UPPSALA
UNIVERSITET

Know Your Data - Steam: Under the Scope

A data mining project analyzing SteamSpy.

Project
Data mining and data science 7.5 credits
(21S063)

Group 12
Noah Johansson
Zackarias Koraish
Viktor Källberg
Tobias Källman Andersson
Olivia Lustig Lindström

Abstract

In this paper we developed a fictional story involving a struggling game-developing company, WHY Games. WHY Games latest releases have been unsuccessful and are now focused on recovery, aiming to produce a game that will turn their negative trajectory around. Enter KYD, Know Your Data. KYD is a data analytics company specialized in processing raw data and extracting value from it. By generating and exploring data from SteamSpy KYD aims to produce a model for measuring popularity. Subsequently, the popularity metric will further be used as a threshold for filtering out non-relevant games. Games passing the threshold will be used as input for methods intending to explore patterns and associations. The project's goal culminates in a set of successful games intended to serve as a source of inspiration for WHY Games in their pursuit of success.

1. Background/description of the case/organization

WHY Games is an organisation aspiring to conquer the game-market for the up and coming generation of future gamers. In the past, WHY Games has experienced success with several of their previous projects. They've explored different genres, ranging from action packed first-person shooter games to puzzlers and party games. However, WHY Games' former glory days are long gone and their chief executive officer recently announced that the former industry-behemoth is betting everything on their next game release. If it fails, WHY Games' future is uncertain to say the least.

WHY Games' philosophy stands upon the pillars of creative license for the content development team. However, as their once upon a time highly valued recipe for success doesn't yield the result it used to produce, they're in dire need of help to improve the development strategy.

In 2017 the global revenue in the video game industry was estimated to be $116 billion US dollars, and grew 10.7% from 2016 to 2017 (Huang et al., 2019). The industry shows no signs of regression, and we can safely assume that it will continue to grow during the years to come.

WHY Games' myopia in an ever-shifting gaming industry market could be the deciding factor for their failing business model, the features distinguishing an unsuccessful game to a successful game is not straightforward. WHY Games' has acknowledged that shortcomings exist in their development strategy. Their philosophy encouraged creative license but could potentially result in few monetization options to generate profits.

As mentioned earlier WHY Games' future depends on the success of their next game. And in regards to the aforementioned reasons, they hired us. Know Your Data (from now on referred to as KYD), a data analytics consulting firm specialized in understanding large quantities of data, discovering patterns, and making predictions based on findings in the data.

WHY Games vaguely described our assignment as to understand why certain games become popular and discover which features and attributes that are highly valued by the gaming community. As social-media has exploded during the second half of the decade, previous projects we have worked on indicate that everything now circuits around trends. It's been a long time since GTA3 was released, and games today have to be edgy, funny, interesting, and high-performing in the social-media sphere.

Therefore, we expect the contributing factors to a successful game differ from merely ten years ago.

We set out to aid WHY Games' in strengthening their understanding of the gaming market. Before they can establish the foundations for the game that hopefully will turn their negative trend around, some questions require an answer . Which categories are popular today, what financial model is likely to increase our success rate and which characteristics are shared by popular games?

WHY Games' decided that they will release their game on the computer game distribution platform Steam. (https://store.steampowered.com/about/). Therefore, in order to produce an answer to the aforementioned questions, digging into Steam and exploring their data is necessary.

SteamSpy is a platform containing data on games available on the Steam platform. The platform provides data from Steam users in such a way that it can be used to analyse the use of games available. (https://steamspy.com/about).

2. Problem description and research questions

2.1 Problem description

WHY Games want us to deliver a summary of successful games, their characteristics, and interesting patterns in their underlying features. In order to do this, there has to be a clear cut definition of what a ***popular*** game is. Subsequently we must cull out the relevant games from the dataset by the earlier defined popular metric. Thus, the success of our work is strongly dependent on the popularity metric and therefore it has to be defined delicately. How can we objectively value what a popular game is, and how to generalize attributes impact on the popularity metric?

Popularity is subjective and thus something that does not have a clear definition. We therefore have an understanding that not everyone will have the same view of popularity as the definition we have identified and that there is possible room for improvement.

In an effort to concretize exactly what constitutes a game as successful, we have identified two key factors to evaluate in the data. These two attributes have each been given a scoring rank and when combined, these will determine whether or not we consider a game to be successful.

The metric considered the most important is the number of owners. The reason for this is because there are likely games that could be considered successful despite having a relatively low user score and a low number of concurrent users. From a developer standpoint a game with a cost that has a high owner count would likely be considered successful even if concurrent users and user score is extremely low as revenue has been generated through sales.

The second attribute to consider is the user score. The ratings that players themselves give a game is interesting as they are the ones actually consuming the products and purchasing or playing the games. The opinion of customers is relevant to any business and developers can adjust video games according to what users prefer. Measuring success from a user score standpoint does have its flaws however, as several things can affect whether or not a player will rate a game positively or negatively. We can

imagine that server quality can impact the ratings of multiplayer games greatly, even though it does not say much about the quality of gameplay.

A third and final attribute was also considered as a metric for success, namely the number of concurrent users. However, it was ultimately decided against, as it could be a poor way of measuring success, especially in single player games. Single player story based games that were released several years ago have most likely been played to completion by most owners, and will therefore have a low CCU value. This does not necessarily mean that the game is unsuccessful, as there may have been plenty of revenue generated from sales and a high number of concurrent users in the past.

