# *Steam*® and *Twitch*® Video Game Data Analysis

Kishan Patel, Ryan Stewart, William Johnson, and Luis Gonzalez

## I. Objective and Motivation

The objective of this project is to utilize data analysis and data archaeology techniques to generate clean data sets from two different Steam APIs in order to compare video game interest between the initial period for the COVID-19 pandemic and other recent years were the population was not under quarantine.

COVID-19 changed the way people lived their lives in 2020. The virus, which according to the Center of Disease Control and Prevention (CDC) has killed almost 700,000 people, was able to place the world in a global state of quarantine last year. The virus was able to affect not only health but people's lifestyles as well. The team carries a passion for video games. The situational changes that COVID-19 presented last year, presents an opportunity to make assumptions and test them. The assumption being that the the virus and quarantine have motivated many people to become involved with video games and to spend money in them as well. With important information on how to utilize visual tools and application in Python, the team wishes to explore this idea, and to proof how much COVID-19 affected and benefited the Video Game Industry during the times of quarantine in the year 2020.

## II. Discussion of Data

The data that we collected for our project was relatively simple, but difficult to acquire. We collected a list of available Steam videos games as well as different statistics for each game. These additional statistics include genre, total downloads, active player count, daily player count, etc. Additionally, we found and used a dataset on Twitch viewership per game over a several year time span. For all of our data, the time span we collected from was from a period before COVID-19, as well as during and after COVID-19.

### *A. Obtaining the Data*

*1) Steam API:* The first tool we used to retrieve Steam information was the Steam API. Steam is a digital distribution platform developed by Valve Corporation that focuses primarily on online video game distribution [1]. Additionally, they serve as a platform for small game developer organizations to promote their products. Steam provides an API called Steamworks that developers can integrate some of Steam's more popular features into their games, such as social networking features, achievements, and friend lists [2]. Steam's WEB API is an HTTP-based API that allows users to access many Steamworks features. Unfortunately, we will see how the API is under the assumption that you are trying to integrate into their service and not scrape data.

The Steam API works by allowing users to create a private key through their developer portal. Once you start a key, you can use their service API. Currently, the only API calls they offer are limited to specific video games. The first one available is ISteamNews, where Steam provides methods to fetch news feeds for available games or applications. Following is the ISteamUserStats, where Steam provides methods to bring global stat information by game. The third is ISteamUser, where Steam provides API calls to provide information about Steam users. The last one is the ITFItems_440, specific to the video game Team Fortress 2 and user-item data. Unfortunately, this was not beneficial information from the team's objectives due to the limited availability of video games.

The team worked on a script to attempt and practice scraping any available data that we could retrieve. It quickly became apparent that there would be barriers, such as Cross Origin Resource Sharing (CORS). CORS is an industry standard that gives servers cross-domain access controls [3]. When calling the Steam Web API, we made a cross-origin HTTP request. The XMLHttpRequest originated in our script from our local host domain and requested Steam's domain, but modern browsers restrict cross-origin HTTP requests for security purposes. In other words, the XMLHttpRequest can only make calls to our domain. The security measure is how modern browsers refuse cross-origin HTTP requests.

Furthermore, when trying to do an XMLHttpRequest, we can only make calls to our domain. After further research, the solution was to develop a server to make our client-side requests parse and work with the returned JSON objects. Due to time constraints and limitations in server implementation, the team decided to search for further options for retrieving the required information. Lastly, the problem with dealing with the Steam Web API is that Steam does not allow CORS requests.

*2) Selenium and SteamDB:* The next tool we used in an attempt to collect data was Selenium, a Python module for automating browser behavior. As other Python modules such as urllib and requests have shortcomings due to novel changes in modern web development such as asynchronous JavaScript and dynamic web pages, it becomes difficult to scrape HTML data. Selenium helps mitigate these shortcomings by using a web driver to execute commands, essentially controlling a browser, rather than simply calling HTTP requests. This tool would be used to extract table data of the number of players that played a video game for each month, dating from the date of the game's release up to the current date.

