FP_Stat184

Bryan Xiao, Daiwik Kashyap

Table of contents


```
Warning: package 'tidytext' was built under R version 4.4.3
```

Introduction

Playing video games has become a global pastime over the last couple of years, especially during the COV-19 pandemic. In this project, we wanted to look at what similarities and trends we could find in the most popular Steam games. When defining what is a most popular game, we would have liked to use popularity over time, but we could only find reliable data that captured a game's player count for one specific 24 hour period and the game's peak player count. We decided to use the peak player count to determine a game's popularity. Our thought process was that the peak player count meant that there was a large influx of people at one point that were willing to play this game. It was likely in our mind that if so many people were playing this game that it had something that just clicked with people.

Questions to Explore: 1. What genres were the most popular? 2. Does a more positively reviewed game influence the popularity of a game?

Data Provenance

For this project, we used two datasets to explore the trends in the most popular Steam games.The datasets were curated from Kaggle and used Steam's Web APIs to gather data from Steam's official servers. These two datasets were combined to produce the final dataset.

Primary Dataset

Source: Kaggle Description: The primary dataset contains detailed information for Steam Games, including number of positive and negative reviews along with the genres of the game. It has 27,000 rows and 18 columns. Purpose: This dataset was collected to gain a record of games released before and around May 2019. Case: Each row represents an unique game with columns containing information related to the game.

Secondary Dataset

Source: Kaggle Description: The secondary dataset contains information related to the player counts of the 5,000 most played games on Steam. Purpose: This dataset was collected to provide insight in player counts for a variety of Steam games. Case: Each row represents an unique game with columns relating to player counts.

Data Wrangling

We cleaned and merged the two datasets, Steam Store Games and Most Played Games of All Time, and isolated the top 400 games based on peak players. They were merged using the name column in our primary dataset and the Name column in our secondary dataset. Afterwards, we selected the relevant columns like Name, Genres, All Time Peak, Positive Ratings, and Negative Ratings. We decided to use only the top 400 games as once you go below that, their peak player count goes below what we considered popular.

Final Dataset

Description: This final dataset is a dataset that merges the primary and secondary datasets and selects the relevant variables for our analysis. It has 400 rows with 5 columns, which are the relevant variables we want to need to answer our questions. Purpose: This dataset makes it easier for us to analyze common factors across popular Steam games. Case: Each row represents a game with columms relating to its peak player count, genre, and reviews.

FAIR Principles

The datasets that we used adhere to the FAIR principles.

Findable

Both datasets are sourced from Kaggle, which is a well-known platform for datasets,

Accessible

Both datasets are openly available for download on Kaggle, which is a well-known and reliable platform for data.

Interoperable

The datasets are formatted as csv files, which is an universally accepted format for most data analysis tools.

Reusable

The datasets come with clear explanations of what each column is supposed to contain, making it easier for others to use the datasets for their own analysis. By cleaning and tidying the final dataset, we hope that other people may be able to find patterns that we may have missed.

CARE Principles

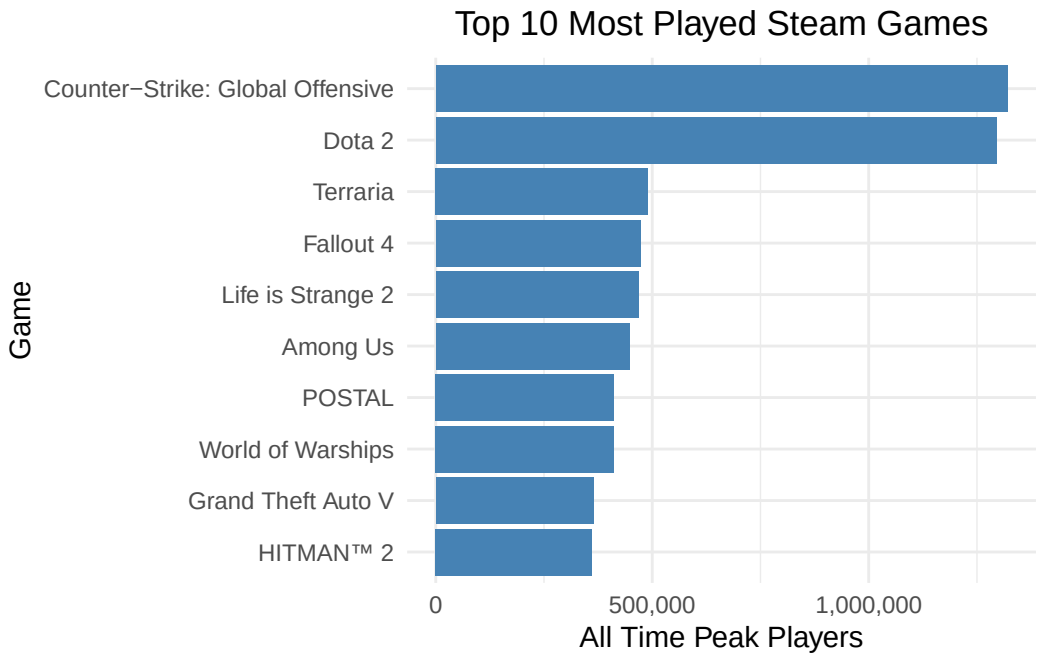
Our data doesn't include any personal or sensitive information, but we did want to maintain ethical standards throughout our project. The datasets are made up of public data available through Steam and our analysis was done with fairness and collective benefit. We also want to make sure that our interpretations of the data are responsible.

```
Warning: Expected 3 pieces. Additional pieces discarded in 106 rows [3, 8, 14, 16, 18,
23, 29, 30, 31, 32, 39, 43, 47, 48, 51, 54, 61, 64, 65, 68, ...].
```

