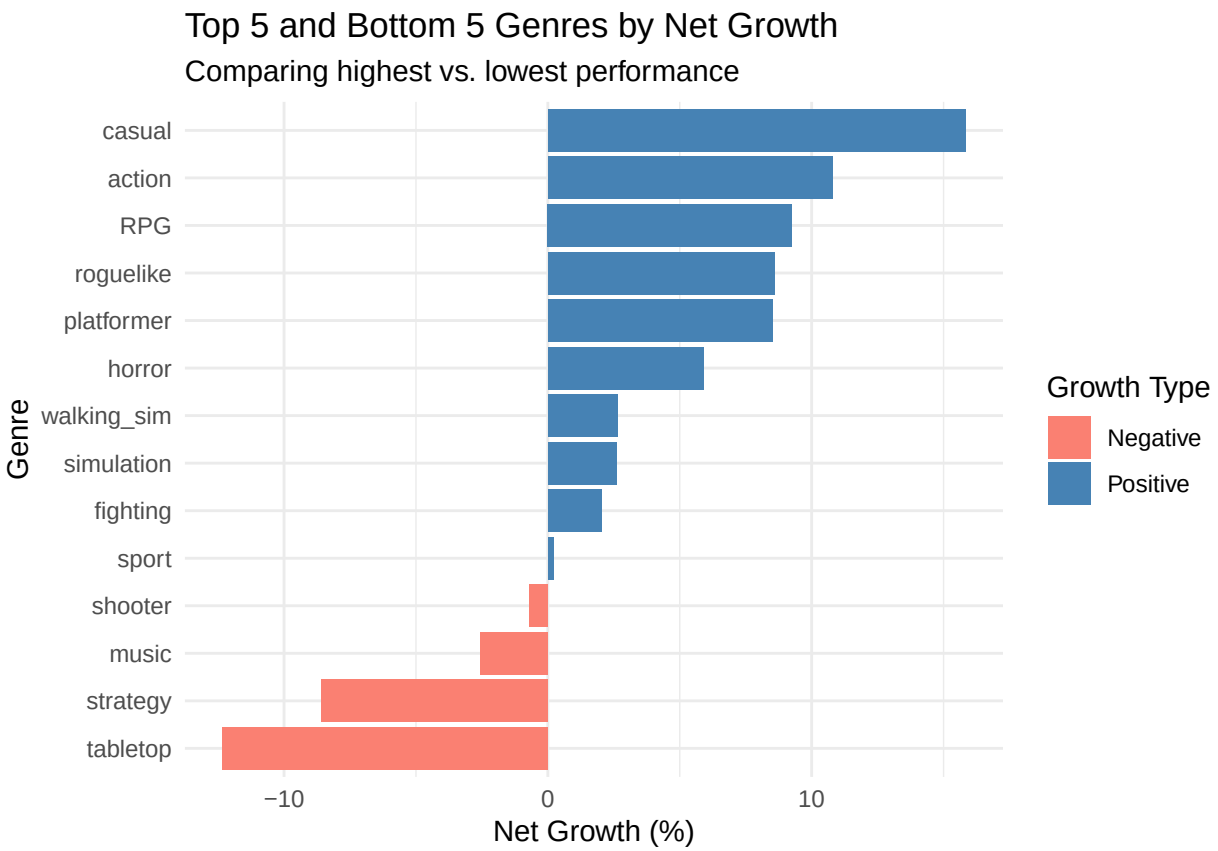
3 Exploratory Data Analysis

Now we going to explore the market share of each genre and how it evolves through time, and the playtime of each genre trying to see which are the genres with more hardcore players.

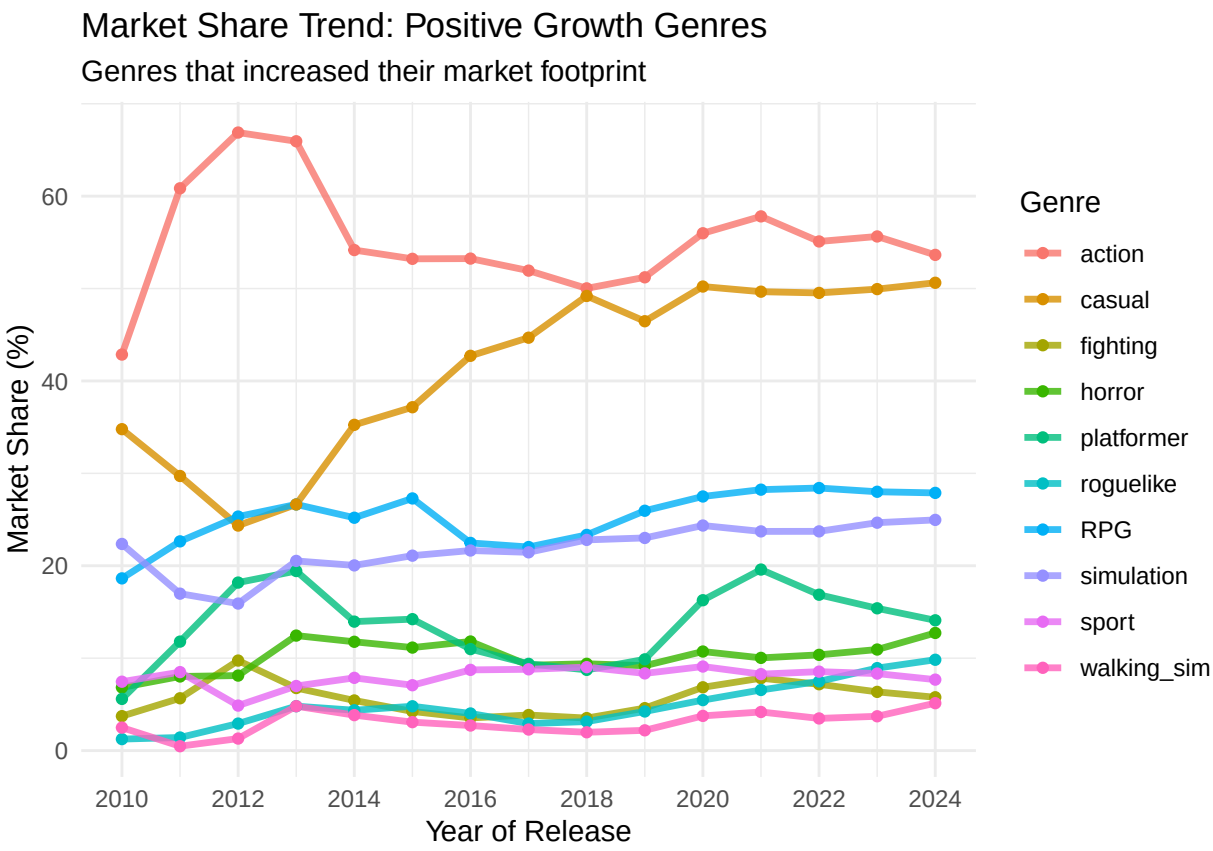
Firstly for having a objective view if the genre have truly grown we going to mesure the market share of each genre of each year. The advantage of using this is that the grown of each year is relative to the total grown of the game industry, if we use the raw game numbers we cannot difference if the game have actually grown or it's because the general game market have a growing tendency.

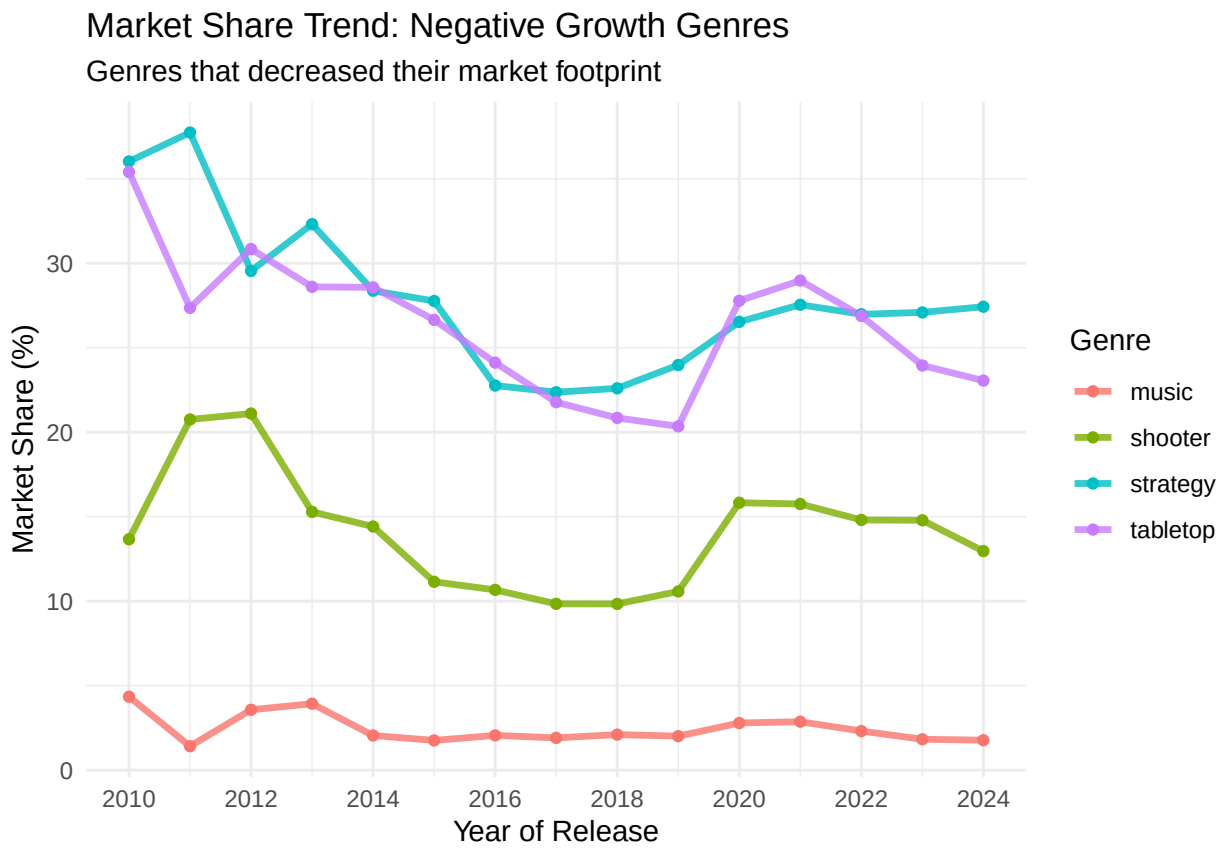
$$\text{Market Share} = \frac{\text{Owners of game } i}{\text{Total owners}}$$

```
## # A tibble: 14 x 3
##    Genre       Total_Volatility Net_Growth
##    <chr>                  <dbl>      <dbl>
##  1 action                  54.0     10.8
##  2 casual                  43.6     15.8
##  3 platformer              41.4      8.51
##  4 tabletop                36.6    -12.3
##  5 strategy                28.8     -8.60
##  6 shooter                 27.6     -0.703
##  7 RPG                     23.7      9.25
##  8 fighting                19.3      2.04
##  9 simulation              18.1      2.60
## 10 horror                  15.4      5.90
## 11 sport                   13.8      0.224
## 12 walking_sim             13.7      2.64
## 13 roguelike               13.2      8.59
## 14 music                   10.3     -2.58
```



