BINF5003_FINAL PROJECT

Brenda Soljic, Sam Lenet, Abrar Faruque

November 20, 2025

Dataset Identification

```
The selected dataset for this project is titled Grand Data Auto ViceCity, published by Eshum Malik (202
```

Context of the Dataset The Grand Data Auto ViceCity dataset was sourced from Kaggle, an online data-sharing platform that hosts community-contributed datasets. This specific dataset was created and shared by Eshum Malik in 2023. It compiles publicly available data about user reviews of the video game Grand Theft Auto: Vice City, reflecting patterns of user engagement and review sentiment, as well as voting behaviour within the games community.

```
The tubular dataset is provided in CSV, where each row represents an individual review, and each column

A notable issue with this dataset is the fact that some (298) reviews are blank (null), though this won
```

Due to its structure, the dataset is compatible with R and its standards in data handling libraries, such as dplyr, for example, for exploration and visualization.

Original Analysis and Anticipated Contribution

This analysis will aim to uncover patterns in reviewer behaviour that can inform audience segmentation and engagement strategies for game developers and marketers. This could be done by classifying reviewers into distinct phenotypic groups, investigating correlations between engagement and sentiment metrics, and assessing if temporal patterns or archetypes can predict engagement. The analysis of Grand Data Auto ViceCity will contribute to a broader understanding of gaming communities and user behaviour by translating this data into quantifiable metrics. Ultimately, the project aims to bridge data analytics and behavioural insights, transforming raw community feedback into structured information by applying R's analytical and visualization tools, that support data-driven decision making for game marketers and makers.

Initial installation and Librarys used

```
#install.packages("uwot")
#install.packages("caret", dependencies = TRUE)
#install.packages("summarytools")

library(summarytools)
```

```
## Warning: package 'summarytools' was built under R version 4.5.2
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.5.2
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(uwot)
```

```
## Warning: package 'uwot' was built under R version 4.5.2

## Loading required package: Matrix
```

