**Basic Information**

**Title:** A Cost Analysis of Video Games on Steam

**Names:** Jake Badger, Ryan Golding, William Tann

**Emails:** jtbadge@g.clemson.edu, rgoldin@g.clemson.edu, wtann@g.clemson.edu

**CUID’s:** C28371039, C18831978, C20981260

**Project Repository:**<https://github.com/whtann/cpsc4030-Project>

**Overview and Motivation**

Each member of the group has played video games at some point in their lives, and most continue to play them today. Video games, however, can be a large time commitment and, more importantly, an expensive hobby. Therefore, the group wants to understand and stress the importance of determining which games are worth their cost. Each game is different depending on its size, genre, and reviews, and we want to understand the correlation between each component to figure out which games are worth an investment according to their pricing. We want to do this to enhance the decision making of buying certain games over others. In the future, there may be a group of games available for purchase, but we cannot afford buying all of them at once. Some games may be worth their price as is, some may be good games but would be better purchases during sales, and some may be outright bad games. We want to use visualizations to compare this group of games and determine which game would be in our best interest to purchase. Each game could be compared through reviews, memory, and other aspects to then decide whether a game would be worth an investment and potentially purchased in the future.

To accomplish the tasks above with our dataset, we felt the need to implement a drill-down style for our website. We figured that people who would look at this dataset would want to compare different video games based on entire genres or the entire dataset. By using the drill-down method, the user can search through the visualizations on our website and get an idea of the distributions of games based on pricing, genre, and other factors that are compared.

**Inspiration**

Aside from the fact that each member of the group has played video games at some point and wants to know which games on steam would be best for purchasing if there wasn’t a game already in mind, inspiration for this project comes from a couple of sources. First, a visualization can be found about games based on the number of global players at: <https://www.reddit.com/r/gaming/comments/awpmj0/top_15_steam_games_by_player_count_jan_2015_dec/>.

This dataset, while ranking the top games on steam by player count, does not give the user information based on price or enjoyment. As we began looking for a dataset, we wanted to see the correlation between price and enjoyment for video games, rather than just having several players. We then figured we could expand even further and determine the relationships between developer, release date, storage, and platform. Because of the limitations of the webpage layout and the differences in release date/developer, we did not incorporate these into our visualizations, but further inspiration could produce a visualization based on these elements.

**Questions**

From our website, we wanted to answer the following questions:

* What video games on steam are worth their price more than others?
* What is the correlation between reviews and prices?
* Has there been a surge in popularity of the game over the past 30 days?
* Does a game’s price fluctuate with its storage requirement?
* Do certain genres have higher costing games than others?
* What’s the correlation between memory and pricing?
* What do games’ positivity ratios look like comparatively?

We are addressing the overall question “What video games on steam are worth their price compared to others?”. We want to understand the correlation between video game prices, total number of reviews, positivity score (the amount of positive to negative reviews), storage requirement, and genre. The reviews will be analyzed both over the last 30 days (to determine whether a game is surging in popularity) and all-time. Using our website, we want to understand the relationship between a games’ positivity score and its price. We will analyze the positivity ratio of the game, determine the impact of the number of players on the score, and compare it to the price of the game. Another goal is to compare the pricing based on the genre of the game. We want to see if there is a certain genre with higher costing games than another genre, or if the overall dataset has a higher number of games listed in a certain price range. Finally, we want to give the user the option to compare two different games, so that they can better understand the relationship between the two games. For example, if one game is very close to another on the visualization, the user will be able to compare both games and see whether one game has higher reviews or positive ratio than the other.

At the beginning of the project, we had some other questions we wanted to answer concerning the impact of developer/publisher, release date, and operating system on the game. Over the course of the project, however, we found that these questions were going to be more difficult to answer than could reasonably be put on our visualizations page. The developer/publisher and release date were going to be different for every game and visualizing 6,000 elements and comparing these values with number of reviews would have been too big for the visualization to be understood. As a result, we had to change the questions we wanted to answer for the website. Originally, we only wanted to look at each question and answer it with their own graph. For example, the question “what is the correlation between developer and reviews?” would have been answered with a simple scatterplot without interaction between the graph and other visualizations. We realized this and evolved the questions we were trying to answer to have an available correlation between each one. For example, when we compare the price of the game with other elements, we can use the genre to show another visualization comparing it and the price. This way, the graphs can interact with one another, and we can answer multiple questions with both visualizations, which is what we tried to implement using our final website design.