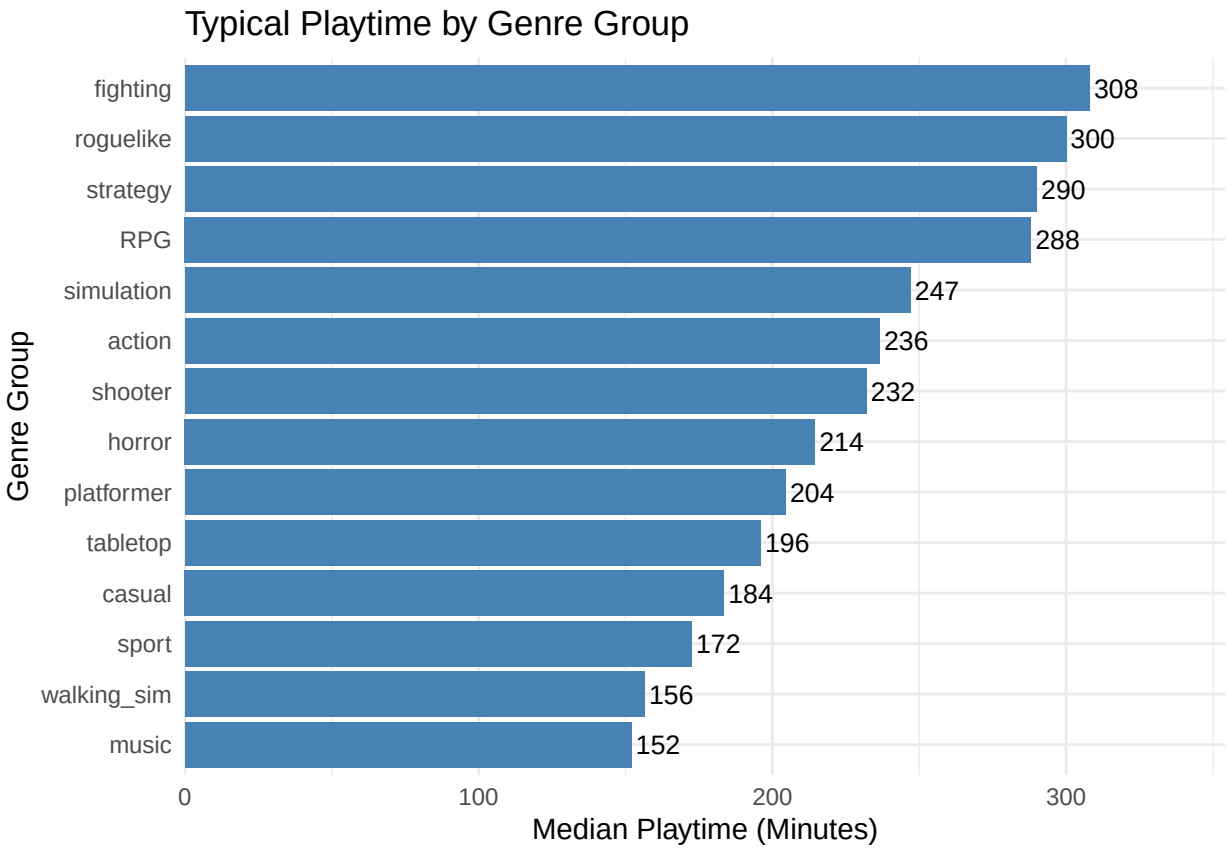

```
positive_genres <- genre_grow %>% filter(Net_Growth > 0) %>% pull(Genre)
negative_genres <- genre_grow %>% filter(Net_Growth < 0) %>% pull(Genre)
```

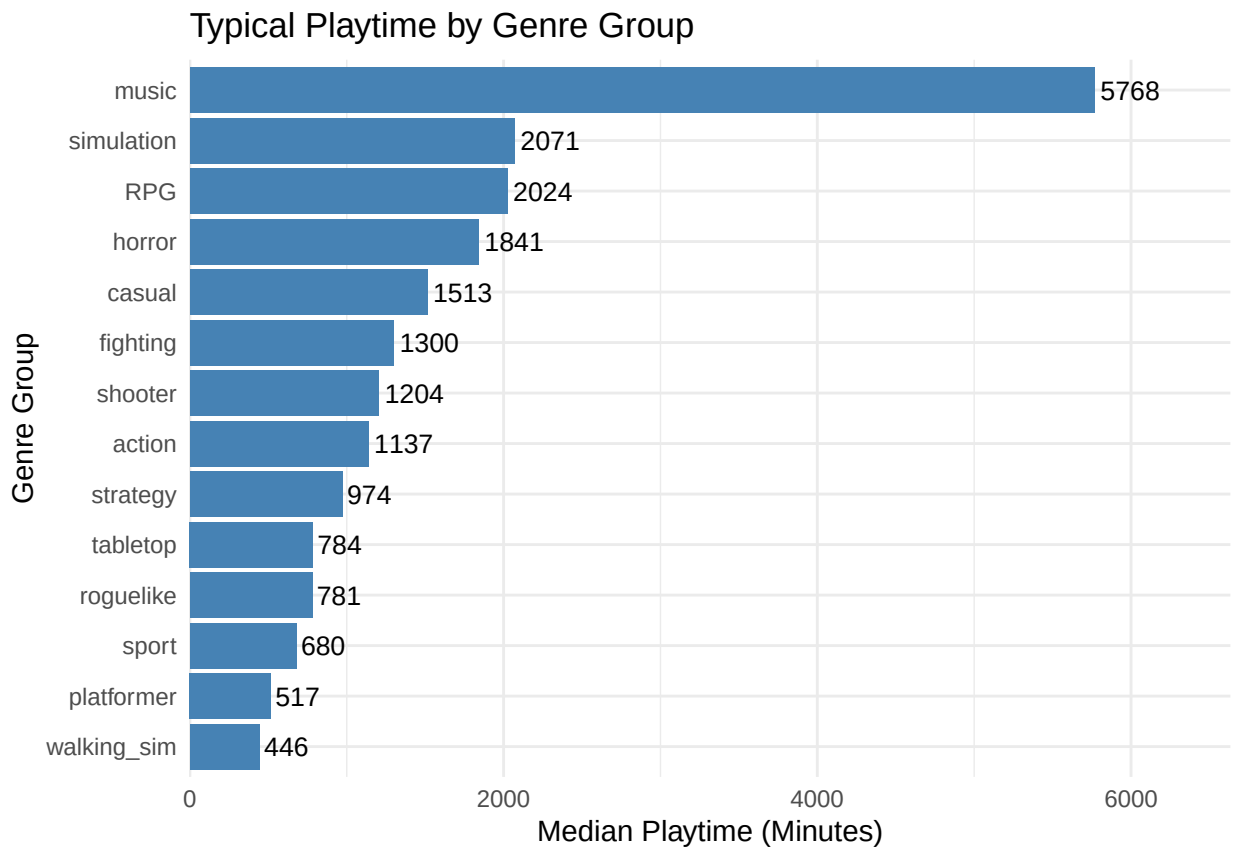

Market Share Trend: Positive Growth Genres
Genres that increased their market footprint
Market Share (%)
0
20
40
60
2010
2012
2014
2016
2018
2020
2022
2024
Year of Release
Genre
action
casual
fighting
horror
platformer
roguelike
RPG
simulation
sport
walking_sim