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

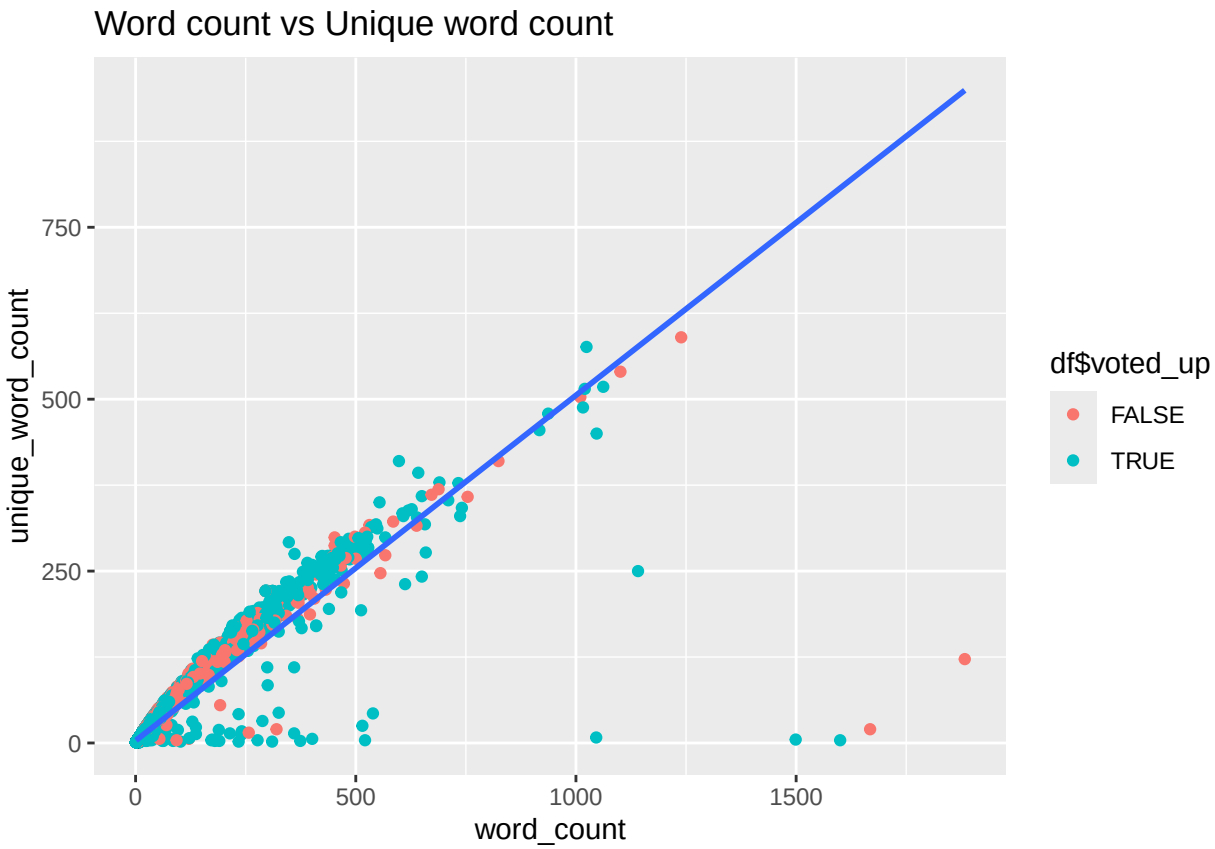
```
library(tidyverse)
```

```
## Warning: package 'purrr' was built under R version 4.5.2

## Warning: package 'stringr' was built under R version 4.5.2

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.1     v stringr   1.6.0
## v lubridate 1.9.4     v tibble    3.3.0
## v purrr     1.2.0     v tidyr     1.3.1
## v readr     2.1.5

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
## x tidyr::pack()   masks Matrix::pack()
## x tidyr::unpack() masks Matrix::unpack()
## x tibble::view()  masks summarytools::view()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Initial Data Wrangling and visualization (trends) of DataSet

```r
df <- read.csv("GTA Vice_city(cleaned).csv", header = TRUE)

#data set variables and metrics
df$id <- as.character(df$id)
df$created_date <- as.Date(df$created_date)
df$author_last_played_date <- as.Date(df$author_last_played)
df <- df %>%
  mutate(voted_up = ifelse(voted_up == "True",TRUE,FALSE)) %>%
  mutate(steam_purchase = ifelse(steam_purchase == "True",TRUE,FALSE))

df_summary <- df %>%
  select(-c(created,author_last_played,steam_purchase))
df_summary <- as.data.frame(dfSummary(df_summary))
write.csv(df_summary, "mydata.csv", row.names = FALSE)

#scater plot of Word count to unique_word_count
ggplot(df,aes(x=word_count,y=unique_word_count))+
  geom_point(aes(colour = df$voted_up))+
  geom_smooth(method="lm", se =FALSE)+
  labs(title="Word count vs Unique word count",
       x="word_count",
       y="unique_word_count")
```

```
## Warning: Use of ‘df$voted_up‘ is discouraged.
## i Use ‘voted_up‘ instead.

## ‘geom_smooth()‘ using formula = ’y ~ x’
```




```
neg_review <- df %>%
  filter(df$voted_up==FALSE)
df$group <- "NO"
df$group[df$word_count < 700 & df$unique_word_count < 175] <- "Subgroup 1"
df$group[df$word_count < 700 & df$unique_word_count < 300] <- "Subgroup 2"
df$group[df$word_count >= 700 & df$unique_word_count < 300] <- "Subgroup 3"
df$group[df$word_count >= 700 & df$unique_word_count > 300] <- "Subgroup 4"