Visualizations

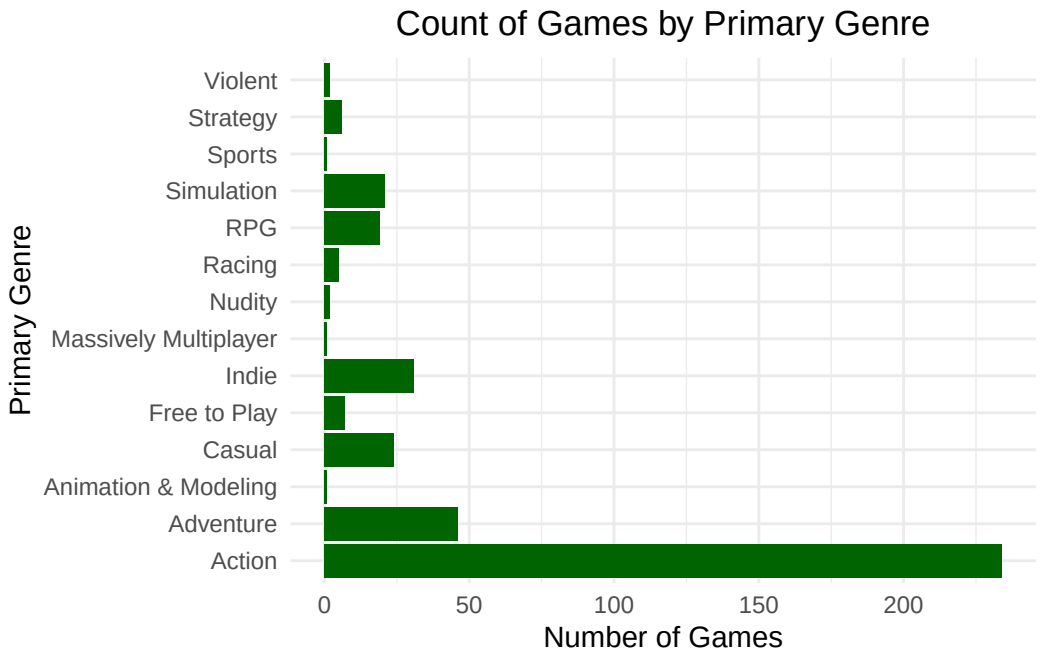
Top 10 Most Played Steam Games

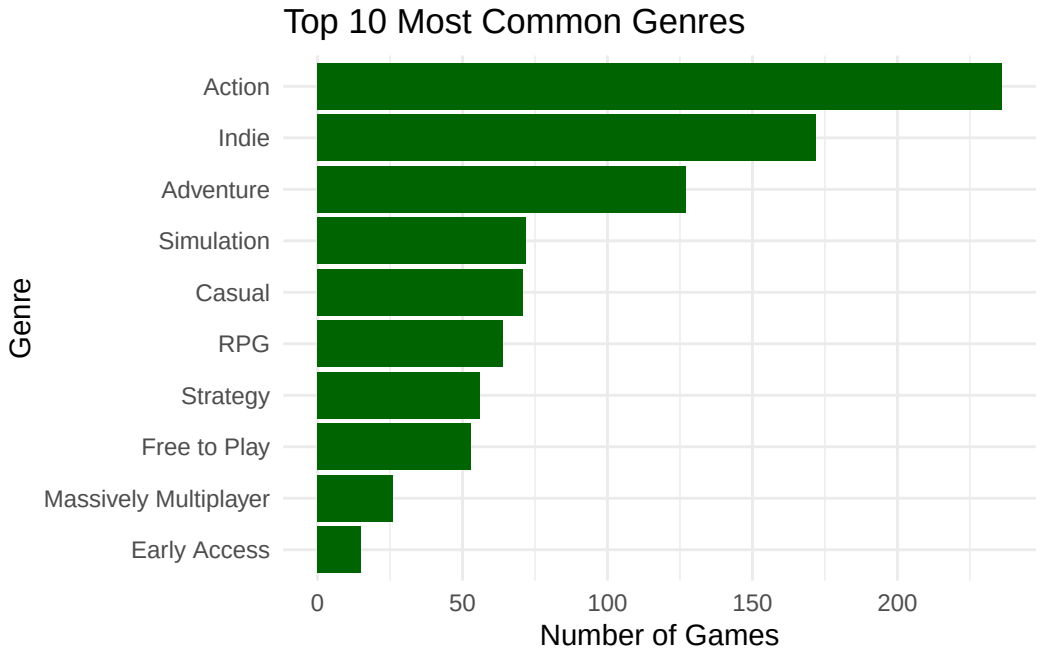
When we were doing our Exploratory Data Analysis, we noticed that there was a huge disparity in peak players between games, even among the top 10 games. The graph below was made to try to visualize the gap in peak player count between the first two games, Counter Strike: Global Offensive and Dota 2, compared to the rest of the top 10 games. The key insight that we want the reader to take away from this graph is that peak player count can have a pretty big gap. This will be something to keep in mind for a graph later on.



Count of Games by Primary Genre