For the video game analysis, we first used urllib with

BeautifulSoup to scrape AppIDs associated with various video game titles from the Steam store's search-by-category web page. To find video game titles that would be a good fit for our data set, the category selected was Sort by Relevance to display popular games and trending games. We used BeautifulSoup to find the table, and traverse through the table rows to find the AppIDs. The AppID collected would be used as a string in the URL of SteamDB, when searching for video games.

Before using Selenium, we had to set up a web driver, so Chrome Driver was used. We also enabled options to the driver to add arguments such as the user agent to ensure that web pages will properly recognize the driver. Testing this driver on different websites, we moved on to testing it on SteamDB and encountered issues with connecting to the webpage.

The first issue we encountered when using Chrome Driver was that CloudFlare prevented the driver from inspecting the page source of SteamDB, since CloudFlare provides services to SteamDB like setting up firewalls and distributed denial-of-service (DDoS) protection. This challenge required us to look for other tools and methods to access the service, so then we used Undetected Chrome Driver to circumvent some of the restrictions imposed by CloudFlare, as this driver does not trigger anti-bot services.

However, we encountered a second issue when using Undetected Chrome Driver. After bypassing Cloudflare's restriction to access, the script from the web page prompted in the page source, "Sorry for the inconvenience, blame the people that keep crawling SteamDB non-stop using residential proxies." Thus, we were unsuccessful at scraping graph data from SteamDB for our data set.

Even if we were successful at accessing the web page, there were other challenges to consider like not getting blocked by the service when collecting thousands to tens of thousands of historical data. Due to how much time was spent trying to scrape data and the many challenges still present, we abandoned using Selenium and SteamDB to look for another method.

*3) Kaggle:* The Twitch data set from Kaggle contained metadata such as year, month, hours watched, average viewers, peak viewers, and the game title, organized as comma-separated value. So for any given game title, there would be data for viewer and streamer data corresponding for each month dating from January 2016 to September 2021.

Although the historical data did not go as far back as we would like, the data was sufficient enough to interpret the overall trend of video games to compare with the years 2020 and 2021, the years that mark the event of COVID-19.

## III. Models and Algorithms Used

For our project, we focused our efforts on the analysis of our data and the conclusions we could draw from it. Because of this, we were unable to find time to apply our data to a model and/or train it with a machine learning algorithm.

## IV. Results

### A. Twitch Dataset

Using matplotlib and the Twitch dataset we pulled from Kaggle, we graphed the total average livestream views on Twitch, the total peak livestream views on Twitch, and the total number of streamers active on Twitch, as shown in Fig. 1.

Each of the graphed data were closely proportional to one another, making this data set a good candidate fit for determining events in each year and how much a certain event impacted engagement on Twitch.

By inspection of the graphed data, there is a big gap in engagement between the years 2017 to 2018 and 2019 to 2020, indicating that there were events that increased the number of engagement. We can also see that there was a decline between the years 2020 to 2021, but as the data for 2021 ends in the month of September, we can assume that the data for the year 2021 will increase.

To better visualize the impact of engagement on Twitch throughout the timeline, we instead plotted each engagement, taking the difference between years, to measure the rate of growth. This is visualized in Fig. 2.

From Fig. 1 and 2, we confirm that the years between 2017 and 2018 and between 2019 and 2020 indicated major events. Similar to the previous graphs displaying total numbers, the graph of growth between years is roughly proportional to one another. However, we can see that streaming grew more than viewing between 2017 and 2018, but declined between 2020 and 2021.

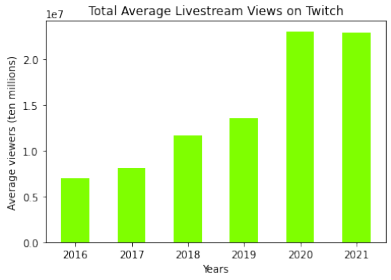
In addition, we see that viewership doubled when transitioning from 2016 and 2017 to 2017 and 2018. Furthermore, we see that viewership quadrupled when transitioning from 2018 and 2019 to 2019 and 2020, indicating that COVID-19 had a massive impact in Twitch streaming.

We suspect that the increase in streaming between 2017 and 2018 was a factor of the release of a major game title, and the decline in streaming between 2020 and 2021 was a factor of quarantine lock-downs subsiding in regions of the world, meaning that streamers are moving back to pre-pandemic life.