#Proportion of Subgroup in dataset
subgroup1 <- df %>%
  filter(df$group=="Subgroup 1")
subgroup2 <- df %>%
  filter(df$group=="Subgroup 2")
subgroup3 <- df %>%
  filter(df$group=="Subgroup 3")
subgroup4 <- df %>%
  filter(df$group=="Subgroup 4")

print(length(subgroup1$group)/length(df$group))
```

```
## [1] 0
```

```
print(length(subgroup2$group)/length(df$group))
```

```
## [1] 0.9991506
```

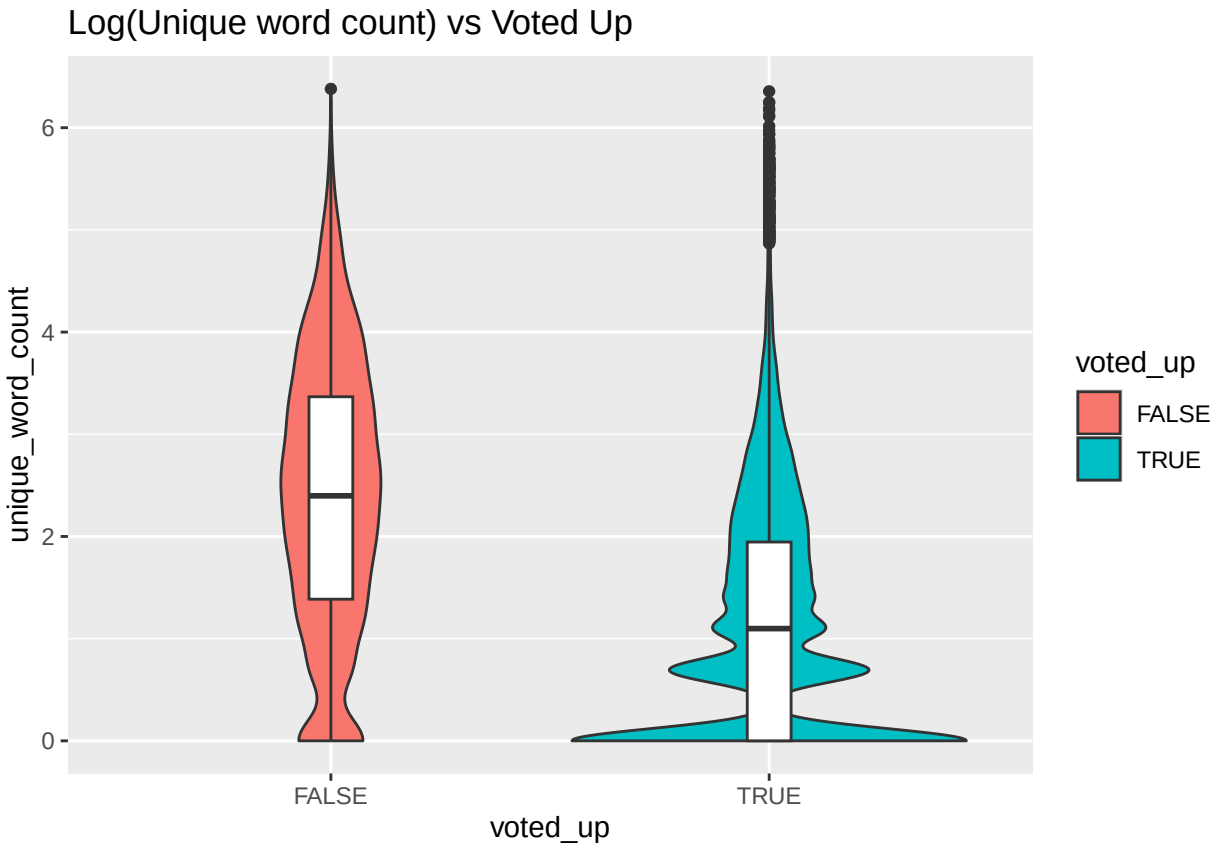
```r
print(length(subgroup3$group)/length(df$group))
```