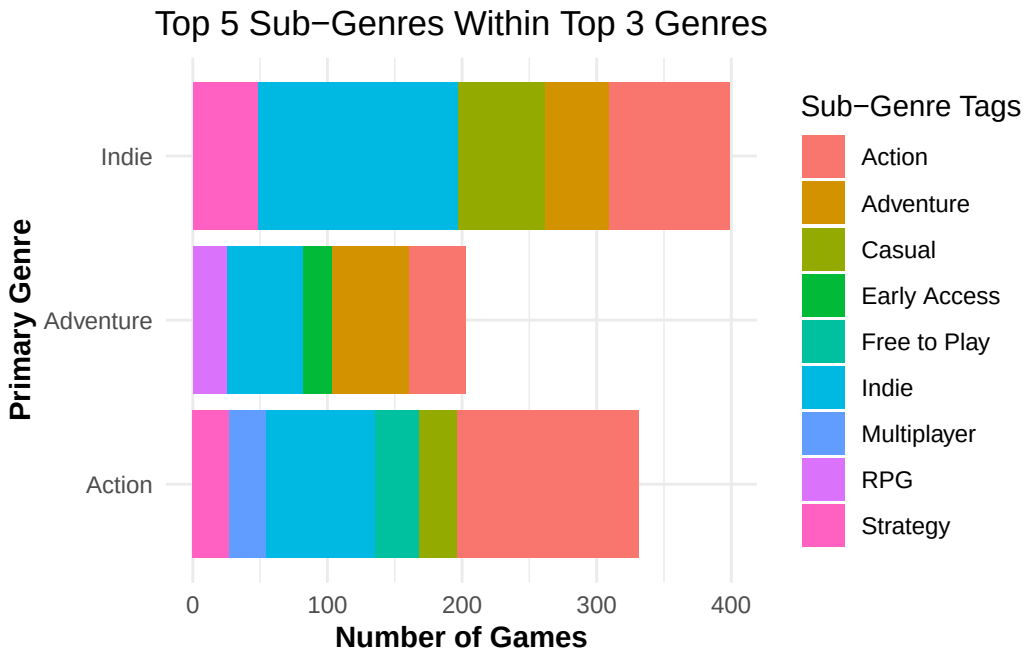
This graph is displaying the amount of games that are listed under their primary genre. You will notice that Action, Adventure, and Indie dominated the number of games with one of the three as their primary genre. So this suggests that these genres may contain something that draws people in. Our first thought was that these genres are the most exciting, since action and adventure would put the player in control of doing something. However, we weren't able to explain why indie was among the top 3 genres. Indie is a shorthand term for independent, usually in reference to a single developer or a small team without support from a large publisher.


