**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>

**Background 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.

**Project Objectives**

Using our given dataset, we plan to answer the main questions:

* What video games on steam are worth their price more than others?
* What is the correlation between reviews and prices?
* Do developers influence the number of reviews a game gets?
* Has there been a surge in popularity of the game over the past 30 days?
* Does platform release influence the number of reviews?
* Does a game’s price fluctuate with its storage requirement?
* Does storage requirement influence the number of reviews?
* Do certain genre’s get more reviews for a game, and increase the price?

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, developers/publishers, total number of reviews, positivity score (the amount of positive to negative reviews), platform requirement, 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. Our process is to determine whether platform release, genre, and developer influences the number of reviews a game receives over a given period, either in the last month or overall. We are looking to see if certain developers attract more reviews and players with their games, as well as whether platform release (Mac, Windows, Linux) causes an increase in players and reviews. After this, we want to understand the relationship between a games’ positivity score and its price. Using the determined relationship between developer, platform, and number of players, 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. Our final goal is to then categorize the data based on its pricing (whether the price is high or low compared to its reviews), it’s storage requirement (to determine if a game requires more storage compared to its price) and its population size (whether the number of reviews and positivity ratio indicates a better game for its price) to determine which games are most worth their price.

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. The benefits are as follows:

* We can determine which developers create better games, thus allowing certain games removed from purchasing due to bad developer track records
* We can determine whether developers influence ratings in good or bad ways, such that certain games may be inflated with reviews and not be worth their price
* We can determine whether the game can be run on more than one system and whether its usage across platforms/system requirements makes it a stronger game
* We can determine whether the pricing of a game is worth it as is and can spend money more wisely. We can also determine whether a game is good, but not worth it’s price, or if a game is just outright bad
* We can determine if a game has a massive population or surge of players by their reviews within the last 30 days and can understand its impact on whether the game is worth its price.

**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.

**Data Processing**

We should not have to do much data cleanup as 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.  These problems, however, don’t conflict much with our problem as the data source just serves as a large pool of information about top selling games. We plan to use our script to pull the game name, release date, number of reviews in the last 30 days, review positivity ratio in the past 30 days, total number of reviews, total positivity ratio, developer, publisher, memory requirement, and platform release. We plan on using the script to pull information from Steam’s website and update the data for new information as more games get released.

**Visualization Design**

For the structure of the website, we want to start out with the simpler questions that will be quicker to answer. We would take our dataset and split the games into different groups based on genre, memory, and potentially other factors. This way our result can break games into certain groups as to not overpopulate the visualization, but still allow for a good comparison between games.

Our first general design idea is to use a scatter plot for our final comparison of games, price, and positivity ratio. We want to understand the price and positivity to assign a certain “score” value to the video game. Once this score value is assigned, we would then place the game on a scatter plot showing the price, the positivity, and the scores. Our goal would be to break the games into 4 separate groups of:

* Good Reviews and Good Price
* Good Reviews and Bad Price
* Bad Reviews and Good Price
* Bad Reviews and Bad Price

We want to do it this was so to avoid a large cluster of games on one simple scatter plot. If we break apart the games into groups, it will be easier to view the datapoints and interact with points to know which games are in which position.

Our second general idea is to create a matrix alignment using each of our selected rows and columns to give a visualization that combines all the data. We would still need to break apart the matrices into smaller groups as to not completely overcrowd the screen with a large matrix but doing this would allow for either a heat map or scatter plot matrix to be used. Using a heat map we would be able to show each game and all of it’s attributes and show which games have better attributes than others. We would be able to color code the variables so that the heat map does not get too overcrowded with one color, but we could see each game’s affect on the different data points we have pulled. We could also take the heat map and give a value to each specific game again, giving it an overall score and comparing the scores for each game.

Our final visualization idea was to use a scatterplot matrix that takes elements from the first and second ideas. Once again, we would need to separate the games for an easier comparison, and we could do that here through genre, memory, and platform release. We would still be able to show a comparison of each game with respect to their attributes and determine which games would be the best through the attributes. Rather than have a small number of games at the top of a single scatter plot, we could compare the games that appear at the top of each given plot in the matrix. This way, it is easier to determine a large number of games that would be considered “good” either through price or reviews, rather than needing to assign a score and determine whether a game is good both on reviews and price or only one attribute.

**Must-Have Features**

These are the comparisons we feel are the most important to the project’s success. We feel these are good comparisons right now but may change later due to simplicity of only comparing two aspects, or the addition of a new variable we hadn’t considered into the comparison.

* Comparison of Video Games, Price, and Positivity Ratio
* Comparison of Developer/Publisher and Total Number of Reviews
* Comparison of Genre, Developer and Price
* Comparison of Memory Size and Price

**Optional Features**

These are the comparisons we feel are optional but useful visualizations for the website. We feel these are good comparisons right now but may change later due to simplicity of only comparing two aspects, or the addition of a new variable we hadn’t considered into the comparison.