```
## [1] 0.0001158279
```

```r
print(length(subgroup4$group)/length(df$group))
```

```
## [1] 0.0003088743
```

```r
#box plot of unique word count and pos/neg review
ggplot(df, aes(x=voted_up,y=log(unique_word_count)))+
  geom_violin(aes(fill=voted_up))+
  geom_boxplot(width=0.1)+
  labs(title="Log(Unique word count) vs Voted Up",
       x="voted_up", y="unique_word_count")
```




```r
pos_review <- df %>%
  filter(df$voted_up==TRUE)

print(mean(log(pos_review$unique_word_count)))
```

```
## [1] 1.165279
```

```r
print(sd(log(pos_review$unique_word_count)))
```

```
## [1] 1.134085
```

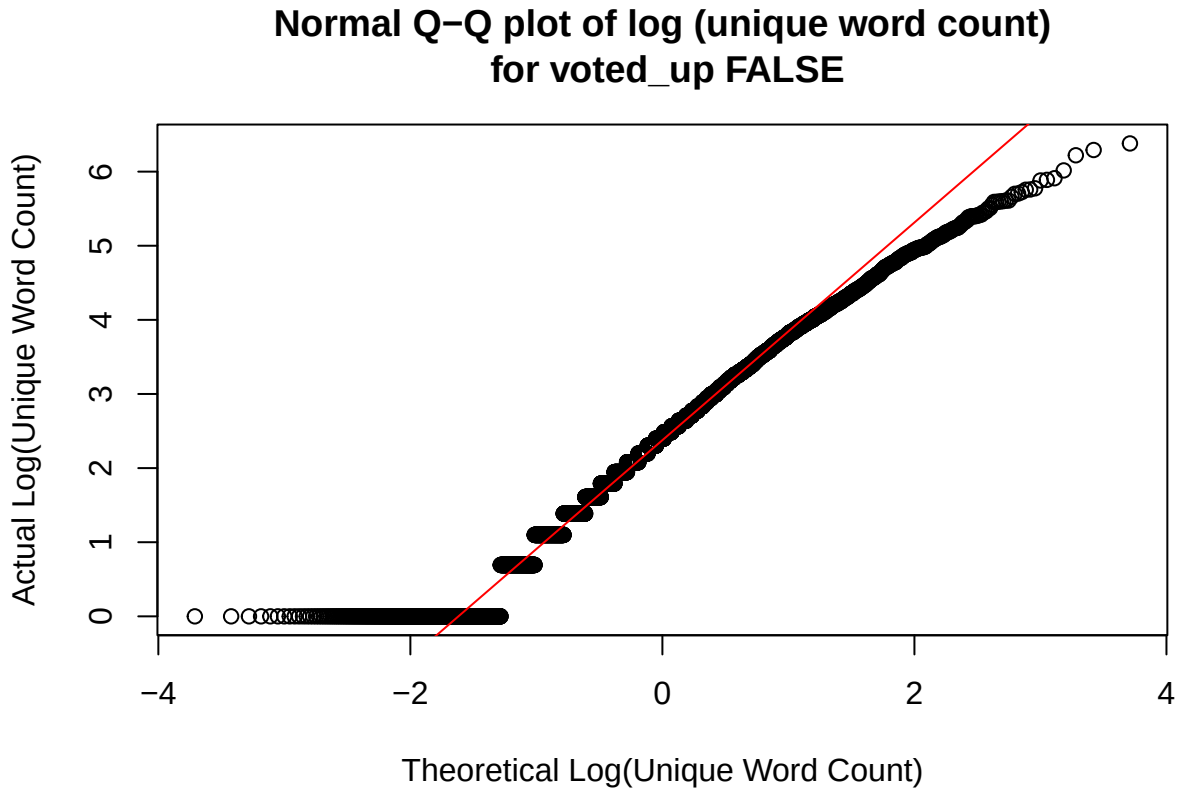
```r
print(mean(log(neg_review$unique_word_count)))
```