With these two attributes we have decided to rank a game with a popularity metric. Games above a specific threshold will be considered successful, and the attributes of these successful games will be analysed to find trends and patterns. With this knowledge, KYD will hopefully be able to identify exactly what makes games successful, and as such be able to develop the massive video game titles of the future.

2.2 Aim and research questions

By collecting data from the Steam API and analyzing it, KYD aims to get a better understanding regarding what is highly valued by the target group and identify key success factors in video game development. The analysis will be used as a basis for WHY games to decide what to focus on when developing new games and its features.

This project aims to answer the following questions:

1. Which are the key factors to a successful game according to the data?
2. What are the most common tags of popular games?

3. Previous studies and related research

A study by Zendle et al. (2020) explored the way game developers generate revenue and how video game monetisation has shifted significantly in the last decade. By enabling purchases of in-game cosmetics and loot boxes through microtransactions, developers can earn revenue in a new way and not have to rely entirely on selling actual game copies. The study found that over 80% of the 463 most played games on Steam had enabled the use of cosmetic microtransactions and that these features are the most trending business model in the studied period (2010-2019).

According to Bailey, E. N., & Miyata, K. (2018) multiplayer games tend to increase sales. A detailed study was conducted where 19,621 games were obtained from Steam using its public API. The games were then analyzed based on whether the game was labeled as “multiplayer” or ”single-player” as well as Metacritic score and rating on steam. The study found a correlation between games with the “multiplayer” tag and a higher number of sales. The study also found that games with multiplayer functionality had a higher median rating according to metacritic, but a lower median user rating on Steam. Finally, the authors conclude the study with the observation that developing games with

multiplayer functionality seems to have a positive effect on the number of sales. However, the authors advise against developing multiplayer features unless there is confidence that the multiplayer mode will be capable of maintaining a strong audience, as it might just be an unnecessary cost and risk during development.

The study described above will be of interest during this project as we intend to investigate what factors are key in developing a successful game in terms of number of sales as well as ratings. One factor that will be analysed is whether the genres of a game have an effect on that game's sales and ratings. As such, we will be able to compare results from our report and the study made by Bailey, E. N., & Miyata, K. (2018) and determine whether or not multiplayer functionality is a key factor to developing a successful game.

4. CRISP-DM vs DST

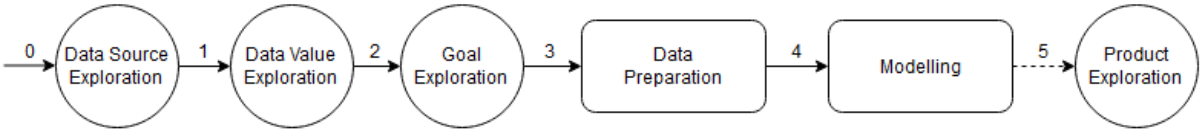


Figure 2: DST, planned trajectory for our project.

CRISP-DM (CRoss-industry Standard Process for Data Mining) is an analytic “methodology” used to categorize and structure each step-in data mining projects. CRISP-DM was released 1996 and although the internet has transformed since then and is now integrated in every aspect of our lives, CRISP-DM has maintained its position as the industry standard model for data mining. However, due to the fact that data collection has revolutionized throughout the 21st century the CRISP-DM model lags behind in recent technological progress. The DST model intends to account for the fallacies in the CRISP-DM model (Martinez-Plumed et al., 2021). Martínez-Plumed et al (2021) has further identified some differences with data mining twenty years ago compared to today and based on this, the authors highlight some weaknesses with CRISP-DM and propose an extension on the methodology that aims to complement and reduce the identified weaknesses. The extended model presented is called DST and stands for Data Science Trajectories. One of the big differences identified by the authors is that CRISP-DM sees data as an important part of achieving its goal but believes that the process has the greatest focus, while DST has more of a data-oriented approach where the data itself is the main focus. Based on this, we have chosen to use DST for our project as we believe that this model will contribute more to our purpose. (Martínez-Plumed et al. 2021).

4.1. Data Source Exploration

The first step, Data Source Exploration, is mainly about exploring different options of data sources and selecting which one will serve the purpose and bring value to the customer WHY Games. WHY

Games specifically gave KYD instructions to explore the data distribution service Steam. Therefore, KYD decided to use the data source SteamSpy which is relevant and accessible for the given cause.

4.1.1. Data Source

Steam, which is "the ultimate online game platform"(https://store.steampowered.com), is a very good source of gaming trends and what games are popular right now, mainly for the PC market. Steam also provides their own API for smooth and easy access to their public data. Steams API provides access to the relevant data. Using the DST, seen above in figure 1, which increases the chances of producing a relevant data set and thus acquiring knowledge of what makes a successful game. A shortcoming with Steams API is that every request has to have very specific information about for example from which game we want data from, this makes it a hassle to compile trends directly from their API. A solution to this problem is provided by SteamSpy. SteamSpy(https://steamspy.com) is a service that provides collected game data statistics from the Web Steam API. SteamSpy structures the data in a way that allows us to gather information on all games, instead of a particular one. However, there are downsides to using SteamSpy. These are that data is statistically collected, meaning that altough, for example the number of owners is generated with an 98% statistical confidence, it could not reflect actual reality and contain some invalid data. SteamSpy also is very up to date with its figures, meaning that the date which we fetch the data has an impact on our findings, which is actually what we want since popularity and trends develop and change over time. Since we collect, clean and present our findings via Jupyter, we will use the PyPi library SteamspyPyPi (https://pypi.org/project/steamspypi/). SteamSpyPy has 44 "pages" of data as Alternatively, if you know the exact number of pages, e.g. 44 as of March 29, 2021.