* Comparison of Platform Release and Number of Reviews
* Comparison of Genre and Number of Reviews
* Comparison of Genre and Platform Release

**Project Schedule**

Because the project is scheduled over the next 7 weeks, the below goals for each member of the group are not set in stone. Therefore, some preliminary tasks for each group member are given, but subject to change in case of major breaks in assignment, schedule changes, or other arising challenges.

Week 1 (10/18-10/24): Receive Feedback from project proposal and make necessary changes to the dataset, as well as fix the skeleton to show better visualizations per comments.

* **Jake:** Edit code/website where necessary to make sure that feedback is understood and implemented. Begin working on first visualization/chart for comparing the topics of developer/publisher and the total number of reviews
* **Ryan:** Edit code/website where necessary to make sure that feedback is understood and implemented. Begin working on first visualization/chart for comparing the topics of memory size and price
* **William:**Edit code/website where necessary to make sure that feedback is understood and implemented. Continue updating the script/dataset if needed and work on starting any secondary comparisons such as developer, genre, and price

Week 2 (10/25-10/31): Continue to work on prototype of website. Our goal is to finish the first couple of visualizations by the 27th, then work on the more complicated graphs going into the next week.

* **Jake:**Continue given tasks on the first simpler visualizations, then begin coding the more complicated graphs and visualizations once the group decides what is needed next.
* **Ryan:** Continue given tasks on the first simpler visualizations, then begin coding the more complicated graphs and visualizations once the group decides what is needed next.
* **William:**Continue given tasks on the first simpler visualizations, then begin coding the more complicated graphs and visualizations once the group decides what is needed next.

Week 3 (11/1-11/7): Prototype of website due on the 7th. Finish the final prototype by the 6th and make small changes on the 7th to be turned in.

* **Jake:**Finish up tasks on more complicated visualizations and consult with the group whether more data or graphs should be implemented before submission. Once the group decides what is needed, work to finish for submission.
* **Ryan:** Finish up tasks on more complicated visualizations and consult with the group whether more data or graphs should be implemented before submission. Once the group decides what is needed, work to finish for submission.
* **William:**Finish up tasks on more complicated visualizations and consult with the group whether more data or graphs should be implemented before submission. Once the group decides what is needed, work to finish for submission.

Week 4 (11/8-11/14): Peer review submission due on the 14th. Make sure that we consult as a group our critique of the visualizations, then come together to fix our own critiques.

* **Jake:**Begin working on submission of Visual Encodings for each prototype of the peer review. Once feedback for the prototype is given, we will work on our own respective graphs and fix errors as necessary.
* **Ryan:** Begin working on submission of Interaction Techniques for each prototype of the peer review. Once feedback for the prototype is given, we will work on our own respective graphs and fix errors as necessary.
* **William:**Begin working on submission of Design Quality for each prototype of the peer review. Once feedback for the prototype is given, we will work on our own respective graphs and fix errors as necessary.

Week 5 (11/15-11/21): Continue working on editing the visualizations and reading group feedback to improve.

* **Jake:**Take the comments from the peer review and address areas of concern on assigned visualizations. Once necessary changes are made, continue to polish the website, and begin work on the process book.
* **Ryan:** Take the comments from the peer review and address areas of concern on assigned visualizations. Once necessary changes are made, continue to polish the website, and begin work on the project screencast.
* **William:**Take the comments from the peer review and address areas of concern on assigned visualizations. Once necessary changes are made, continue to polish the website, and add final changes to improve the code/visualizations.

Week 6 (11/22-11/28): Continue working on editing the visualizations and reading group feedback to improve. Our goal is to have the presentation complete by the 28th so that the 29th and beyond can be used for small fixes and polishing.

* **Jake:**Continue work on the assigned visualizations and work towards a rough completion of the process book. Make sure that everything is polished before the presentation.
* **Ryan:** Continue work on the assigned visualizations and work towards a rough completion of the project screencast. Make sure that everything is polished before the presentation.
* **William:**Continue work on the assigned visualizations and work towards a rough completion of the other items as needed. Make sure that everything is polished before the presentation.

Week 7 (11/29-12/5): Oral presentation given on the 30th or 2nd. Final deliverable given on the 5th.

* **Jake:**Take comments from the oral presentation and make necessary changes to the code. Finish the process book. Once complete, meet with the group to discuss everything needed before the final submission of all the deliverables.
* **Ryan:** Take comments from the oral presentation and make necessary changes to the code. Finish the project screencast. Once complete, meet with the group to discuss everything needed before the final submission of all the deliverables.
* **William:**Take comments from the oral presentation and make necessary changes to the code. Once complete, meet with the group to discuss everything needed before the final submission of all the deliverables. Help as needed with final delivery and other items.