Other questions that arose over the implementation process were along the lines of how to incorporate averages and get an effective visualization, and how to incorporate the operating system into the price/genre relationship mentioned previously. We also added the questions regarding genre and price to the visualizations after receiving feedback that it would be interesting to show the correlations. For the operating system, we visualized the relationship between the genre, price range, and operating system using a bar chart, but found that the bar chart would be too large with increasing games in certain price ranges, and that the only operating system that could stand alone with a game was Windows. As a result, we could not see any relationship between the operating system and the reviews/genre/price, so we deemed it unnecessary in our final visualization. The correlating bar chart is shown below for games with the genre “action” and the price range “$<1”. The operating system is colored where blue is windows and orange is windows + mac.

Chart, bar chart

Description automatically generated

We are expecting that our dataset will help us learn which games are worth their price, and what to look for in determining future purchases. We want to learn the pros and cons of certain games and learn how to spend money wisely on a hobby like playing video games. We are hoping to accomplish a visualization that allows us to understand the relationship between each aspect and determine the best use of money and time when purchasing a game.

**Data**

Our dataset is based on a custom built script that takes a list of games on the Steam Marketplace.  The script essentially scrapes web data provided from the top selling Steam storefront webpage:

<https://store.steampowered.com/search/?sort_by=_ASC&ignore_preferences=1&filter=topsellers>

What the script is looking for is any entry that has reviews on its page.  This is done to mitigate the problem of running into pre-ordered games that haven’t been played or products that steam sells separately.  If the script finds a page that has reviews, it will visit the page for the game and pull all information that we have deemed relevant to our problem.  It then formats this information into an excel spreadsheet, where we believe we will have a much easier time being able to parse this data.

The main issue with using the script was that the data pulled from steam required a large amount of necessary cleanup and filtering. The script automatically fills out the table in the format we are looking for, however we may need to look through to make sure everything is formatted correctly and in the right place, since this is a custom script. The fact that we are pulling this data from Steam presents a few other challenges.  For example, Steam doesn’t account for other platforms of gaming (like consoles) because it is a PC games marketplace.  Also, Steam’s algorithms for displaying top selling games might have some unseen bias towards certain games rather than others.

As we began to use the dataset, a large amount of cleanup was needed for a couple of reasons. First, some games were pulled from the website with names as the only information given, where every other value in the dataset returned “N/A”. To combat this, we used to excel to filter out games that did not have a release date listed, then filled in numerical values, such as storage capacity and number of reviews, to 0. This can skew our dataset a bit, as games with either no price or no reviews are listed at the bottom of our scatterplot, leading people to believe that the game is a bad game to purchase. Second, some small cleanup was needed to rename characters and values in the names of the games, so that the data was able to be visualized and the game itself could be read using the tooltip. Some of these games still have character issues, as saving to a csv file causes some error with items such as the trademark logo. We felt that this was only a small issue, however, and moved on. Finally, a large amount of cleanup was needed for the genre of the game. The script returned a new genre value for each game, meaning we had a similar issue that we had with the developer and release date. Luckily, the script returned multiple values for the genre. We had to manually go through all 6,000 games and insert them into a list of genres so that we could shrink the value down to 30. Some games may therefore be linked with a genre that isn’t the first listed in its store value, but we wanted to condense them into game genres and see a correlation with price and a heat map. We therefore chose a genre within the list of games that was like others and shrunk the genres down to compare the dataset more easily.

**Exploratory Data Analysis**

To look at the dataset initially, we used mostly scatterplots and bar charts since we felt that these would be the best visualizations to compare our large distribution of prices, games, and other aspects. Some examples of our original designs are below:

Graphical user interface, application, table

Description automatically generated

From this scatterplot, we were able to see groupings between the price of the game (x-axis) and its positivity ratio, or the number of reviews that were positive toward the game (y-axis). From this result, we still felt that this was a good option for our dataset, as we could easily see the relationships between the price and the ratio. The small problem that we noticed with this visualization was that there is a lot of clumping of points at certain prices, but this is expected and not something we can really change with the dataset. The price of a game is normally fixed near a value of 10 but costs 99 cents (i.e, 59.99, 49.99). Since we can’t change the pricing of the data, we felt that a new visualization would be better, and we implemented our heat map to compare the price and genre in our final visualization. With this result though, we do see a trend like what we were expecting, although with a slight twist. We expected to see a large clumping of datapoints in the corners of the visualization and a linear trend between the cost of the game and its positivity ratio. While we don’t see much of a linear trend, we do see a lot of clumping near the top left corner, indicating that there are a lot of games with high positivity and low pricing, so there are a lot of games we would consider “good” from a purchasing perspective. In the graph, we also considered how to better implement color. For a while, we had many colors due to the thousands of genre identities and decided to remove them and add a single color. Once we realized we could compare the genres with more cleaning, however, we implemented the color on the scatterplot again and have a legend indicating the colors and their respective genres.

Timeline

Description automatically generated

From this bar graph, we were able to visualize the differences between each game and its total number of reviews. We decided to remove this graph and all other implementations regarding the game by itself, the release date, and the developer/publisher. This graph is completely unreadable due to the large number of objects being graphed on the page, and since we wanted the designs to fit into one page, we felt that this would be impossible without extensive formatting within the data. We also realized from feedback that we would need to implement the graphs with interaction and show the differences between each graph when an interaction occurs. Again, regarding the large number of components being graphed on the page, and the fact that we would have to shrink it to show interaction, we decided to scratch the bar chart idea for anything more than a couple of data points. This way if we wanted to implement a bar chart we could, however we would not implement one with an x-axis as large as this (and much bigger than a couple of points).

Chart

Description automatically generated with low confidence

The last graph we used to initially look at the dataset was a forces graph. From this graph, we wanted to compare the number of games on a given operating system, and later increase the sizes of the bubbles to show the number of reviews given the operating system. Once we saw this graph and received feedback on the slowness of our website due to forcing the points together, we decided to scrap the idea. Firstly, the graph doesn’t tell us anything other than the operating system, and still looks a bit confusing to understand or see any correlation between the game and something else. Secondly, the colors graphed here are for windows, mac/windows, linux/windows, and mac/windows/linux games. We realized that there aren’t any games solely listed as mac or linux games, and thus we could not compare a platform to its number of reviews as every game had windows listed as a playable platform. We originally decided to go a different direction and then ended up scrapping the operating system from the visualizations altogether, as it was not showing anything useful for comparison.

**Design Evolution**

Over the entire course of the project, there were 5 main visualizations we considered for the project: scatterplots, bar charts, forces/area graphs, bubble graphs, and heat maps. Originally, we felt that the scatterplots, bar charts, and area graphs would be sufficient since we wanted to compare every game with each value (price, reviews, etc.). Once we implemented the bar charts and area plots shown above, as well as some other plots shown below, we decided to scratch the area charts in favor of the heat map design. Shown below are some original design ideas and excess charts:

Chart

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Shown above is two size charts, one showing the developer and the number of reviews a game has, and the other showing the game vs. platform. For the first graph, the radius of the circle corresponds to the number of reviews and the color corresponds to the developer. We decided to remove this graph for a few reasons. Firstly, the size of the circles needed to be scaled down so games with thousands of reviews did not fly off the page, but this caused other circles to become impossible to read. Secondly, there are too many colors on the page to give the reader an idea of what’s happening with the dataset. Very few circles have the same color, but having the color doesn’t show a comparison between the developers, rather it shows a comparison between games and reviews. Using the forces to bring the graph together was also causing an issue, so we cut this graph. For the second graph, the circles show the platform comparing games again, without the color implementation. We see the same problem comparing with the first graph, and the opacity of the circles also gives more issue here since the values are all one color.

Text, timeline

Description automatically generated with medium confidence

The other bar cart we considered was that of the genre comparing the total number of reviews. We scratched this graph for similar reason to that of the number of games, as there are just too many elements to graph and fit into the width of the page while effectively showing the user what is being visualized. The other issue going on here is that the genres could overlap based on the dataset, and the bars don’t do much else other than report a value. No real comparisons between the genre and number of reviews can be made and correlated with the price of the game. While reporting the numerical value is good information for the user, it would be hard for them to understand the correlations between the bars as many of the genres overlap and some games had very little if any reviews at all, thus causing some tiny visualization reports.