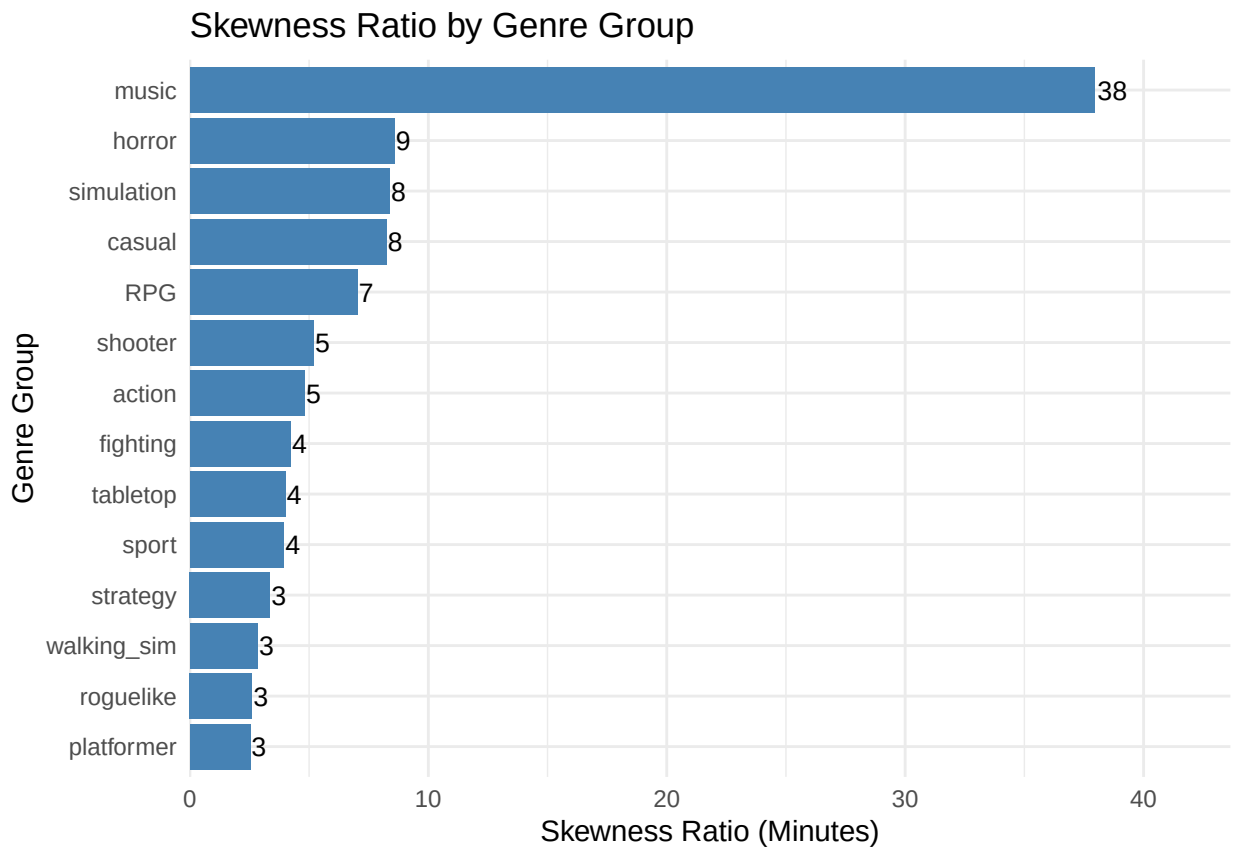



```
## # A tibble: 14 x 4
##    Genre_Group Mean_Playtime Median_Playtime Total_Games
##    <chr>               <dbl>           <dbl>       <int>
##  1 fighting            1300.             308         689
##  2 roguelike            781.             300         624
##  3 strategy             974.             290        2526
##  4 RPG                 2024.             288        2851
##  5 simulation          2071.             247        2292
##  6 action              1137.             236.       5062
##  7 shooter             1204.             232        1397
##  8 horror              1841.             214.       1238
##  9 platformer           517.             204.       1194
## 10 tabletop             784.             196        2033
## 11 casual              1513.             184.       3164
## 12 sport                680.             172.        700
## 13 walking_sim          446.             156.        340
## 14 music               5768.             152         196
```

```
## [1] 7981
```


Typical Playtime by Genre Group
Genre Group
fighting
roguelike
strategy
RPG
simulation
action
shooter
horror
platformer
tabletop
casual
sport
walking_sim
music
308
300
290
288
247
236
232
214
204
196
184
172
156
152
0
100
200
300
Median Playtime (Minutes)


Typical Playtime by Genre Group
Genre Group
music
simulation
RPG
horror
casual
fighting
shooter
action
strategy
tabletop
roguelike
sport
platformer
walking_sim
5768
2071
2024
1841
1513
1300
1204
1137
974
784
781
680
517
446
0
2000
4000
6000
Median Playtime (Minutes)



4 Commonalities among successful indie games

Indie game development is high-risk: budgets are limited, marketing reach is uncertain, and audience discovery can be unpredictable. This section explores commonalities across successful indie games, with two practical goals: * **Understand market audience:** identify what combinations of genres/mechanics tend to co-occur among games that reach larger audiences. * **Reduce risk for game-making:** extract patterns that can guide design decisions without prescribing a single "correct" formula.

We focus on four questions: * **Does genre matter?** * **Do mechanics matter (especially in combination with genre)?** * **Do game characteristics matter (e.g., camera, player modes, VR)?** * **Does pricing matter (relationship with owners)?**

4.1 Selecting indie games

Steam has done the hard work for us by including "Indie" as a genre/tag (and related tags such as "Crowdfunded" / "Kickstarter"). Given that these tags have been assigned by users world-wide, we can agree on these tags representing widely-considered indie games. We filter the dataset to games containing any of these signals in Genres, Tags or Categories.

4.2 Determining "successful" indie games

Determining "successful" indie games

"Success" is not directly labeled in the dataset, so we build an operational proxy based on market and engagement signals. The main goal of this step is to isolate a subset of indie games that consistently perform

better than the rest, so that later sections (association rules and plots) focus on patterns that appear among higher-impact titles.

4.2.1 Clustering approach

We cluster indie games using a set of numeric variables that capture outcomes and engagement (e.g., recommendations, reviews, playtime, peak CCU, owners/revenue estimates, price, and platform availability). Before fixing the final configuration, we experimented with multiple numbers of clusters: * **2 clusters** (attempting to represent successful vs not successful) * **3 clusters** (attempting to represent massively successful, successful and not successful) * **4 clusters** * **5 clusters** * **10 clusters**

The best segmentation for interpretability and separation was obtained with 3 clusters.

In our dataset, representative examples of these mid/high clusters included games such as Deceit, Graveyard Keeper, Unturned (mid-tier), and Stardew Valley, Subnautica, Terraria (top-tier). These examples illustrate how the clustering captures meaningful outcome tiers rather than arbitrary partitions.

We implement the final clustering using k-means with k = 3 on standardized variables.

Using this proxy, 37,791 games are labeled successful (57.8% of indie games).

4.2.2 Validation: do clusters separate meaningfully?

Because clustering is unsupervised, we validate quality using a simple but practical hypothesis: *If the most discriminant variable is unable to separate the clusters, then the clustering is likely not meaningful.* We therefore compute the variable that best differentiates the clusters (highest ANOVA F-statistic) and verify that it produces a clear separation across groups.

In our case, the most discriminant variable is pct_pos_total and it produces a clean ordering between the three clusters, which supports using this partition in the rest of the report.

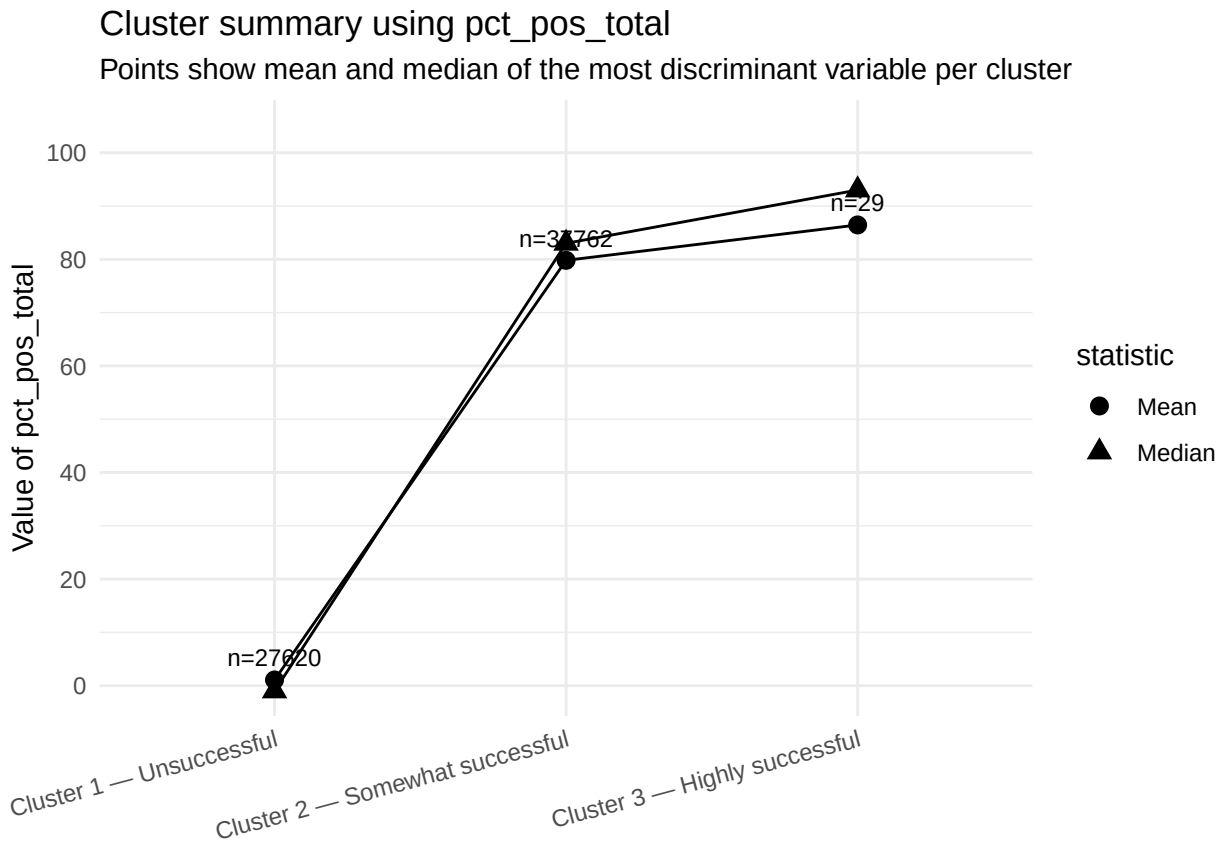
As an additional sanity check, we compute a global silhouette score on a random sample of games in the standardized feature space. Silhouette values closer to 1 indicate well-separated clusters, values near 0 indicate overlap, and negative values suggest poor assignments.