4.2. Data Value Exploration

Given the available data from the first collection(get_all_pages) we can't really do anything except present what games are available and their 'appid' attribute. This is because a lot of the available attribute data doesn't fetch with this general API-call, which means a lot of games contain null-data. Hence the further API-calls to get details about each game via 'appid'. Attributes within the data that might have some value for analysing is:

1. appid
2. name
3. developer
4. publisher
5. average_forever
6. average_2weeks
7. median_forever
8. median_2weeks
9. ccu
10. price
11. initialprice
12. tags
13. languages
14. positive
15. negative

These attributes were chosen because they can all be compared with each other and have some inherent value together with domain knowledge about games.

4.3. Goal Exploration

To align our business goals together with the particular project's goal, we looked at several different ways in which the data would contribute to the client. Initially, we've had to quantify success in order to measure it. Needless to say, this was not a simple task. KYD provides data analytics solutions to businesses in need of data understanding to aid them achieving their goals. A recurring task that occurs in our process is therefore to define how patterns found in the data will aid the client.

4.4. Data cleaning process

Cleaning data is handled in CleanData.ipynb. Once we have all data in a DataFrame we remove games without any price information available(i.e. null data) and also if there is no language data available(i.e. null data). All columns are then transformed to their expected type and columns that will not be used are removed('score_rank', 'discount', 'genre'). The price column is converted from cents to dollar, for a more human readable format. The column 'userscore' is filled with quotients (positive / negative), which creates a percentage of users who found the game a positive experience.

4.5. Data Preparation

Data preparation is handled in PresentData.ipynb. The preparation of data is an extension of our cleaning, but more specific to what the data will actually be used for. Unnecessary columns for the analysis are removed and new columns are created.

The owner's column is transformed from a categorical interval to a usable integer in the form of the midpoint(average) on the interval.

A new column, popularity, is created by taking the userscore(percentage of a positive experience according to users) and multiplying with the average number of owners of that game, which is then normalized by taking the square root of the result. This is the measure of popularity of a game.

With domain knowledge of games available in the Steam library, there are approximately 2000 games that can be considered popular at any given time. With this knowledge a new column is created, is_popular, which takes the 2000 most popular games and marks them as popular.

Via previously exploration of our data we know that the most popular tags, excluding the tag 'indie', are 'action', 'singleplayer', 'multiplayer', 'adventure', 'co-op', 'rpg', 'strategy', 'open_world', 'atmospheric', 'shooter', 'horror', 'survival' and 'simulation'. With this knowledge a new column is created for each game that denotes if the game has the specific tag attached to it.

The last preparation of the data is done by creating a new categorical attribute 'price_range'. This gives each game a label based on its current price, 'free', 'cheap', 'normal' and 'expensive'.

4.6. Modelling

With domain knowledge we make the assumption that there are approximately 2000 popular games at any given time. To produce a model, manual clustering of the 2000 most popular games using the popularity index mentioned above in 4.5 was required. Furthermore association rule mining was used to analyze the different tags and their association. And lastly we used wordcloud, barcharts and boxplots to visualize the data.

4.7. Product Exploration

KYD's purpose is not to produce a set of attributes that compose a recipe for successful games, but rather a collection of the most popular games according to our self-defined metrics.The product can therefore be viewed as a set of popular games that can further be explored by WHY games to inspire which direction they're taking in their development project.

Our first step, when using our DST illustrated in figure 1, is to explore our chosen data source(API). The second step is to choose the data which will give our project relevant data to use. We will then formulate a goal with our chosen data as a base, i.e. what we want to use the data for. Then we fetch and prepare the data from our source. After that, we will create our model systems as planned, present our findings and failures/successes. As the last step, we will present possible products/services our models can be used for.

5. Data collection process and the properties of the data

5.1 Data Collection

All data used in this project is collected, as mentioned earlier, using the publicly available SteamSpy API. The data is initially downloaded as individual files in JSON format and later combined to a single data frame containing all 42 635 games available on Steam. This dataframe is saved as one .csv file and all data cleaning and data mining will be done on the data in this file.

Our collection of data is handled in 'CollectData.ipynb'. We use the available python library 'Steamspypi' to directly integrate our API calls with python code. The first step is to collect data on what games are available, namely the attribute 'appid'. Without this attribute we are not able to collect details about each game, since our first available collection(without this attribute) only gathers general data. This is done via the 'steamspypi.download_all_pages' function, which takes about an hour(the API only allows one call to download a page a minute) to download all 44 pages of available data into 44 different json-files. The next step is to collate all appid's into a list to begin fetching details about each game via API call 'appdetails' with the 'appid' attribute. The call to get app details is allowed once a second, and for all available games at the time(42635 answers from the API), which means it approximately takes 12 hours to fetch all details.

5.2 Data dictionary

The data dictionary below intends to explain what the different attributes in the data contain. Field name represents the name of the attribute. Data type is the data format the value is stored in. Description is a short summary of the attribute and its association to our project.