```
## [1] 2.378818
```

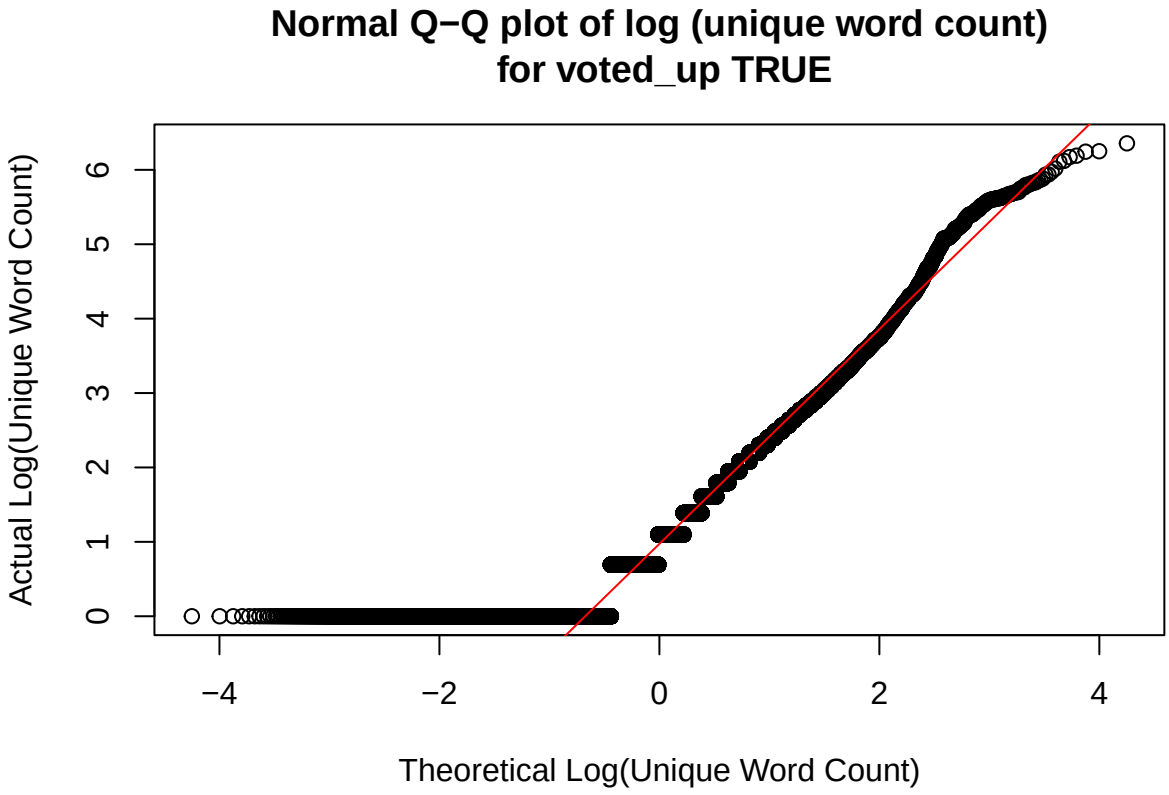
```r
print(sd(log(neg_review$unique_word_count)))
```

```
## [1] 1.34633
```

```r
qqnorm(log(neg_review$unique_word_count), main = "Normal Q-Q plot of log (unique word count)\n for voted
       xlab = "Theoretical Log(Unique Word Count)", ylab="Actual Log(Unique Word Count)")
qqline(log(neg_review$unique_word_count), col = "red")
```




```r
qqnorm(log(pos_review$unique_word_count), main = "Normal Q-Q plot of log (unique word count)\n for voted
       xlab = "Theoretical Log(Unique Word Count)", ylab="Actual Log(Unique Word Count)")
qqline(log(pos_review$unique_word_count), col = "red")
```




```
OUWC_pos_review <- pos_review %>%
  filter(unique_word_count==1)
print(nrow(OUWC_pos_review)/nrow(pos_review))
```