The average silhouette score on the sample is 0.404.

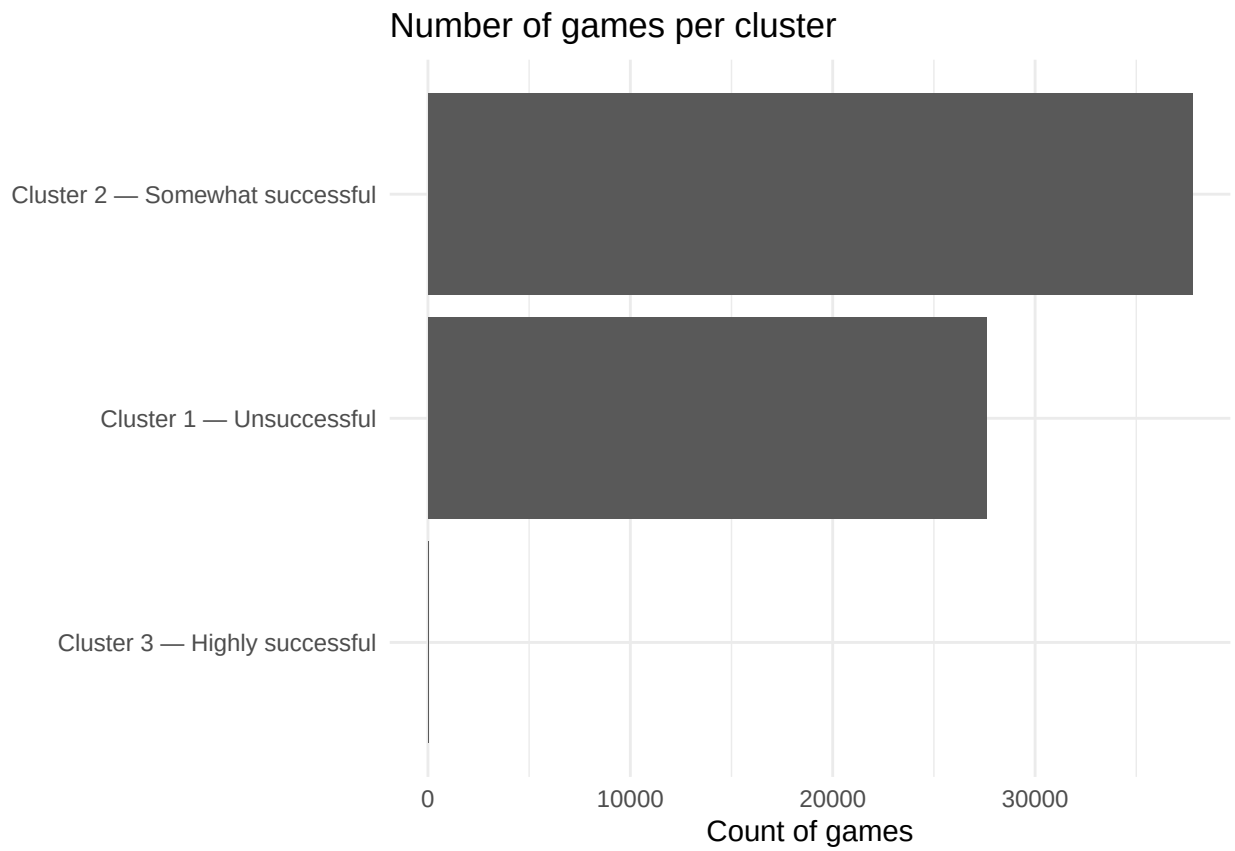
4.2.3 Cluster interpretation

The figure below summarizes the mean and median values of the discriminant variable (pct_pos_total) across clusters, alongside the number of games per tier.

In our run, the discriminant variable is pct_pos_total, which acts as a strong quality/visibility signal: the unsuccessful long-tail cluster concentrates near the lowest values, while the two top-tier cluster achieves the highest values, separating games with limited traction from games with stable success.



A key takeaway from the cluster sizes is that the distribution is highly imbalanced: a large share of games are concentrated in the lower-to-mid tiers, while the top tier contains only a small number of games. In this run, the lowest tier accounts for , whereas the highest tier represents only of indie games.



Since the clusters are interpretable and separable (both visually and by the discriminant variable test), we use them to define a successful subset and proceed with pattern mining in the next sections.

4.3 Genre study

Steam genre labels are sometimes inconsistent, so we use a grouped genre taxonomy created for this project (e.g., action, RPG, strategy, platformer, etc.). Each successful game is mapped into one or more of these grouped genres using its Genres/Tags/Categories. The Genres have been described in a previous section of the document.

Before adopting this approach, we initially attempted to mine association rules directly predicting success, by fixing the right-hand side of the rules to `successful = TRUE`.

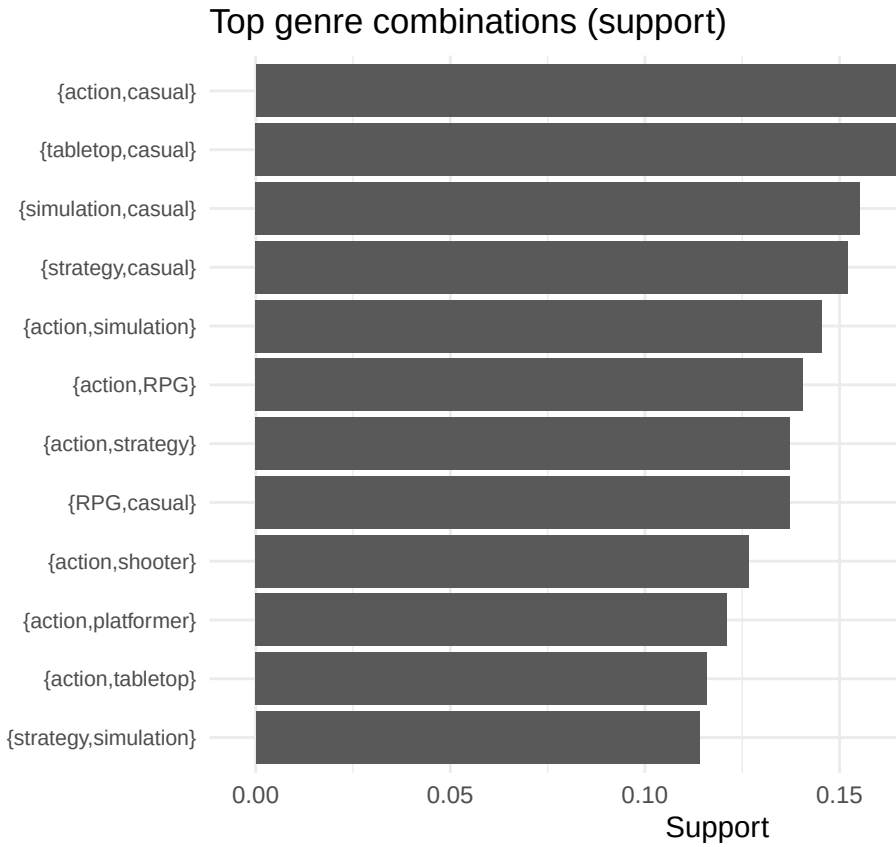
However, this produced obscure and highly specific combinations with limited generality: many rules were driven by niche tag mixes or small subsets of games, and they did not reflect the kinds of genre patterns seen in widely recognized successful indie titles.

For that reason, we switched to a more interpretable strategy: First filter the dataset to successful indie games, and then mine frequent itemsets (genre combinations) with sufficient support inside this subset.

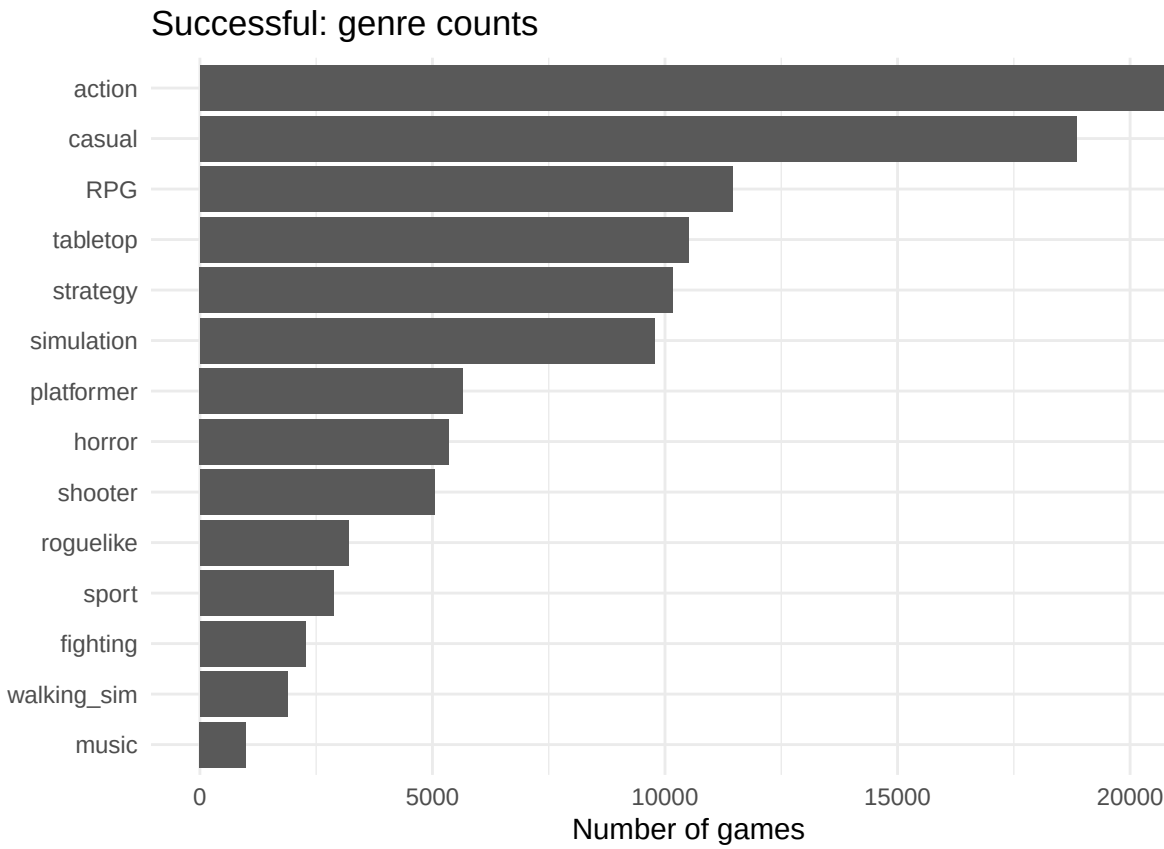
This approach still answers our question: *Amongng successful games, what genre combinations are most common?*