Field Name	**Data Type**	**Description**
appid	Integer	An integer assigned to the app/game
name	Text	The name of the app/game
developer	Text	The name of the developer
publisher	Text	The name of the publisher
userscore	Integer	Score rank of the game based on user reviews(always starts empty, have to generate ourselves)
owners	Text	Categorical interval as string, e.g. “50000 … 100000”
average_forever	Integer	Average playtime since release based on players. In minutes
average_2weeks	Integer	Average playtime in the last two weeks. Only counts players who played. In minutes
median_forever	Integer	Median playtime since release. Only counts players who played at least once. In minutes
median_2weeks	Integer	Median playtime in the last two weeks, amongst players who played in the previous 2 weeks. In minutes

ccu	Integer	Peak concurrent users yesterday (20/11)
price	Integer	Current US$ price in cents
initialprice	Integer	Original US$ price in cents
discount	Integer	Current discount in percents
tags	List of tuple Text and Integer	Game's tags with votes
languages	Text	List of supported languages
genre	Text	List of genres
positive	Integer	Number of positive reviews
negative	Integer	Number of negative reviews
price_range	Text	Price classification ranging from free, cheap, normal, expensive

Figure 1: Table with available attributes from our data set.

6. Data mining methods

6.1 Frequent-terms analysis

Frequent-terms analysis is a method used for analysing what words appear most frequently in a data set. The motivation for choosing frequent-terms analysis in this project was to investigate which tags were most common among popular games. The results from the analysis were presented in a bar graph as well as in a word cloud. These can be seen in figure 12.

6.2 Association rule mining

In data mining, one of the most common methods utilized is association rule mining. The method intends to investigate interesting patterns in data sets and extract valuable correlations between items. The method uses two essential measures for association rules, namely support and confidence. The former describes the percentage of records that contain X and Y to the total amount of records in a data set. The latter measure is defined as the percentage of instances that if a record contains X, that

record will also contain Y. (Zhao, Q., & Bhowmick, S. S. 2003) Association rule mining was used in this project with the aim of investigating the correlation between certain tags of popular games.

6.1.1 Apriori algorithm

The Apriori algorithm enables us to check the frequency of tags and their association rules to see if the tags have any connection and their association. The algorithm is a good way to handle asymmetric binary data with a big dataset (Witten et al., 2011).

The Apriori algorithm checks the number of times the tag and the combination of tags is used. The algorithm then discards some tagsets depending on the support value and the count, this procedure is repeated for each combination until no value reaches the support value. The algorithm then checks rules, their confidence value and prune values with low confidence.

To see the strength of the association rules we look at different measurements to come to a conclusion. Support weight is the number of instances that the set of tags occur together, the greater the support weight the higher support we have from the material. The rules confidence is the chance that the association rule is followed, as an example: FPS, First-person, Action → Shooter, with the confidence 88,6 %, so if the game has the tags FPS, first-person and Action, there is a 88,6% probability that the game also has the tag Shooter. Lift is the last metric we look at, lift determines if it is a good rule or just coincidence.

7. Evaluation of results

7.1 Analysis of results

At the date of collection there are 42631 games available for analysis. For these games there are 33 attributes that have been chosen for possible analysis.

The popularity attribute is the most important for the analysis, and since this has to be defined by a chosen algorithm, it has a big impact on the resulting data: what makes a game successful? The chosen one is to simply take the number of average owners and multiply that with the percentage of positive reviews, and finally square root on the product for normalization. This approach has the disadvantage that the number of owners has a huge impact on the popularity of a game. With domain knowledge and exploring what games have high popularity according to this, it is a defendable metric to use since the most actual popular games are the ones that also have a high popularity with this metric.

The other way to look at what makes a game successful/popular is to look at all quantitative attributes and fetch the 2000 highest rated amongst them, to then compare how many times a game is amongst the top of an attribute. The 13 quantitative attributes are positive, negative, userscore, owners, average_forever, average_2weeks, median_forever, median_2weeks, price, initialprice, ccu and popularity. This produces similar results compared to the popularity attribute, meaning that the 2000 games in the data frame are still there, but the ordering has changed quite a bit. Amongst the top 15 of the popular games, according to popular attribute, there is only one game that has the maximum number of “showings” in the attribute top 2000, which is 10. There is also one game with a very low

count, 3. This shows the shortcomings with the defined popularity attribute, if what you are after is the most current popular/successful games. See *figure 5* under appendix.

There are good indications showing that free games are the most popular ones, which implies that the freemium model is a good choice (Zendle. D et al. 2020). This can be seen in *figure 7*, which presents boxplots of popularity vs price_range category.

Given that the third most common tag of the popular games that were analysed was multiplayer, there seems to be a correlation with multiplayer functionality and popularity. This is in accordance with the conclusion of the study made by Bailey, E. N., & Miyata, K. (2018). Although that study was conducted nearly four years ago, the results seem to align with the findings of this report.

The Apriori table as a whole is one of the results of our project, the table can be used as a guideline for tags that go together in successful games.

The results are very actionable purely statistically, meaning that if a game developer has no idea what game to develop next then the presented models can be used as guidelines to make a decision amongst the varying choices.

