

Mini Project 1: Steam User Analysis of Papers, Please!

Noah Jansen

```
papers_please_reviews <- read_csv("/Users/noahmj123/SDS264F2025/Mini_Project_1/papers_please.csv")  
  
Rows: 11295 Columns: 5  
-- Column specification -----  
Delimiter: ","  
chr (2): app_name, review_text  
dbl (3): app_id, review_score, review_votes  
  
i Use `spec()` to retrieve the full column specification for this data.  
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Introduction

Released in 2013, “Papers, Please!” puts players into the role of an immigration officer in the fictional country of Arstotska in 1982. While checking peoples passports, looking for contraband, and protecting their glorious country from hostile individuals, players must also take care of a family, weigh their morals for citizens with faulty papers, and debate joining the revolution!

Being acclaimed as one of the best games of 2013, “Papers, Please!” has gone on to sell over 5 million copies and even inspired a short film based on the game. Despite having the esteemed “Overwhelmingly Positive” rating on Steam, that doesn’t fully encapsulate how players perceived the game as reviewers have the option of selecting either “recommend” or “don’t recommend”. Through analyzing the written reviews of players, I aim to see how truly players perceive the game and feel about it.

Scored Reviews

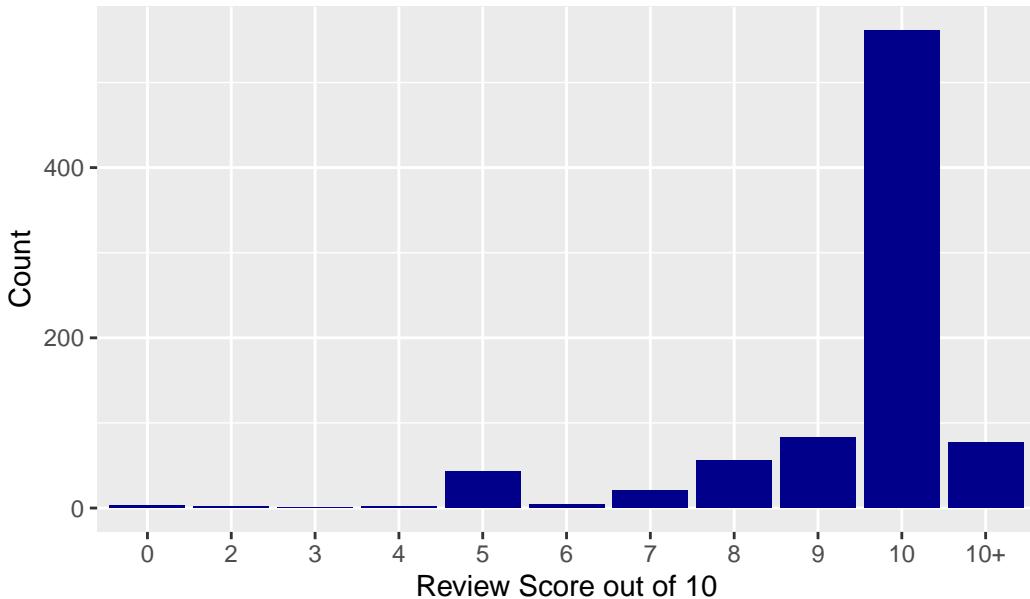
As mentioned, there isn't a lot of room for players to accurately display their emotions of the game through the recommendation system. Regardless, people will put their own scores into their written reviews for people to notice. This can come in many different ways like out of 5 stars, on a scale of 1 to 10, or any numerical scale long enough to provide interesting feedback, but for this instance we will focus on the most popular which is out of 10.

```
paper_num <- papers.Please_Reviews |>
  filter(str_detect(review_text, "[^\\d](\\d)?\\d\\/10[^\\d|\\b]")) |> #finds scores out of 10
#string function number 1: str_detect
#regular expression number 1: [^] anything except
#regular expression number 2: \\d any number
#regular expression number 3: \\b any boundary
  mutate(score = str_extract(review_text, "(\\d)?(\\d)\\/10")) |> #takes only the scores out of 10
#string function number 2: str_extract
  mutate(score = str_extract(score, "(\\d)?(\\d)")) |> #takes score
  mutate(score = as.numeric(score)) |>
  mutate(score = ifelse(score >= 11, "10+", score)) |> #get our over 10 values together to make them 10+
  mutate(score = fct_relevel(score, "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "10+")
```