4.3.1 Most common genre combinations (frequent itemsets)

The table below lists the most frequent genre combinations among successful games (support = proportion of



successful games containing that combination).



Genre prevalence
Conclussions from genres The grouped genre counts show a highly concentrated distribution: successful indie games most frequently fall under action and casual, followed by a second tier including RPG, tabletop/puzzle, strategy, and simulation. This suggests that, within the successful segment, many games belong to genres that are either broad and audience-friendly (action/casual), or built around deep progression and replayability (RPG/strategy/simulation/tabletop).

Looking at frequent genre combinations (Eclat), the strongest result is the pairing {action, casual}, which appears in roughly one quarter of successful games. More generally, casual acts as a "bridge" genre: it appears in many of the top combinations (e.g., tabletop + casual, simulation + casual, strategy + casual, and RPG + casual). This indicates that many successful indie games mix a core genre identity with accessible design traits (short sessions or low entry difficulty).

The second major pattern is that action combines well with several popular genres: action + RPG, action + simulation, action + strategy, action + shooter, and action + platformer all appear as frequent itemsets. This highlights a typical indie design strategy: start from a strong core action loop and enrich it with complementary systems such as progression (RPG), management (simulation/strategy), or movement challenges (platformer).

4.4 Mechanics study

Mechanics are stored as list of tags (e.g., resource_management, procedural, narrative, card/deckbuilding, etc.) in the steam dataset so they are first one-hot encoded in a binary manner so the apriori algorithm can be executed over them. Instead of only asking "which mechanics are common?", we also ask: *Given a genre combination, which mechanics are most strongly associated with it?*

We mine association rules of the form: * LHS (antecedent): genre group(s) * RHS (consequent): mechanic group

Rules are ranked by lift to highlight mechanics that occur more often than expected within a genre context. The support is set to `0.01` and confidence to `0.3`. Higher values fail to find interesting relationships in the

data.

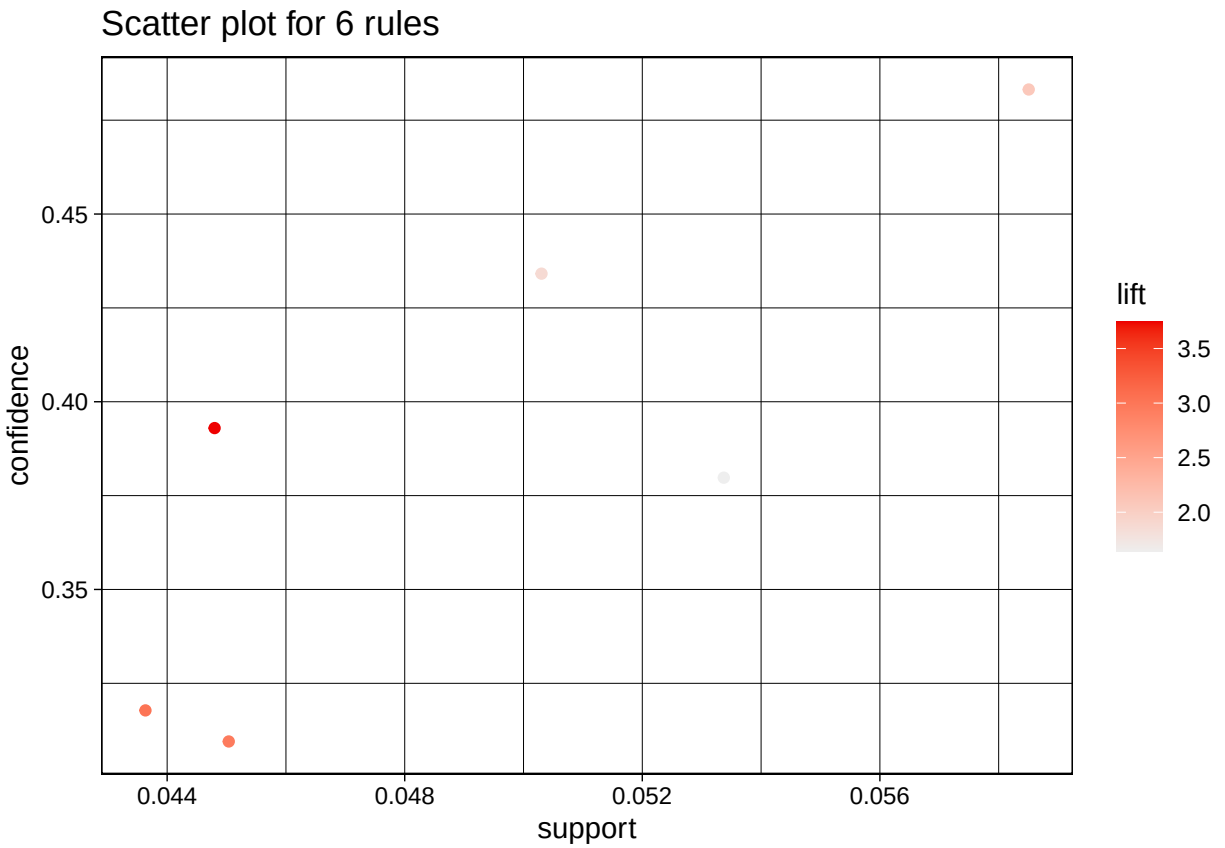


Table 1: Top genre->mechanic rules among successful indie games (ranked by lift)

Rule (Genres -> Mechanic)	Support	Confidence	Lift
{simulation,strategy} => {resource_management}	0.045	0.393	3.746
{action,strategy} => {resource_management}	0.044	0.318	3.029
{action,simulation} => {resource_management}	0.045	0.310	2.950
{action,platformer} => {exploration}	0.059	0.483	2.088
{action,tabletop} => {exploration}	0.050	0.434	1.876
{action,RPG} => {exploration}	0.053	0.380	1.641

4.4.1 Conclussions from genres and mechanic rules

A first clear pattern is the strength of resource management inside strategy/simulation hybrids. The rule {simulation, strategy} -> {resource_management} has the highest lift (around 3.75), meaning that management-oriented mechanics are several times more likely than expected when a game sits at the intersection of these genres. Similar high-lift rules also appear for {action, strategy} and {action, simulation}, suggesting that successful hybrids often combine an action layer with systems such as crafting, automation, building, or economy loops.

A second pattern is that exploration frequently complements action-driven genres. Rules such as {action, platformer} -> {exploration}, {action, tabletop} -> {exploration}, and {action, RPG} -> {exploration} show relatively high confidence, indicating that when successful games blend action with movement/progression,

they often reinforce the experience through discovery loops (new areas, loot, dungeons, collectables, or open-ended progression paths).

Overall, the results support a practical conclusion: successful indie games often rely on a core genre identity and then amplify engagement through a matching mechanic; management systems for strategy/simulation hybrids, and exploration loops for action-driven combinations. These associations do not prove causality, but they highlight combinations that repeatedly co-occur among successful titles and can reduce design risk by aligning with common audience expectations.

4.5 Game characteristics study

Characteristics are captured from Steam categories/tags that describe presentation and play modes (e.g., 2D/3D, first-person/third-person, singleplayer/co-op, VR). We mine frequent characteristic combinations using Eclat.

4.5.1 Top characteristic combinations

Table 2: Top characteristic combinations among successful indie games

	Characteristic combination	Support
7	{Co-op,PvP}	0.0529
5	{Massively Multiplayer,PvP}	0.0165
6	{Massively Multiplayer,Co-op}	0.0117
4	{Massively Multiplayer,Co-op,PvP}	0.0092
2	{VR Only,PvP}	0.0055
3	{VR Only,Co-op}	0.0044
1	{VR Only,Co-op,PvP}	0.0026

4.5.2 Conclusions from characteristics

The most frequent characteristic combinations in the successful subset are primarily multiplayer-focused, especially the pairing {Co-op, PvP} (support ~ 0.053). This suggests that, among successful indie games that include multiplayer features, a common design choice is to combine collaboration and competition within the same title.

We also observe VR Only appearing in the top combinations but with low support, suggesting that VR-exclusive successful games exist but represent a niche segment compared with traditional PC titles. A plausible explanation is that the VR market offers fewer alternatives overall, so the relatively small number of VR-only titles can capture a larger share of VR players and reach the engagement thresholds needed to be labeled as successful in our proxy, which makes VR-only features show up among the top itemsets.

Overall, characteristics appear to matter mainly as experience modifiers: multiplayer modes (co-op/PvP) are recurring patterns among successful games, while more specialized formats (like VR-only) are less common.