7.2 Critique of results

Given the simplicity of the popularity metric used in this report, there are significant opportunities for improvement in future research. Considering that only two attributes were used to determine the popularity of a game, there are arguments to be made that the measurement could be inaccurate. One drawback of the current metric is the fact that game popularity is greatly dependent on the number of owners on Steam. There is a possibility that older games were heavily purchased in the past and therefore have a high number of owners, but have since become unappealing and are no longer popular. This is not taken into account in the current popularity metric, as the number of recently active players is not considered. A better measurement would incorporate more factors to determine whether or not a game is popular.

To make models that are usable in practice there would be certain more metrics that would be needed. Examples of these missing metrics are payment model, publishing date, SteamSpy's exact statistical models and how/what users they gather data from. Generally more data surrounding the owners/players of games would be needed, such as country of origin, language used in games(if available), average hours of gaming per week/month, etc.

Our available models do not incorporate any analysis of the language attribute, which could provide further analysis of how important the specific language(s) are to a game's success.

8. Ethical aspects

It is important to consider the ethical implications when conducting any kind of data mining project. Fortunately, in regards to the EU:s regulation of data protection with GDPR, our project and findings do not rely on any personal data and none of the data presented in our report could be used to identify specific individuals. However, there are still some ethical aspects to be considered. Although the data collected with SteamSpy relies on Steam users setting their profile to public instead of private, some users might not consent to their data contributing to Steam Spy or projects such as this one. But

considering the fact that a Steam user can at any time decide to set their profile to private, this is not a major issue.

A more important ethical aspect to consider is not so much the findings of our project, but rather what these findings could potentially lead to in the future if more data is analyzed. If in continuation of this project there is more research done on whether video games containing micro-transactions and loot boxes are a factor for success, game developers might be more inclined to utilize these features in future games. There is empirical evidence that video game loot boxes are clearly connected to problems with gambling. There is reason to suggest that authorities and governments classify loot boxes as a form of gambling and as such regulate it accordingly. The video game industry is still growing and the revenue generated by microtransactions and consumers purchasing loot boxes is increasing year by year. (Zendle, Cairns, 2018) If data findings conclude that integrating loot boxes in video games truly is a success factor, it will become even more prevalent than it is today. As such, a rise in gambling addiction among gamers should not come as a surprise.

9. References

Bailey, E. N., & Miyata, K. (2018). Estimating the Value of Multiplayer Modes in Video Games: An Analysis of Sales, Ratings, and Utilization Rates. In *KDIR* (pp. 153-160).

Huang, Y., Jasin, S. & Manchanda, P. 2019, "“Level Up”: Leveraging Skill and Engagement to Maximize Player Game-Play in Online Video Games", *Information systems research,* vol. 30, no. 3, pp. 927-947.

Martinez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernandez-Orallo, J., Kull, M., Lachiche, N., Ramirez-Quintana, M.J. & Flach, P. 2021, "CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories", *IEEE transactions on knowledge and data engineering,* vol. 33, no. 8, pp. 3048-3061.

Steamspy. (2021). *Steam summary from yesterday.* https://steamspy.com [Retrieved: 2021-11-05]

Witten, I.H., Frank, E. & Hall, M.A. 2011, *Data mining: practical machine learning tools and techniques*, 3rd edn, Morgan Kaufmann, Burlington, MA.

Zendle, D., Meyer, R. & Ballou, N. 2020, "The changing face of desktop video game monetisation: An exploration of exposure to loot boxes, pay to win, and cosmetic microtransactions in the most-played Steam games of 2010-2019", PloS one, vol. 15, no. 5, pp. e0232780-e0232780.

Zendle, D., & Cairns, P. (2018). Video game loot boxes are linked to problem gambling: Results of a large-scale survey. *PloS one*, *13*(11), e0206767.

Zhao, Q., & Bhowmick, S. S. (2003). Association rule mining: A survey. *Nanyang Technological University, Singapore*, *135*.

10. Appendix

```
In [84]: clean_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42631 entries, 0 to 42630
Data columns (total 33 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   appid            42631 non-null  int64
 1   name             42623 non-null  object
 2   developer        42484 non-null  object
 3   publisher        42561 non-null  object
 4   positive         42631 non-null  int64
 5   negative         42631 non-null  int64
 6   userscore        42575 non-null  float64
 7   owners           42631 non-null  float64
 8   average_forever  42631 non-null  int64
 9   average_2weeks   42631 non-null  int64
 10  median_forever   42631 non-null  int64
 11  median_2weeks    42631 non-null  int64
 12  price            42631 non-null  int32
 13  initialprice     42631 non-null  float64
 14  ccu              42631 non-null  int64
 15  languages        42609 non-null  object
 16  tags             42631 non-null  object
 17  popularity       42575 non-null  float64
 18  is_popular       42631 non-null  bool
 19  action           42631 non-null  bool
 20  singleplayer     42631 non-null  bool
 21  multiplayer      42631 non-null  bool
 22  adventure        42631 non-null  bool
 23  co-op            42631 non-null  bool
 24  rpg              42631 non-null  bool
 25  strategy         42631 non-null  bool
 26  open_world       42631 non-null  bool
 27  atmospheric      42631 non-null  bool
 28  shooter          42631 non-null  bool
 29  horror           42631 non-null  bool
 30  survival         42631 non-null  bool
 31  simulation       42631 non-null  bool
 32  price_range      42631 non-null  category
dtypes: bool(14), category(1), float64(4), int32(1), int64(8), object(5)
memory usage: 6.3+ MB
```