Count of Games by Primary Genre
Primary Genre
Violent
Strategy
Sports
Simulation
RPG
Racing
Nudity
Massively Multiplayer
Indie
Free to Play
Casual
Animation & Modeling
Adventure
Action
0
50
100
150
200
Number of Games


Top 10 Most Common Genres
Genre
Action
Indie
Adventure
Simulation
Casual
RPG
Strategy
Free to Play
Massively Multiplayer
Early Access
0
50
100
150
200
Number of Games

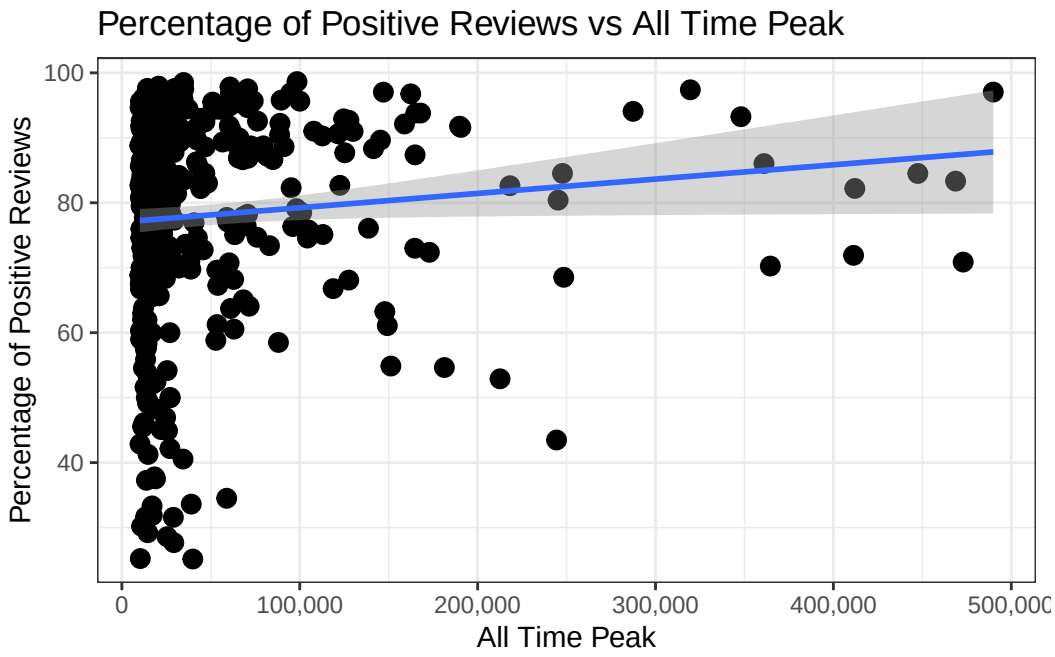
Sub-Genres in Top 3 Genres

One of the comments we received during our Work-In Progress Presentation is if we could split up or subdivide the action category since it looked like it was too broadly defined. To help rectify this issue, we thought it may be best to show the other tags that each game in the genres: Indie, Action, Adventure. We looked at this to break down what other genres were connected to the ones listed above.



Reception to Popularity

Another thing we wanted to explore was whether a higher reception, which would be taking number of positive reviews and dividing it by the sum of positive reviews and negative reviews, and comparing it to the peak player count. Steam only uses a binary system for its review system, so while we would have liked to have a more naunced way to calculate reception, it simply is not possible. We decided after looking at the initial graph to exclude the top 2 games, since their player count expanded out the x-axis so much that it made reading the graph difficult. We hypothesized that the higher reception a game had, the higher its peak player count would be. Our thought process was that a higher reception would make more people recommend the game, which would cause a positive feedback loop. However, our graph only showed a slightly positive relationship between the peak player count and its reception.



Conclusion

Throughout our analysis, we wanted to find any similarities and/or trends in the most popular Steam games. From our analysis, we found that action, adventure, and indie games seemed to draw in the most players. Our analysis also found that player reception is not as important as we initially hypothesized. It only showed a slightly positive relationship between the two variables.