##Pricing and owners Finally, we explore whether price is related to estimated owners. We initially attempted to approach this with a simple scatter plot which ended up hard to interpret due to heavy overlap and the long-tail nature of owners. Instead the best analysis technique has been running a multivariate regression on non-free games and studying the weights.

4.5.3 Multivariate regression

A raw relationship between price and owners can be misleading because both variables are correlated with other factors. To reduce this bias, we estimate a multivariate regression on paid games only (price > 0), controlling for: * **Review volume** (addittional approximate for audience size) * **Rating quality** (proxy

for perceived value) * **Recommendations** (proxy for engagement) * **DLC count and language count** (rough indicators of production scope)

The coefficient of log-price can be interpreted as the partial association between price and owners after accounting for these variables.

The price has been capped at 90 euros and games above that price have been removed since those aren't realistic representations of indie games.

Table 3: Multivariate regression coefficients (paid games only)

Variable	Estimate	Std. Error	t value	p value
log_price	-0.1027	0.0097	-10.55	0.00000
log_reviews	0.6396	0.0107	59.66	0.00000
pct_pos_total	-0.0058	0.0004	-13.96	0.00000
log_recs	-0.0155	0.0059	-2.62	0.00877
dlc_count	0.0003	0.0004	0.83	0.40600
languages_count	-0.0400	0.0005	-79.03	0.00000

Table 4: Variance Inflation Factors (VIF) for multicollinearity diagnosis

Variable	VIF
log_reviews	6.79
log_recs	6.69
log_price	1.12
pct_pos_total	1.04
languages_count	1.02
dlc_count	1.00

Table 5: Standardized coefficients

Variable	Std_Estimate	Std. Error	t value	p value
log_reviews	0.648	0.011	59.66	0.00000
languages_count	-0.332	0.004	-79.03	0.00000
pct_pos_total	-0.059	0.004	-13.96	0.00000
log_price	-0.047	0.004	-10.55	0.00000
log_recs	-0.028	0.011	-2.62	0.00877
dlc_count	0.003	0.004	0.83	0.40600

The VIF table provides an explicit check for multicollinearity. When VIF values are elevated, predictors overlap strongly (for example, review volume and recommendations both reflect visibility and player engagement). In that situation, individual coefficients should be interpreted cautiously: their signs and magnitudes can shift because the model is separating very similar signals.

Multicollinearity does not affect the interpretation of price, since log_price has VIF 1.12. Therefore, the negative coefficient of price can be interpreted as a stable partial relationship: higher prices are associated with slightly fewer owners, although the magnitude is small compared to visibility signals such as review volume.

After controlling for closely related visibility/quality variables, price has at most a small partial association with owners. This suggests that in the indie market, audience size is driven more by discoverability and perceived value (captured by reviews and engagement signals) than by price alone.

4.6 Conclusion

We grouped indie games into three success tiers using clustering, and the separation between tiers was driven most strongly by overall audience approval (pct_pos_total). The distribution is highly uneven: most indie games fall into the unsuccessful or somewhat successful tiers, while the highly successful tier is rare, reinforcing the idea that standout success is uncommon and risk is structurally high in the indie market.

Does genre matter? Successful games are heavily concentrated in a few genre groups, dominated by Action and Casual, with the most common combination being {action, casual}. Many other frequent pairs include casual as the "bridge" (e.g., tabletop/simulation/strategy + casual), suggesting that successful indies often mix a clear genre identity with accessible play patterns, while still adding depth through RPG/strategy/simulation-style progression.

Do mechanics matter? The genre->mechanic rules show consistent associations: resource management is strongly linked to simulation/strategy (and action hybrids), while exploration repeatedly complements action combinations (platformer, tabletop, RPG). This supports the idea that successful games often align mechanics with what players expect from that genre blend.

Do game characteristics matter? The strongest recurring characteristic pattern is multiplayer design that combines Co-op + PvP, while VR-only appears but with low support, suggesting it's a niche path rather than a mainstream success driver.

Does pricing matter? Only weakly compared to visibility and engagement signals. In the paid-only regression, price has a small negative association with owners, and its standardized effect is much smaller than review volume, which is by far the strongest correlate of audience size. Multicollinearity mainly affects review/recommendation signals (VIF ~ 6–7), but price is stable (low VIF) and still relatively minor.

Overall, these results are correlational, not causal—they don't prove that picking a genre or adding a mechanic causes success. However, they provide practical, evidence-based guidance to understand the market audience and reduce risk, helping us make more informed game-design decisions around genre direction, mechanic fit, feature scope, and pricing expectations.

5 Can a single game have enough influence to make other games have its tag?

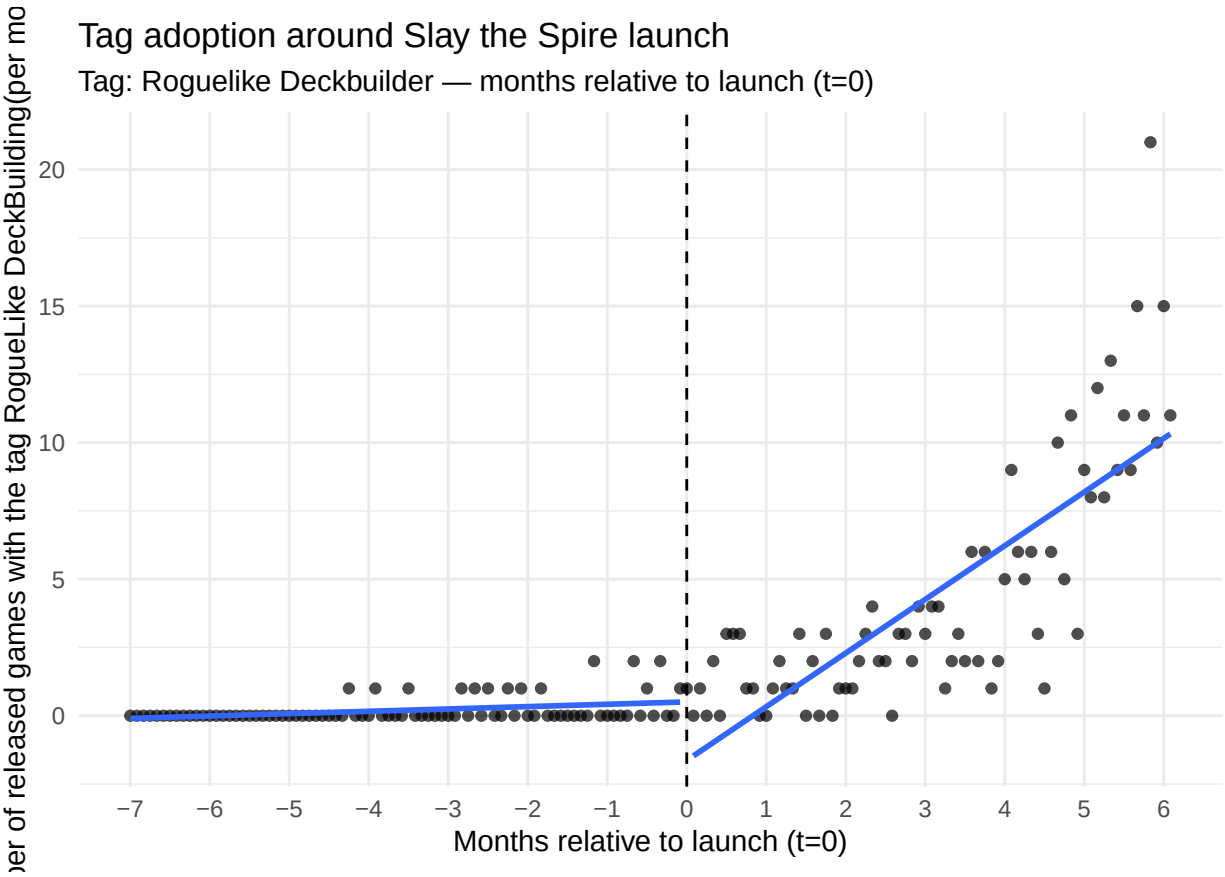
5.1 Game study: Slay the Spire (Roguelike Deckbuilder)

```
## [1] "Slay the Spire"
```

```
## [1] "2019-01-23"
```

```
##
## Call:
## lm(formula = freq ~ t + post + t:post, data = ts_slayTheSpire)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2087 -0.4368 -0.1021  0.5213 11.1710
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.529412   0.403337   1.313 0.191278
## t            0.007563   0.008292   0.912 0.363151
## post        -2.164115   0.599569  -3.609 0.000414 ***
## t:post       0.156203   0.013316  11.731  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

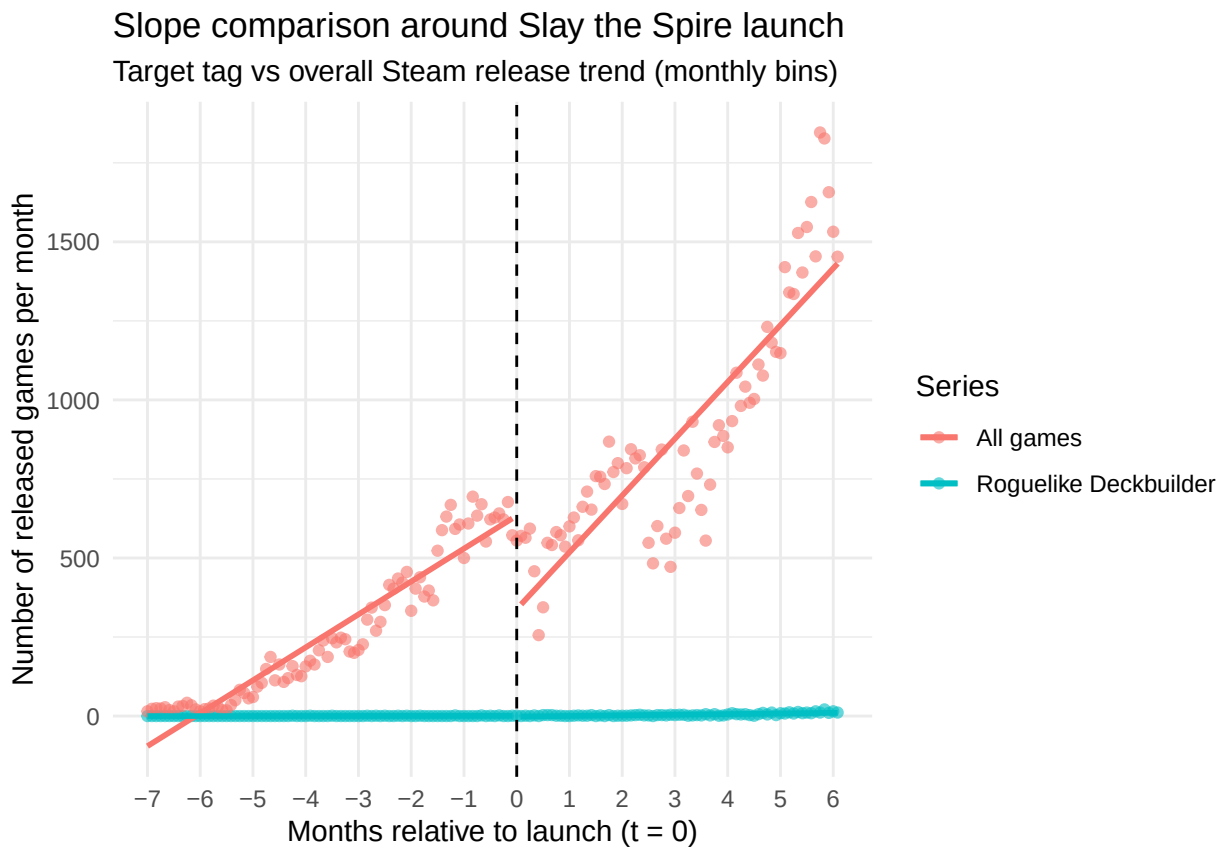
```
##
## Residual standard error: 1.876 on 154 degrees of freedom
## Multiple R-squared:  0.7433, Adjusted R-squared:  0.7383
## F-statistic: 148.7 on 3 and 154 DF,  p-value: < 2.2e-16
```