Figure 3: The DataFrame after cleaning and preprocessing the data set.

```
In [85]: #Exploration of our WHOLE data set
         whole_data = clean_data
         shape = whole_data.shape
         shape_string = "We have {nr_attributes} attributes and {nr_games} games in our data set after cleaning the data."
         print(shape_string.format(nr_attributes=shape[1], nr_games=shape[0]))
```

```
We have 33 attributes and 42631 games in our data set after cleaning the data.
```

Figure 4: Continued from *figure 3,* summation of the DataFrame.

appid	count	name	developer	publisher	positive	negative	userscore	owners	average_forever	average_2weeks	...	co-op	rpg	strategy
570	8	Dota 2	Valve	Valve	1344991	260528	0.837730	150000000.0	36023	1820	...	True	True	True
440	7	Team Fortress 2	Valve	Valve	771699	51686	0.937227	75000000.0	8377	2522	...	True	False	False
730	8	Counter-Strike: Global Offensive	Valve, Hidden Path Entertainment	Valve	5301391	713807	0.881333	75000000.0	29195	1001	...	True	False	True
105600	8	Terraria	Re-Logic	Re-Logic	817927	17427	0.979138	35000000.0	7389	497	...	True	True	False
550	7	Left 4 Dead 2	Valve	Valve	541773	14739	0.973515	35000000.0	2154	160	...	True	False	False
4000	8	Garry's Mod	Facepunch Studios	Valve	732915	25880	0.965893	35000000.0	8556	493	...	True	False	False
945360	8	Among Us	Innersloth	Innersloth	544118	44163	0.924929	35000000.0	1245	147	...	True	False	False
304930	7	Unturned	Smartly Dressed Games	Smartly Dressed Games	416537	39101	0.914184	35000000.0	5869	3195	...	True	False	False
238960	7	Path of Exile	Grinding Gear Games	Grinding Gear Games	157499	15997	0.907796	35000000.0	7199	2545	...	True	True	False
230410	7	Warframe	Digital Extremes	Digital Extremes	414315	43745	0.904499	35000000.0	8147	1088	...	True	True	False
218620	8	PAYDAY 2	OVERKILL - a Starbreeze Studio.	Starbreeze Publishing AB	484650	61410	0.887540	35000000.0	4220	875	...	True	False	True
340	3	Half-Life 2: Lost Coast	Valve	Valve	8513	1218	0.874833	35000000.0	258	0	...	False	False	False
359550	8	Tom Clancy's Rainbow Six Siege	Ubisoft Montreal	Ubisoft	868249	125095	0.874067	35000000.0	13888	445	...	True	False	True
252490	10	Rust	Facepunch Studios	Facepunch Studios	606104	92561	0.867517	35000000.0	16736	1050	...	True	False	False
1172470	8	Apex Legends	Respawn Entertainment	Electronic Arts	295739	46399	0.864385	35000000.0	4729	740	...	False	False	False

15 rows × 33 columns

Figure 5: Top 15 games sorted on popularity with count included(count is how many times the game appear in top 2000 sorted on each quantitative attribute)

```
In [87]: #Tag analysis of all games
         result_tags = clean_data[['action', 'singleplayer', 'multiplayer', 'adventure', 'co-op', 'rpg', 'strategy', 'open_world', 'atr
         tag_frequency(result_tags).sort_values()
```

```
Out[87]: open_world       7.238864
         co-op            8.193568
         survival         8.442213
         horror          10.928667
         shooter         11.665220
         atmospheric     15.390209
         multiplayer     16.068119
         rpg             20.332622
         simulation      20.714973
         strategy        22.415613
         adventure       43.421454
         singleplayer    43.890596
         action          46.930637
         dtype: float64
```

Figure 6: The 13 most popular tags, with 'indie' removed, and their frequency in percentage within the whole data set.

```
In [88]: #Create boxplot of popularity vs price_range
         plot_data = clean_data.sort_values('popularity', ascending=False)
         plot_data = plot_data.head(250)
         plot_data.boxplot(by = 'price_range', column = ['popularity'], grid = False, figsize = (16,10))
```

```
Out[88]: <AxesSubplot:title={'center':'popularity'}, xlabel='price_range'>
```