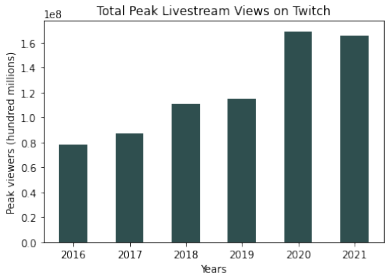
Illustrated in Fig. 3, is a graph of peak views of ten popular games since 2016 to help understand what major game titles influenced the overall growth between 2017 and 2018, and what titles were popular in the years 2020 to 2021.

Using Fig. 3, we can see that the battle royale video game Fortnite, colored in bright red, had its first major peak around month 30, which corresponds to the year 2018, which explains why the growth rate between 2017 and 2018 increased greatly. We can also see that Fortnite peaked the highest around month 52, which would have corresponded to the months of March and April of 2020 in which COVID-19 lock-downs became prevalent.

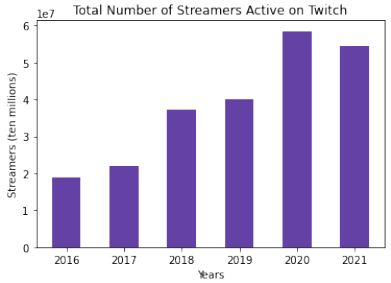
Interestingly, the title that peaked the highest on Twitch was a streaming category called Just Chatting, colored in brown, near the beginning of 2021, indicating the popularity




(a) Total Average Livlivestream Views on Twitch

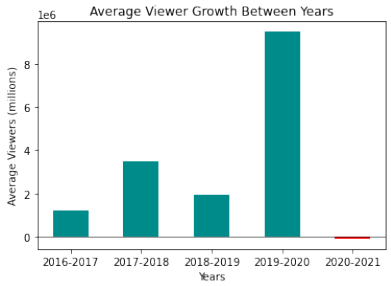



(b) Total Peak Livestream Views on Twitch

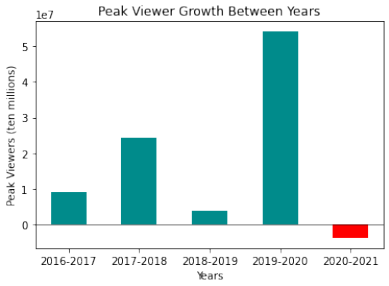



(c) Total Number of Streamers Active on Twitch

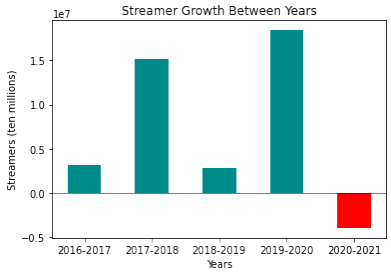
Fig. 1. Twitch Viewship/Streamer Count Metrics




(a) Average Viewer Growth Between Years on Twitch




(b) Peak Viewer Growth Between Years on Twitch




(c) Streamer Growth Between Years on Twitch

Fig. 2. Twitch Growth Metrics

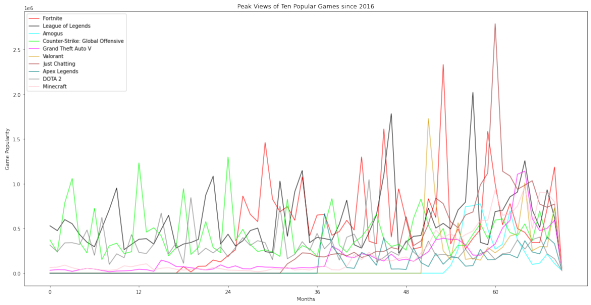



Fig. 3. Peak Viewship of Ten Popular Games Since 2016

of using Twitch as a social platform for streaming rather than just a video games platform.

### B. Steam Dataset

The Steam dataset from Kaggle contains metadata such as game name, year, month, average number of players played at same time, month-to-month difference in average, highest number of played same time, and average peak percentage. This dataset contains 1,200 games on Steam. We had data from 2012 to 2021, so we could look at trends and compare with the year corresponding to COVID-19.