```
##
## Call:
## lm(formula = freq ~ t + post + t:post, data = ts_slayTheSpire)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2087 -0.4368 -0.1021  0.5213 11.1710
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.529412   0.403337   1.313 0.191278
## t            0.007563   0.008292   0.912 0.363151
## post        -2.164115   0.599569  -3.609 0.000414 ***
## t:post       0.156203   0.013316  11.731  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.876 on 154 degrees of freedom
## Multiple R-squared:  0.7433, Adjusted R-squared:  0.7383
## F-statistic: 148.7 on 3 and 154 DF,  p-value: < 2.2e-16
```

```
##
```

```
## Call:
## lm(formula = freq ~ t + post + t:post, data = ts_global)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -427.10  -66.85   -1.87   81.84  474.64
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  630.1863    28.6368  22.006  < 2e-16 ***
## t              8.6137     0.5887  14.631  < 2e-16 ***
## post        -291.8553    42.5691  -6.856 1.61e-10 ***
## t:post         6.3578     0.9454   6.725 3.24e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 133.2 on 154 degrees of freedom
## Multiple R-squared:  0.9061, Adjusted R-squared:  0.9042
## F-statistic: 495.2 on 3 and 154 DF,  p-value: < 2.2e-16
```




```
##
## Call:
## lm(formula = freq ~ t * post * series, data = combined)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -427.10   -4.05   -0.12    4.33  474.64
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       630.1863    20.2513  31.118  < 2e-16 ***
## t                   8.6137     0.4163  20.689  < 2e-16 ***
## post             -291.8553    30.1039  -9.695  < 2e-16 ***
## seriestag        -629.6569    28.6396 -21.986  < 2e-16 ***
## t:post              6.3578     0.6686   9.509  < 2e-16 ***
## t:seriestag        -8.6061     0.5888 -14.617  < 2e-16 ***
## post:seriestag    289.6912    42.5733   6.805 5.29e-11 ***
## t:post:seriestag   -6.2015     0.9455  -6.559 2.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 94.18 on 308 degrees of freedom
## Multiple R-squared:  0.9488, Adjusted R-squared:  0.9476
## F-statistic: 815.6 on 7 and 308 DF,  p-value: < 2.2e-16
```

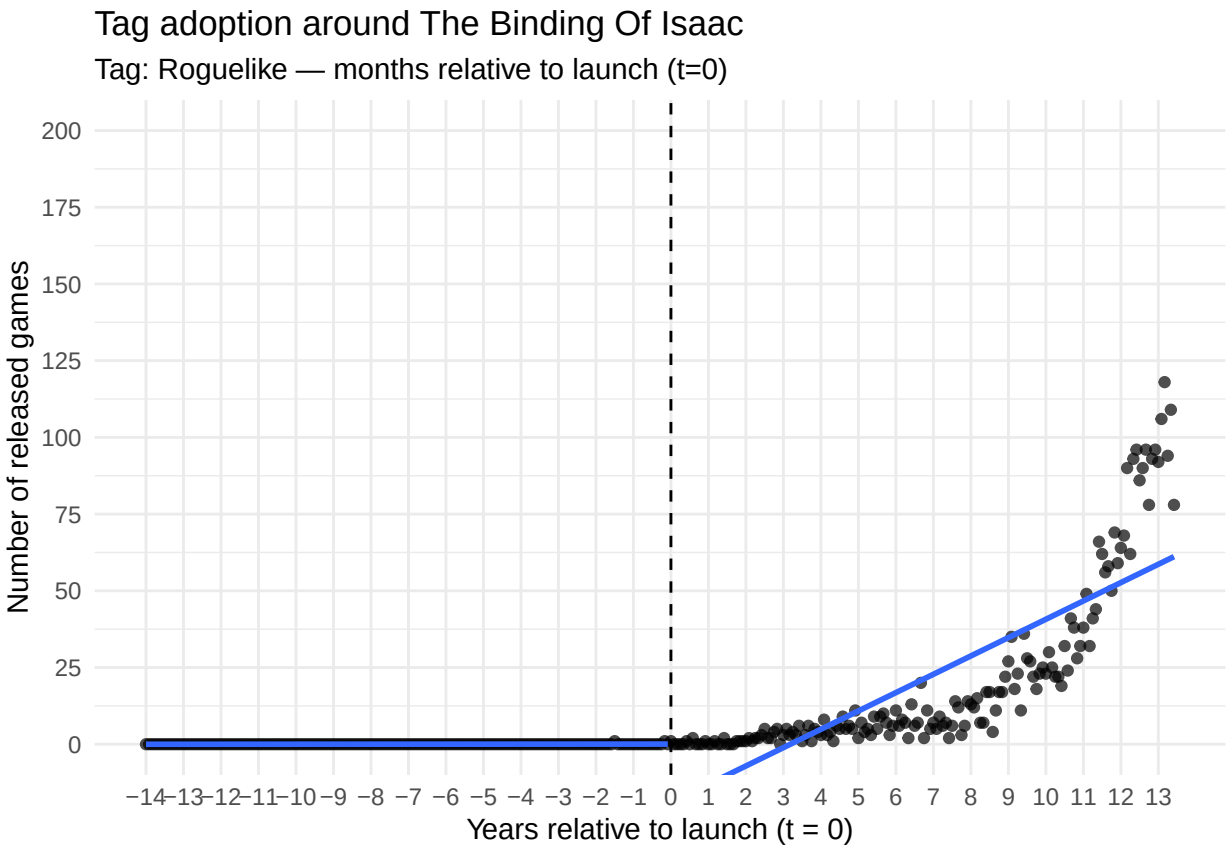
5.2 Game Study: The Binding Of Isaac + The Binding Of Isaac Rebirth (Roguelike)

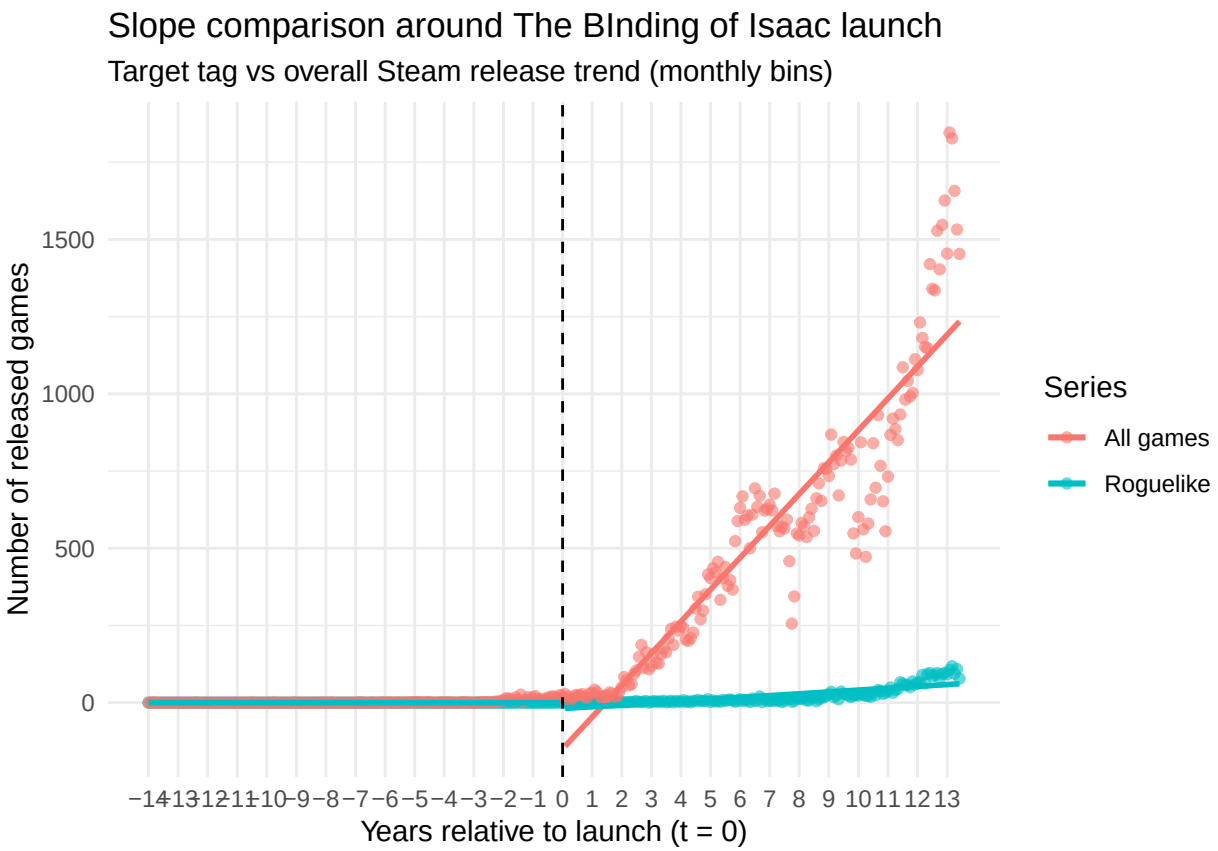
```
## [1] "The Binding of Isaac"
```

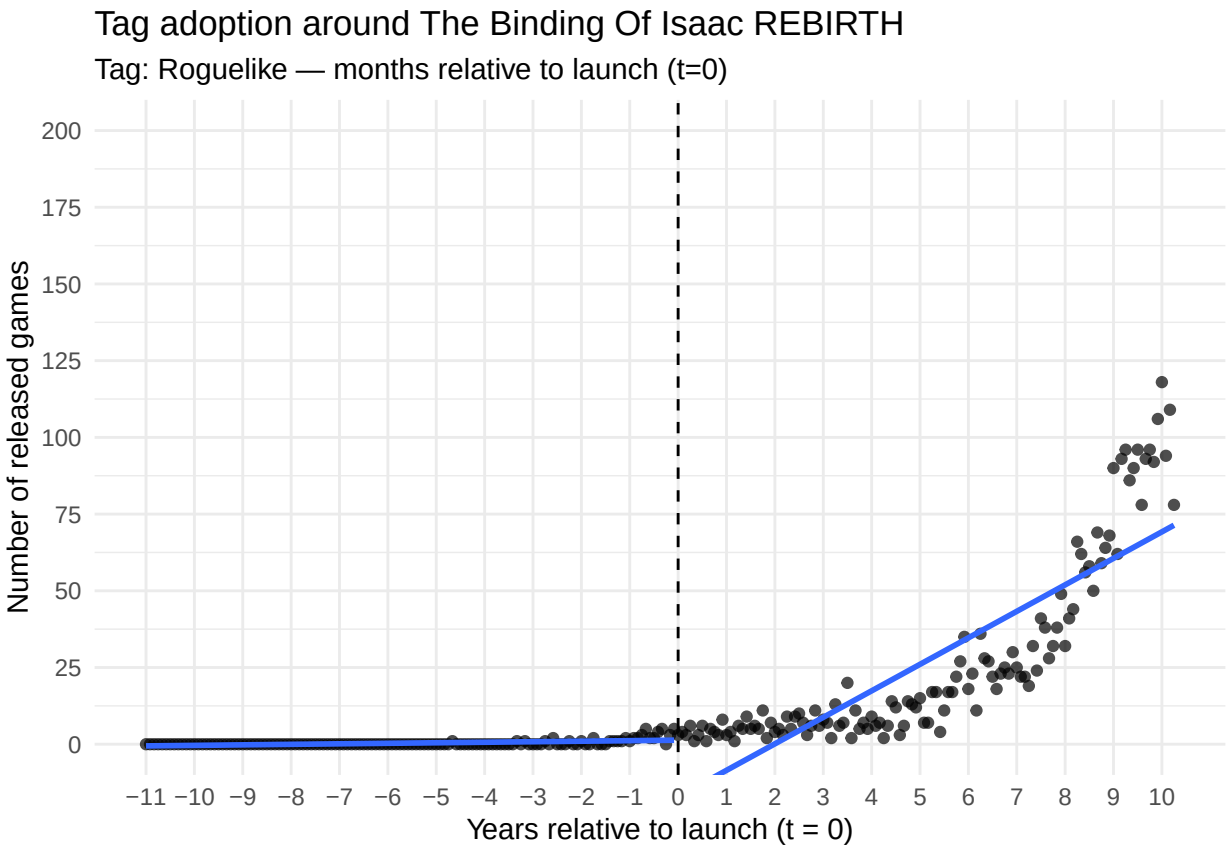
```
## [1] "2011-09-28"
```