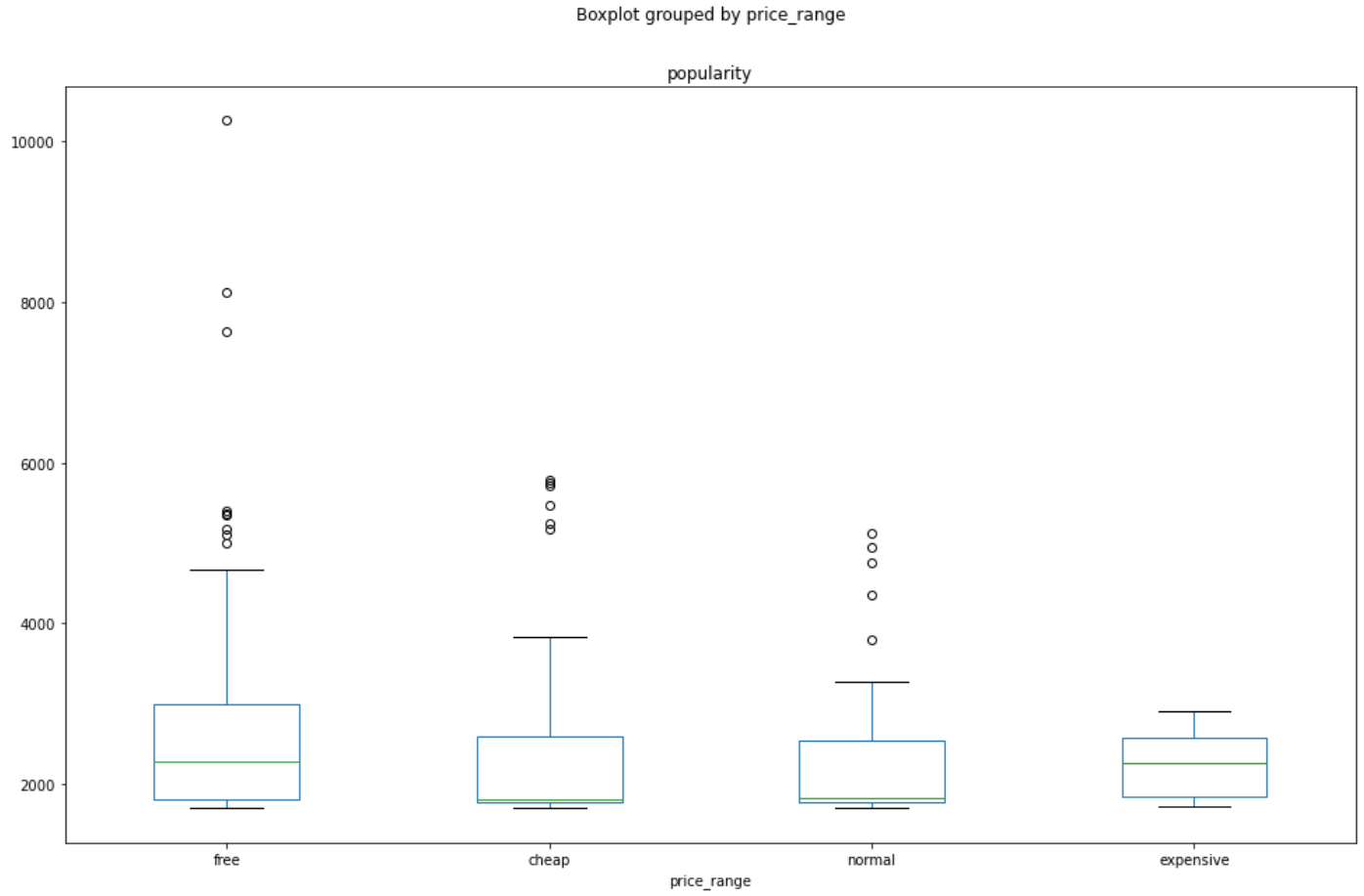


Figure 7: Top 250 games according to the popularity attribute sorted by their category within price_range. Y axis is popularity and X axis is price_range.

```
In [89]: #Present tag % on whole data set
         plot_data_tags = plot_data[['action', 'singleplayer', 'multiplayer', 'adventure', 'co-op', 'rpg', 'strategy', 'open_world', 'a
         tag_frequency(plot_data_tags)

Out[89]: action          83.6
         singleplayer    78.0
         multiplayer     68.8
         adventure       66.4
         co-op           54.0
         rpg             30.0
         strategy        36.0
         open_world      38.0
         atmospheric     48.0
         shooter         42.4
         horror          20.0
         survival        29.6
         simulation      28.4
         dtype: float64
```

Figure 8: Continued with data from *figure 7.* Frequency in percentage of the 13 chosen most popular tags of the top 250 games from *figure 7*.

In [114]:
```
#Top 40 tags
tags.sum().sort_values(ascending=False)[:40]
```

Out[114]:
```
Indie                 28306
Action                19258
Singleplayer          18711
Casual                17935
Adventure             17642
2D                     9748
Strategy               9287
Simulation             8831
RPG                    8059
Puzzle                 6884
Atmospheric            6561
Early Access           5653
Multiplayer            5608
Story Rich             5354
Pixel Graphics         5029
Arcade                 4427
First-Person           4385
Fantasy                4274
Colorful               4226
3D                     4126
Cute                   4097
Shooter                4037
Funny                  4029
Anime                  3999
Great Soundtrack       3951
Free to Play           3904
Retro                  3882
Exploration            3848
Platformer             3829
VR                     3827
Horror                 3821
Sci-fi                 3754
Difficult              3631
Family Friendly        3509
Female Protagonist     3204
Open World             3031
Survival               2940
Relaxing               2913
Violent                2803
Co-op                  2766
dtype: int64
```

Figure 9: Top 40 counted tags in the whole data set, none excluded.

```
In [95]: #Frequencies of the tags
         tag_frequency(tags).sort_values(ascending=False)
```

```
Out[95]: Indie           66.397692
         Action          45.173700
         Singleplayer    43.890596
         Casual          42.070324
         Adventure       41.383031
                            ...
         Hardware         0.021111
         Boss Rush        0.016420
         8-bit Music      0.014074
         Masterpiece      0.009383
         Batman           0.002346
         Length: 429, dtype: float64
```

```
In [96]: #Plot of tags
         _ = item_frequency_plot(tags, 0.125).plot.bar()
```