Using plotly, we plotted most games played in 2020 and 2019. To compare pre-lockdown and during lockdown. Wanted to see how big of an impact the COVID-19 lockdown had on the gaming industry.

As we can see in Figures above, some games had their peak early in the pandemic and slowly decreased through the year. Games like Hitman 2 had its peak in April 202 after that month it started to decline but Hitman 2 was a popular game in 2019. Another we

looked at was Counter-Strike: Global Offensive game was played regularly by the gamer before pandemic, but game saw the rise in playing game late 2019 when COVID-19 was starting to hit major countries and in early 2020 game was the most played game on Steam.

We also wanted to look at some individual games like Grand Theft Auto 5 because PC gamers started to use mods like online role play that's not provided by game companies but third party created online servers which got popular during lockdown. Here are the results:

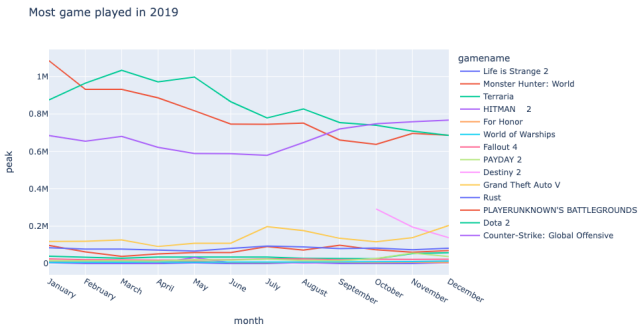
## V. Issues and Limitations

One of the primary issues met during this project was that our team size was large, creating a dependency for each member early in the project. That is, each member depended on one member to get the Steam API working.

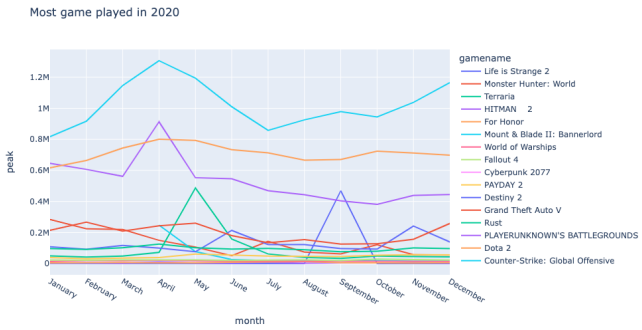
In consequence, the team had to break down responsibilities so that everyone could find their own tools and methods to conduct research for this project. Instead of each member utilizing the Steam API, each member looked for similar tools like SteamSpy, SteamDB, etc.

Another issue met during this project was that many tools and methods were novel in our experience. In other words, each member had to learn how to use a tool and method, and were at times unsuccessful.

Finally, the last issue met during this project was that the team had to resort to a pre-existing dataset from Kaggle, which we discovered around October while searching for data sets for Miniproject 2.