```
## [1] 0.3295012
```

```
#Reviews are collected from 2023-06-27 to 2024-02-01
print(min(df$created_date))
```

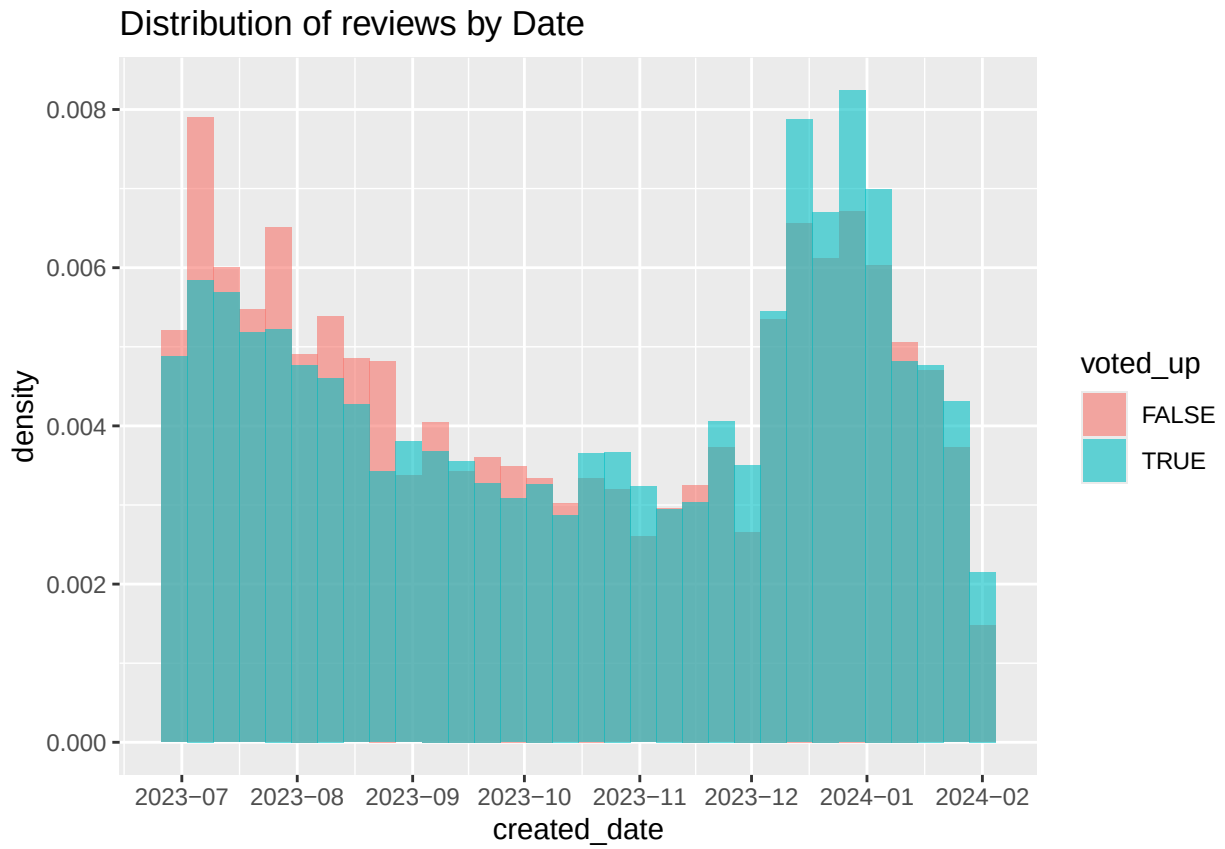
```
## [1] "2023-06-27"
```

```
print(max(df$created_date))
```

```
## [1] "2024-02-01"
```

```
ggplot(df, aes(x=created_date, y = ..density..))+
  geom_histogram(aes(fill=voted_up),alpha=0.6, position = 'identity',binwidth = 7)+
  scale_x_date(date_breaks = "1 month", date_labels = "%Y-%m")+
  ggtitle("Distribution of reviews by Date")
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
df_Dec <- df %>%
  filter(created_date>="2023-12-01" & created_date<"2024-01-01")
df_Nov <-df %>%
  filter(created_date>="2023-11-01" & created_date<"2023-12-01")
print(nrow(df_Dec)/nrow(df_Nov))
```