Thank you for taking the time to explore our analysis!

```
# Load necessary libraries
library(tidytext)
library(tidyverse)
library(googlesheets4)
library(ggplot2)
library(dcData)
library(knitr)
library(tinytex)
library(stringr)
library(scales)

# Load datasets
```

```r
MostPlayedDataset <- read.csv("~/GitHub/Sec4_FP_BryanXiao_DaiwikKashyap/data/data.csv",heade
SteamStoreDataset <- read.csv("~/GitHub/Sec4_FP_BryanXiao_DaiwikKashyap/data/steam.csv")
# Wrangle Most Played dataset
MostPlayedDataset$All_time.peak <- str_replace_all(MostPlayedDataset$All_time.peak, ",", "")
MostPlayedDataset$All_time.peak <- as.numeric(as.character(MostPlayedDataset$All_time.peak))

# Making sure SteamStoreDataset and MostPlayedDataset have a common column
SteamStoreDataset <- SteamStoreDataset %>%
  rename(
    Name = name
  )

# Merge datasets and tidy
MergedData <- merge(SteamStoreDataset, MostPlayedDataset)

MergedDataTidy <- MergedData %>%
  arrange(desc(All_time.peak), .by_group = TRUE) %>%
  select("Name", "genres", "All_time.peak", "positive_ratings", "negative_ratings") %>%
  rename(
    Genres = genres,
    All_Time_Peak = All_time.peak,
    Positive_Ratings = positive_ratings,
    Negative_Ratings = negative_ratings
  ) %>%
  slice(1:400) %>%
  separate(
    col = "Genres",
    sep = ";",
    into = c("Genre 1", "Genre 2", "Genre 3"),
    fill = "right"
  )

# Saving merged dataset as CSV file
#write.csv(
#  MergedDataTidy,
#  file = "cleanedData.csv",
#  row.names = TRUE
#)

Top10Games <- MergedDataTidy %>% arrange(desc(All_Time_Peak)) %>% slice(1:10)

ggplot(Top10Games, aes(x = reorder(Name, All_Time_Peak), y = All_Time_Peak)) +
```

```
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(title = "Top 10 Most Played Steam Games",
       x = "Game",
       y = "All Time Peak Players") +
  scale_y_continuous(labels = comma) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))

# Totaling the genres for each game
Top10Genres <- MergedDataTidy %>%
  pivot_longer(cols = starts_with("Genre"), names_to = "GenreType", values_to = "Genre") %>%
  filter(!is.na(Genre)) %>%
  count(Genre, sort = TRUE) %>%
  slice(1:10)

# Making the horizontal bar graph for counts of genres
ggplot(MergedDataTidy, aes(x = `Genre 1`)) +
  geom_bar(fill = "darkgreen") +
  coord_flip() +
  labs(title = "Count of Games by Primary Genre",
       x = "Primary Genre",
       y = "Number of Games") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))


ggplot(Top10Genres, aes(x = reorder(Genre, n), y = n)) +
  geom_col(fill = "darkgreen") +
  coord_flip() +
  labs(
    title = "Top 10 Most Common Genres",
    x = "Genre",
    y = "Number of Games"
  ) +
  theme_minimal()

# Prepare a subset with only top 3 genres
TopGenres <- c("Indie", "Action", "Adventure")

# Filter and explode tags
TopSubGenres <- MergedData %>%
```

```r
  slice(1:400) %>%
  filter(str_detect(genres, paste(TopGenres, collapse = "|"))) %>%
  select(Name, genres, steamspy_tags) %>%
  separate_rows(genres, sep = ";") %>%
  filter(genres %in% TopGenres) %>%
  separate_rows(steamspy_tags, sep = ";") %>%
  filter(steamspy_tags != "") %>%
  group_by(genres, steamspy_tags) %>%
  summarise(count = n(), .groups = "drop") %>%
  arrange(genres, desc(count)) %>%
  group_by(genres) %>%
  slice_max(count, n = 5)  # top 5 sub-genres for each genre


ggplot(TopSubGenres, aes(x = genres, y = count, fill = steamspy_tags)) +
  geom_col() +
  coord_flip() +  # ← Flip axes
  labs(
    title = "Top 5 Sub-Genres Within Top 3 Genres",
    x = "Primary Genre",
    y = "Number of Games",
    fill = "Sub-Genre Tags"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5),
    axis.title.x = element_text(face = "bold"),
    axis.title.y = element_text(face = "bold")
  )

MergedDataReviews <- MergedDataTidy %>%
  slice(3:400) %>%
  select(Name, All_Time_Peak, Positive_Ratings, Negative_Ratings) %>%
  mutate(
    PercentageOfPositiveReviews = Positive_Ratings / (Negative_Ratings + Positive_Ratings) *
  )
ggplot(
  data = MergedDataReviews,
  mapping = aes(
    x = All_Time_Peak,
    y = PercentageOfPositiveReviews
  )
```

```
) + geom_point(size = 3) +
  geom_smooth(method = "lm") +
  labs(
    x = "All Time Peak",
    y = "Percentage of Positive Reviews",
    title = "Percentage of Positive Reviews vs All Time Peak"
  ) + scale_x_continuous(labels = comma) +
  scale_y_continuous(labels = comma) +
  theme_bw()
```