```
## [1] "The Binding of Isaac: Rebirth"
```

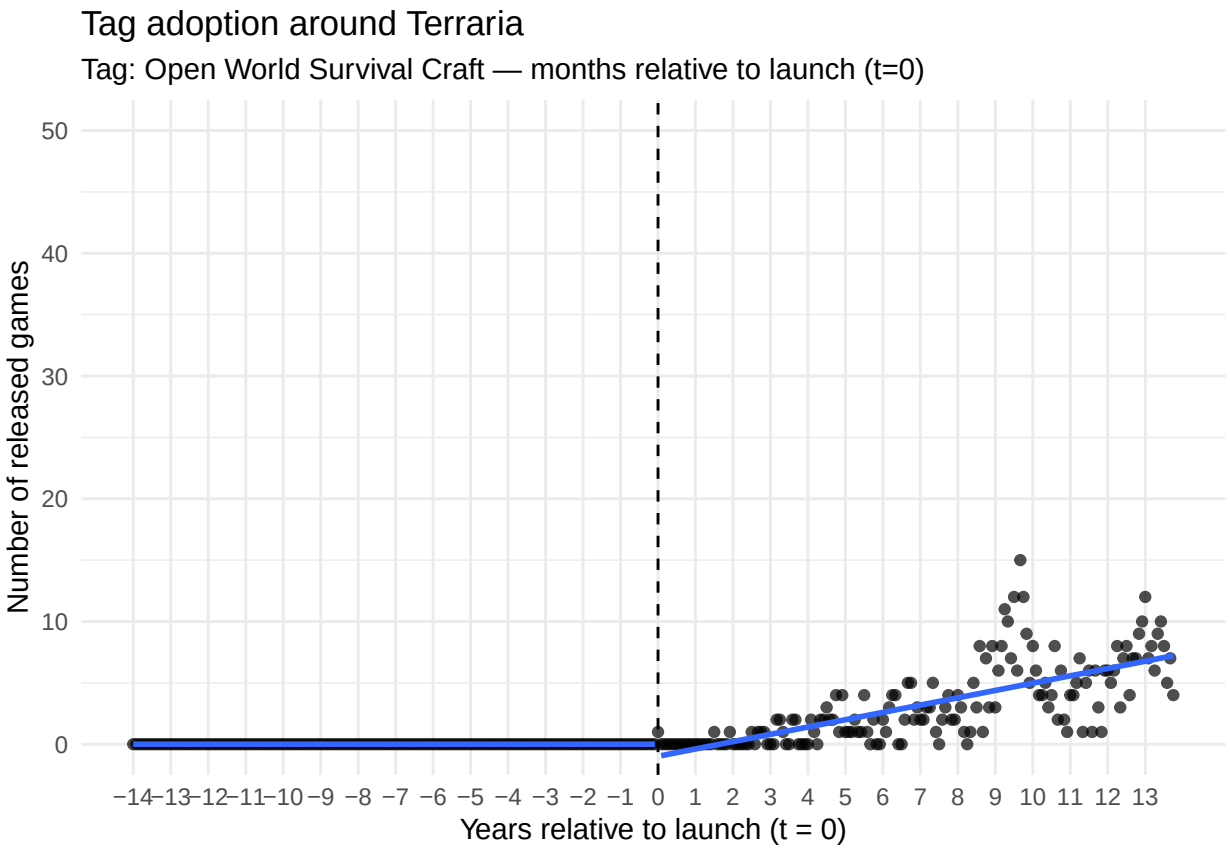
```
## [1] "2014-11-04"
```


Tag adoption around The Binding Of Isaac
Tag: Roguelike — months relative to launch (t=0)
Number of released games
200
175
150
125
100
75
50
25
0
−14 −13 −12 −11 −10 −9 −8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8 9 10 11 12 13
Years relative to launch (t = 0)


Slope comparison around The BInding of Isaac launch
Target tag vs overall Steam release trend (monthly bins)
Number of released games
1500
1000
500
0
−14 −13 −12 −11 −10 −9 −8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8 9 10 11 12 13
Years relative to launch (t = 0)
Series
All games
Roguelike


Tag adoption around The Binding Of Isaac REBIRTH
Tag: Roguelike — months relative to launch (t=0)
Number of released games
200
175
150
125
100
75
50
25
0
−11 −10 −9 −8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8 9 10
Years relative to launch (t = 0)


```
## [1] "Terraria"
## [1] "2011-05-16"
```


Tag adoption around Terraria
Tag: Open World Survival Craft — months relative to launch (t=0)
Number of released games
50
40
30
20
10
0
−14 −13 −12 −11 −10 −9 −8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8 9 10 11 12 13
Years relative to launch (t = 0)