Chart, bubble chart

Description automatically generated

Lastly, to compare the memory, positivity ratio, and price, we attempted a bubble plot so that we could see the distribution across the games, as well as compare them to genre via color. While this would be a good plan with a smaller dataset with more distribution across the x and y axes, here it causes a bit of an issue for the reader to understand. Firstly, the prices are all only on one line, causing the bubbles to greatly overlap. Secondly, the size of the bubbles is going off of the screen, and the bubble size would become too large for the reader to see any trends going on in the dataset. While the memory size would be nice to compare in our current scatterplot via dot radius, would cause the already clustered dataset to appear even more clustered, and small elements would get completely lost in the visualization.

A picture containing timeline

Description automatically generated

The scatterplots tended to be effective in showing different correlations based on price, as well as presenting expected trends based on positivity ratio and storage capacity. The main difference between the first and final scatterplots was implementing a color code based on genre as opposed to the name of the game, as well as including a scale for the user to change and more easily see the prices of games within each group. We also see some clumping as with the positivity ratio, but only based on price, likely due to the wider scaling comparative to price (prices are fixed near $0.99 where storage could be any value). Although this graph showed some clumping we did like that we could see a lot of the games spread out and could compare easily, so we decided to keep the scatterplots.

Once our initial graphs were complete, we decided to look at the layout of the website. Originally, we planned to have buttons corresponding to each graph as to have the maximum amount of space possible. Once we received feedback and knew we had to implement interactions, we changed our design to the one shown below:

Chart

Description automatically generated

Rather than implementing each graph, there are buttons above the scatterplot corresponding to changing the parameters and what component is being graphed. For example, clicking positivity ratio vs price would change the x and y axes, while clicking the max buttons would change the scale of the axis. We originally liked the layout of the website, however there are a few issues. First, the scatterplot is very hard to read with only one color, and some of the data could be categorized to make it easier to read, Secondly, the bar chart is only graphing one element, and is not really comparing anything other than its total reviews and last 30 days reviews to itself. Finally, the forces chart is taking up a large amount of space. We wanted to graph the operating system here, but did not know a proper graph for displaying the appearance of a value. Once we edited the charts on the website, we decided to go a different direction on the layout, and have our final presentation as a result.

**Implementation**

Chart

Description automatically generated

Graphical user interface, chart

Description automatically generated

Shown above is the final implementation of the website, complete with a scatterplot displaying the positivity ratio vs. price, a heatmap showing the specific number of games per genre in a price range, and a bar chart to compare two games at any time. The idea behind this layout was to allow the user to see as large of a selection of games as they wanted, while still being able to interact with the different graphs and buttons to compare values.

Scatterplot

The goal of the scatterplot location was to graph the positivity ratios and memory sizes versus price. This way, we could see any correlations between the points and genre, since that is our color of the dots, and decide which games would be good to buy based on their position. Some functionality is shown below:

Chart, scatter chart

Description automatically generated

Hovering over dots on the heatmap gives a tooltip that tells the user the name, price, rating, and size of the game. This way, the user can read the data and know which game is positioned where.

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

There are buttons located above the graph. The first 3 buttons change the y axis, such that the size of the game, positivity ratio over the last 30 days, and positivity ratio all time can be graphed on one scatterplot. This allows the user to see differences between the games and their respective values, while showing the name of the game on the chart using the tooltip.

Chart, scatter chart

Description automatically generatedChart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

The third implementation of the scatterplot is the other 3 buttons located above the graph and to the right. These buttons change the scale of the x axis. The goal of this implementation was to break apart the clumping located at the bottom of the graph when graphing the maximum values. This way, users can more easily see data that has been hidden by lots of points at one given location.

Graphical user interface

Description automatically generated with medium confidence

More buttons were implemented to the right of the scatterplot, allowing the user to toggle the dots based on genre. This way, if a user has a specific genre in mind they wish to compare when graphing the games, they could toggle the dots and more clearly see those selected games. The toggles also work to allow multiple genres, giving the user free range of all the games in the dataset.

Chart

Description automatically generatedChart

Description automatically generated

Lastly, clicking on any of the dots of the scatterplot will change the variable located at the top of the page, Game 1 and Game 2. The click will also change the values located in the bar chart. This was implemented so that users could compare two games at given points and see their number of reviews all time and over the last 30 days. We wanted the user to be able to compare games if, say, there were two games extremely close together on the scatterplot, and someone wanted to make an informed decision about which one would be more worth buying. Clicking on any one point changes one of the games and sets the previous game 1 to game 2, so comparing any two games would need both clicks on the dots.