```
Warning: There was 1 warning in `mutate()` .
i In argument: `score = fct_relevel(...)` .
Caused by warning:
! 1 unknown level in `f`: 1
```

```
#Plot 1 with Alt Text
paper_num |>
  ggplot(aes(x = score)) +
  geom_bar(fill = "darkblue") +
  labs(
    x = "Review Score out of 10",
    y = "Count",
    title = "Review Scores Left by Players Out of 10"
  )
```

Review Scores Left by Players Out of 10



With an overwhelming amount of players scoring the game as a 10 and a small minority of the scores being lower than an 8, “Papers, Please!” shows an impressively positive viewpoint from consumers. There are even people who rated the game higher than a 10, showing how amazing the game truly was to them.

Vulgarity

With honest reviewing comes a lot of vulgarity. Steam implemented an automatic profanity detection system that will censor any vulgar language with the symbol `\u2022` replacing every letter of the word. While negative reviews seem more obvious to having vulgar language, people can also use it in a positive light to display their strong emotion and love for the game. To adjust for this, we can view the proportion of reviews containing vulgar words, and the amount, depending on how the player recommended the game.

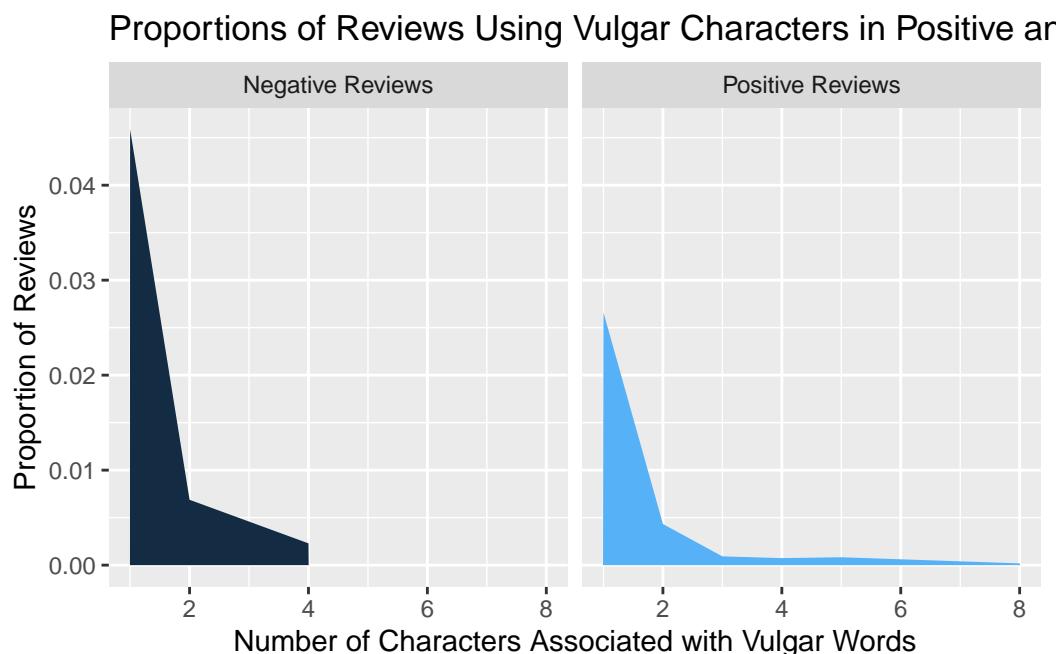
```
rscore.labs <- c("Negative Reviews", "Positive Reviews")
names(rscore.labs) <- c("-1", "1") #rename our rescore values in labels so looks nice for plot

#Plot 2
papers_please_reviews |>
  group_by(review_score) |>
  mutate(vulgar = str_count(review_text, "\u2022 +")) |> #heart expresses censor
  #string function number 3: str_count
```