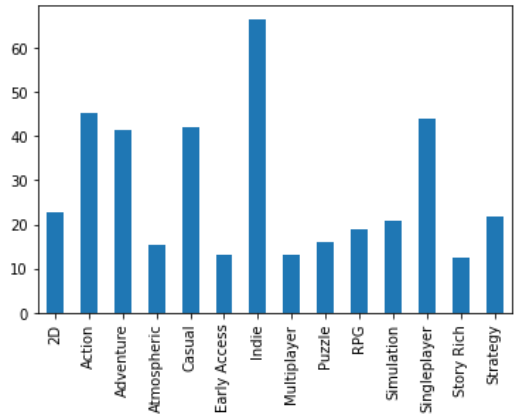


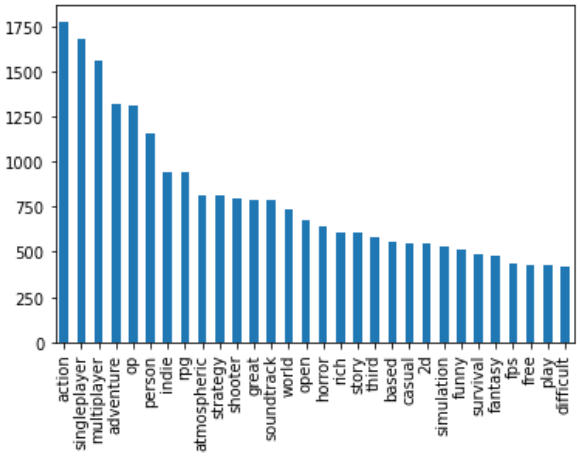
Figure 10: Continued with data from *figure 9*, presented in frequency in percentage and those frequencies in a bar chart.

In [117]:
```
#Top 15 best rules
tags_rules[(tags_rules.num_antecedents >= 3)
               & (tags_rules.confidence > 0.6)
               & (tags_rules.support > 0.05)
         & (tags_rules.lift > 2)].sort_values('support', ascending=False).head(15)
```

Out[117]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	num_antecedents
15368	(Adventure, Atmospheric, Singleplayer)	(Story Rich)	0.2890	0.3040	0.1850	0.640138	2.105718	0.097144	1.934077	3
12050	(FPS, First-Person, Action)	(Shooter)	0.1840	0.2735	0.1630	0.885870	3.239011	0.112676	6.365524	3
12052	(FPS, Action, Shooter)	(First-Person)	0.1780	0.2860	0.1630	0.915730	3.201854	0.112092	8.472800	3
12053	(First-Person, Action, Shooter)	(FPS)	0.1700	0.2160	0.1630	0.958824	4.438998	0.126280	19.040000	3
11398	(Multiplayer, Online Co-Op, Action)	(Co-op)	0.1710	0.3700	0.1570	0.918129	2.481429	0.093730	7.695000	3
16926	(Adventure, Great Soundtrack, Singleplayer)	(Story Rich)	0.2505	0.3040	0.1530	0.610778	2.009140	0.076848	1.788185	3
12067	(First-Person, Action, Singleplayer)	(FPS)	0.2135	0.2160	0.1525	0.714286	3.306878	0.106384	2.744000	3
12066	(FPS, Action, Singleplayer)	(First-Person)	0.1690	0.2860	0.1525	0.902367	3.155129	0.104166	7.313091	3
12374	(Singleplayer, Action, Shooter)	(FPS)	0.2245	0.2160	0.1455	0.648107	3.000495	0.097008	2.227949	3
12372	(FPS, Action, Singleplayer)	(Shooter)	0.1690	0.2735	0.1455	0.860947	3.147886	0.099278	5.224617	3
11998	(FPS, Action, Multiplayer)	(First-Person)	0.1665	0.2860	0.1445	0.867868	3.034503	0.096881	5.403682	3
11999	(Multiplayer, First-Person, Action)	(FPS)	0.1855	0.2160	0.1445	0.778976	3.606369	0.104432	3.547122	3
12243	(FPS, Action, Multiplayer)	(Shooter)	0.1665	0.2735	0.1430	0.858859	3.140252	0.097462	5.147330	3
12244	(Multiplayer, Action, Shooter)	(FPS)	0.2035	0.2160	0.1430	0.702703	3.253253	0.099044	2.637091	3
12843	(Singleplayer, Action, Shooter)	(First-Person)	0.2245	0.2860	0.1415	0.630290	2.203810	0.077293	1.931241	3

Figure 11: Top 15 best rules by support attribute after apriori algorithm.




```
wordcloud.fit_words(tag_list_popular_counted_tags.to_dict())
plt.figure()
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

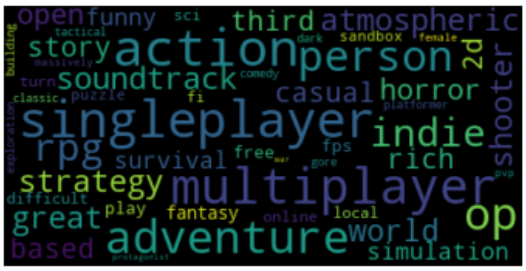


Figure 12: Bar chart and word cloud image on tags on the 2000 games where attribute is_popular is true.