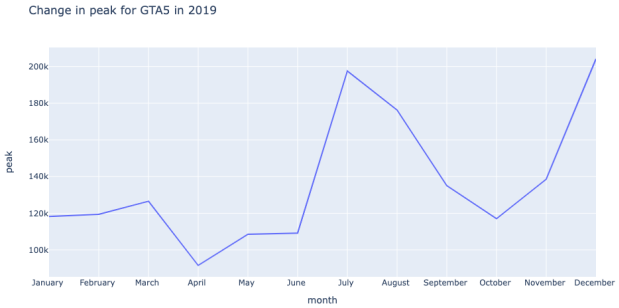



(a) Games Played Most in 2019

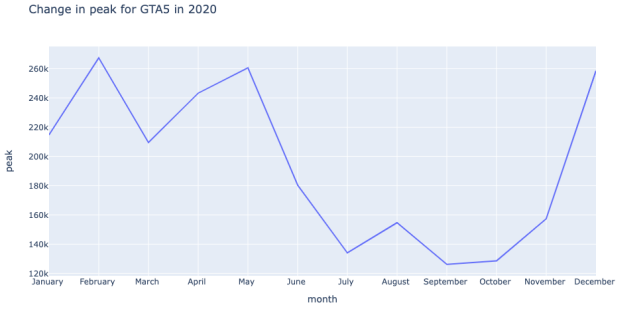



(b) Games Played Most in 2020

Fig. 4. Steam Games Played the Most




(a) Peak User Count for Grand Theft Auto 5 in 2019




(b) Peak User Count for Grand Theft Auto 5 in 2020

Fig. 5. User Counts for Grand Theft Auto 5

While the unsuccessful attempts of scraping data delayed our milestones in this project, the Kaggle datasets allowed the team to finally progress. In addition, although this project did not intend to delve into other video game platforms, we utilized the Twitch data set found on Kaggle to make up for some of the time loss early in this project.

## VI. Future Work

If the team could conduct further research in this video game analysis, we would collect data sets from other video game platforms such as consoles.

Another future work we could conduct is training a model to predict what the rest of 2021 could potentially look like given that the data sets provided from Kaggle included seven months of data for that year. Thus, for one of the data sets, the Twitch data set, we believe that we would see a positive growth rate rather than a negative growth date if we had the full year of data.

Furthermore, training a model could help predict the next five years, but since the team did not have enough historical data, we did not forgo this method.

## VII. Org Chart

### A. Time-line

Our timeline of plans did change due to unsuccessful results from using methods and tools early on in the project. However, we did expand our original plans to include Twitch data.

The original plans we had set up, along with each date, are as follows:

- Request access to API from Steam by **October 8**.
- Request access to API from ISteamUserStats by **October 8**.
- Request access to API from ISteamUser by **October 8**.
- Data cleaning and retrieve data set with help of Numpy by **October 15**.
- Generate list of Apps IDs by **October 29**.
- Generate the most used apps during COVID-19 by **November 5**.
- Generate the plot of the most used apps data by **November 5**.
- Compare data before COVID-19 and during COVID-19 by **November 12**.
- Generate the plot of the comparison data by **November 12**.

The new plans we set up, along with each date, are as follows:

- Request access to API from Steam by **October 8.**
- Request access to API from ISteamUserStats by **October 8**.
- Request access to API from ISteamUser by **October 8**.
- Look for an alternative to retrieve the dataset with Google Dataset Search engine by **October 15**.
- Search for Twitch dataset by **October 29**.
- Generate the most used apps during COVID-19 by **November 5**.

- Generate the plot of the most used played and streamed data by **November 5**.
- Compare data before COVID-19 and during COVID-19 by **November 12**.
- Generate the plot of the comparison data by **November 12**.

### *B. Group Member Responsibilities*

Every member collaborated in creating the Final Presentation and Report.

*1) William Johnson:* Responsible for using Selenium to scrape data from SteamDB and plotting results from the Twitch dataset.

*2) Kishan Patel:* Kishan was reponsible for finding the dataset for Steam and Twitch. Also plotted results from the Steam dataset.

*3) Luis Gonzalez:* Responsible for finding a way to scrape, clean, and develop a dataset from Steam API. Additionally, he collaborated to create presentation and report.

*4) Ryan Stewart:* Responsible for using SteamSpy api in Python to generate list of Steam apps and their corresponding data (except usage data). Additionally, he was responsible for formatting and compiling the Final Report.

## REFERENCES


[1] "Steam Web API Documentation," Steam community. [Online]. Available: https://steamcommunity.com/dev. [Accessed: 30-Oct-2021].

[2] D. Beyer, Steam Web API How-To-Guide. [Online]. Available: https://danbeyer.github.io/steamapi/index.html. [Accessed: 04-Nov-2021].

[3] "Cross-origin resource sharing (CORS) - http: MDN," HTTP | MDN. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS. [Accessed: 02-Nov-2021].

[4] ajay19, "Popular games on Steam Starter," Kaggle, 17-Mar-2021. [Online]. Available: https://www.kaggle.com/ajay19/popular-games-on-steam-starter. [Accessed: 02-Nov-2021].

[5] Ran.Kirsh, "Top games on twitch 2016 - 2021," Kaggle, 20-Mar-2021. [Online]. Available: https://www.kaggle.com/rankirsh/evolution-of-top-games-on-twitch. [Accessed: 22-Nov-2021].