```

ungroup() |>
group_by(review_score, vulgar) |>
count(amount = n()) |>
ungroup() |>
mutate(amount = ifelse(review_score == -1, 436, 10859)) |> #436 is number of negative reviews
mutate(prop_n = n/amount) |>
filter(vulgar > 0) |>
ungroup() |>
ggplot(aes(x = vulgar, y = prop_n, fill = review_score)) +
geom_area() +
facet_wrap(~review_score, labeller = labeller(review_score = rscore.labs)) +
labs(
  title = "Proportions of Reviews Using Vulgar Characters in Positive and Negative Reviews",
  x = "Number of Characters Associated with Vulgar Words",
  y = "Proportion of Reviews"
) +
theme(legend.position = "none")

```



Overall, the negative reviews show a much higher proportion of their reviews having profanity compared to the positive reviews. With a peak of around 4.5% of reviews containing a vulgar word for negative reviews compared to 2.6% for positive, there is a quite noticeable difference. Despite this, the positive reviews show multiple reviews reaching over 4 vulgar words which isn't recorded once in the negative reviews. This could be possibly due to the much higher

proportion of positive reviews and people loving the game so much they use a ton of vulgar language to describe their passion.

Positive vs. Negative Word Choice

Instead of looking purely at vulgar language, lets take a peek at the choice of words found in reviews. By removing stop words like “and” or “the”, we can look at words with substance in player’s reviews. We can separate the words of substance depending on if they’re considered positive or negative words to see how individuals feel about the game.

```
#plot 3: word cloud

bing_sentiments <- get_sentiments(lexicon = "bing")

tidy_paper <- papers_please_reviews |>
  mutate(review_num = row_number()) |>
  unnest_tokens(word, review_text, token = "words") |>
  anti_join(stop_words) |>
  count(word) |>
  inner_join(bing_sentiments)
```

```
Joining with `by = join_by(word)`
Joining with `by = join_by(word)`
```

```
#Text Analysis number 1: bing_sentiments

#Text Analysis number 2: wordcloud

positive <- tidy_paper |>
  filter(sentiment == "positive")

negative <- tidy_paper |>
  filter(sentiment == "negative")

wordcloud(positive$word,
          positive$n,
          max.words = 50,
          random.order = FALSE,
          random.color = FALSE,
          colors= c("green"))
title("Positive Words in Reviews")
```

Positive Words in Reviews



```
wordcloud(negative$word,
          negative$n,
          max.words = 50,
          random.order = FALSE,
          random.color = FALSE,
          colors= c("red"))
```

Warning in wordcloud(negative\$word, negative\$n, max.words = 50, random.order = FALSE, : depressing could not be fit on page. It will not be plotted.

Warning in wordcloud(negative\$word, negative\$n, max.words = 50, random.order = FALSE, : complicated could not be fit on page. It will not be plotted.

Warning in wordcloud(negative\$word, negative\$n, max.words = 50, random.order = FALSE, : strangely could not be fit on page. It will not be plotted.

Warning in wordcloud(negative\$word, negative\$n, max.words = 50, random.order = FALSE, : bored could not be fit on page. It will not be plotted.

Warning in wordcloud(negative\$word, negative\$n, max.words = 50, random.order = FALSE, : cheap could not be fit on page. It will not be plotted.

```
Warning in wordcloud(negative$word, negative$n, max.words = 50, random.order = FALSE, : cons could not be fit on page. It will not be plotted.
```

```
Warning in wordcloud(negative$word, negative$n, max.words = 50, random.order = FALSE, : slowly could not be fit on page. It will not be plotted.
```

```
Warning in wordcloud(negative$word, negative$n, max.words = 50, random.order = FALSE, : killed could not be fit on page. It will not be plotted.
```

```
Warning in wordcloud(negative$word, negative$n, max.words = 50, random.order = FALSE, : stress could not be fit on page. It will not be plotted.
```

```
title("Negative Words in Reviews")
```

Negative Words in Reviews



While there are negative words like “boring”, “bad”, and “tedious”, a large amount of the negative words pertain to the good qualities of the game itself. For instance, “hard”, “challenging”, and “difficult” all go to show how the game can test a player and see if they can get through the challenges. While these can also be downsides for a player, they are also very positive qualities. Meanwhile, right smack in the middle of the positive words is “fun”, with few other words coming close in repetitions.

Conclusion

With such a high computed score from player's personal reviews, positive word choice to express enjoyment and engagement with the game, and vulgarity used in positive context, "Papers, Please!" has a wide variety of reviews that express the intense emotions players experience and enjoy through their playthroughs. Digging into the textual reviews gave us a good glimpse into the actual words of players compared to a simple recommend or not recommend. With an overwhelming amount of players rating the game a 10 or higher, and a lot of the negative words used to describe the game are mostly complements, "Papers, Please!" is well deserving of the high marks and awards it has received throughout the years. More often than not, "Highly Recommend" is an understatement of a player's true feelings for the game.