Chart, bubble chart

Description automatically generatedHeatmap

The heat map has implemented a tooltip that shows the number of games within a square. This way, the user can hover over the squares on the screen and see the number of games within a price range and genre. The user can see the relationship between genre and price this way, and we can answer the questions of “Which genres have more games in them?” and “What price range has the most games?”

We also wanted to implement a function where the heatmap is connected to the scatter plot and displays the genre and range of games. We tried doing this but got errors when trying to call .on(‘click’) for the rectangles and the actual heat map. Doing only one would put the scatter plot on the wrong graph while doing both would not graph anything. We then tried to implement a function to just connect it to the buttons but found error with that as well and weren’t sure how to fix the issue. If we had this implementation, it would connect the heat map to the scatterplot, and we could see the different implementations comparing the genre and price range. At this point, the heat map mostly serves as a way of showing the most reported price of games.

Bar chart

Chart, bar chart

Description automatically generated

Text

Description automatically generated with medium confidence

The bar chart has a tooltip implemented to show the total reviews of games both all time and over the last 30 days. This was done for the comparison of two game elements. If two games are close together on the scatterplot, they can be compared to determine which one has more players/more people saying the game is positive. The chart also has a tooltip that shows which games are being compared at any given time.

**Evaluation**

Using our visualizations, we’ve learned a couple of ideas from the scatterplots. The first is that a large chunk of games would be considered “good” by our standards, i.e. having a low price and a high positivity score. We can see this through the large clustering of the points towards the upper left quadrant of the graph. The second thing from the scatterplot is like the first, in that a lot of games have relatively low storage requirements compared to their prices. A few points straggle towards the top of the graph, but a lot of the points cluster towards the lower left corner of the visualization.

From the heat map, we can also understand an idea about the number of games under a certain genre, as well as the price range for those games. From our visualization, Action, Adventure, Early Access, Indie, RPG, Simulation/VR, and Strategy games are the genres with the highest number of games. Also from this graph, most games lie within the $10-30 range, as each of the previously listed genres have their highest density of games in this price range.

Lastly, from the bar chart, we can understand the relationship between a game’s reviews compared to other games within the dataset. For example, if there are two games with similar pricing and memory size, we could click on both games and compare them to see which has more reviews, and therefore has more people who enjoy the game. Using this, we can develop an understanding of the reviews and pricing/memory size. We can determine that the number of reviews on a game is affected by other factors (potentially developer or release date) and can compare the two games to make a better decision on which game to purchase.

From our questions above, we’ve used each dataset to answer a direct question we had when planning our website design. First, we used the heatmap to answer which genres have higher priced games than others and saw a large concentration of games between $10-30. We also used scatterplots to develop a correlation between a review ratio and price, as well as an idea of the storage requirement and its affect on pricings. Finally, we used the bar chart to determine if there were many reviews for a game over the last month, and the difference in number of reviews between games. Using each of the data visualizations, we can determine the best games within their genre, price range, and positivity for purchasing, as well as compare games for future purchases down the line.

We believe our visualization is effective at answering the questions listed above but believe our visualizations could be improved to give a much larger idea of general trends in the data. First off, we believe our bar chart could be improved to compare multiple games. We could implement this now, but because of having the datasets limited to one page for easy interaction, as well as not wanting to take away from the scatterplots and heat map, we did not implement the bars to compare any more than 1 game. Having a large amount of data would have caused the bar chart to be unreadable. We believe if we could move the bar chart lower and move it off the page, it would allow us to compare all the games in a given genre, which would connect better with the scatterplot and heat map. Another thing we could do to improve our visualizations is incorporate a plot that compares our values with release date and developer. Since we could not shrink the developers like how we shrank the genres, we left that key component out of our visualizations. We think that incorporating either an area chart (bubble chart) or a histogram would allow us to compare developer average values and show the different games/reviews in comparison to those averages. We also could incorporate dates to show how earlier release dates most likely have an affect on the number of reviews for the game. We could also incorporate more functionality with the heat map, allowing it to connect to the buttons and figuring out how to connect it to the scatterplot more smoothly. Finally, we could improve this dataset by doing deeper cleaning of the values to compare on each plot. A lot of games pulled using the steam script caused errors such as “N/A” appearing. For the sake of simplicity, games with an N/A value in certain areas were replaced with a 0, just so the groups of games could be compared more easily since they had a price.