```
## [1] 2.080777
```

Based on the trends observed above, further data wrangling was done to remove categorical and redundant variables to perform PCA and UMAP analysis PCA analysis

```
df_short <- df %>%
  select(-c(author_last_played,voted_up,steam_purchase))
df_short <- df_short[ , 5:13]

cor_mat <- cor(df_short, use = "pairwise.complete.obs")
high_cor_idx <- findCorrelation(cor_mat, cutoff = 0.9)
colnames(df_short)[high_cor_idx]
```

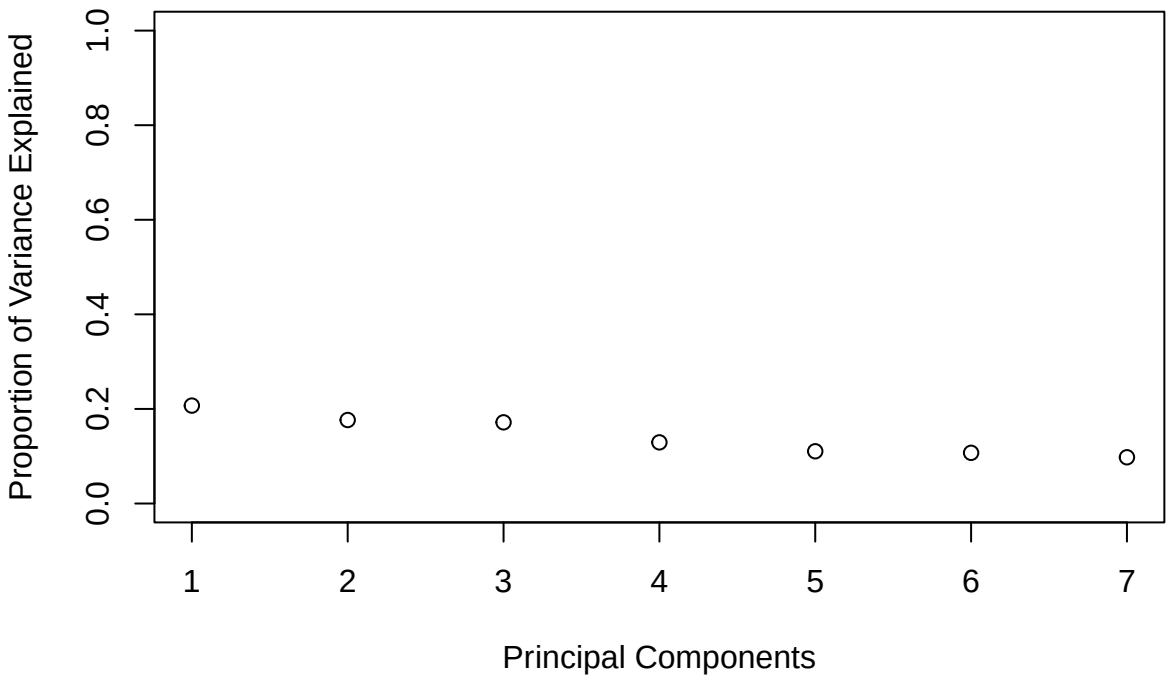
```
## [1] "author_playtime_forever" "unique_word_count"
```

```
df_short <- df_short %>%
  select(-c(author_playtime_at_review, word_count))
```

```r
pca_df <- prcomp(df_short, scale = TRUE)

#Number of PC
length(pca_df)
```

```
## [1] 5
```

```r
#There are 5 principal components

variance_df = pca_df$sdev^2
variance_df / sum(variance_df) -> prop_df

# plot variance in a scree plot
plot(prop_df,
     xlab = "Principal Components",
     ylab = "Proportion of Variance Explained",
     ylim = c(0, 1))
```

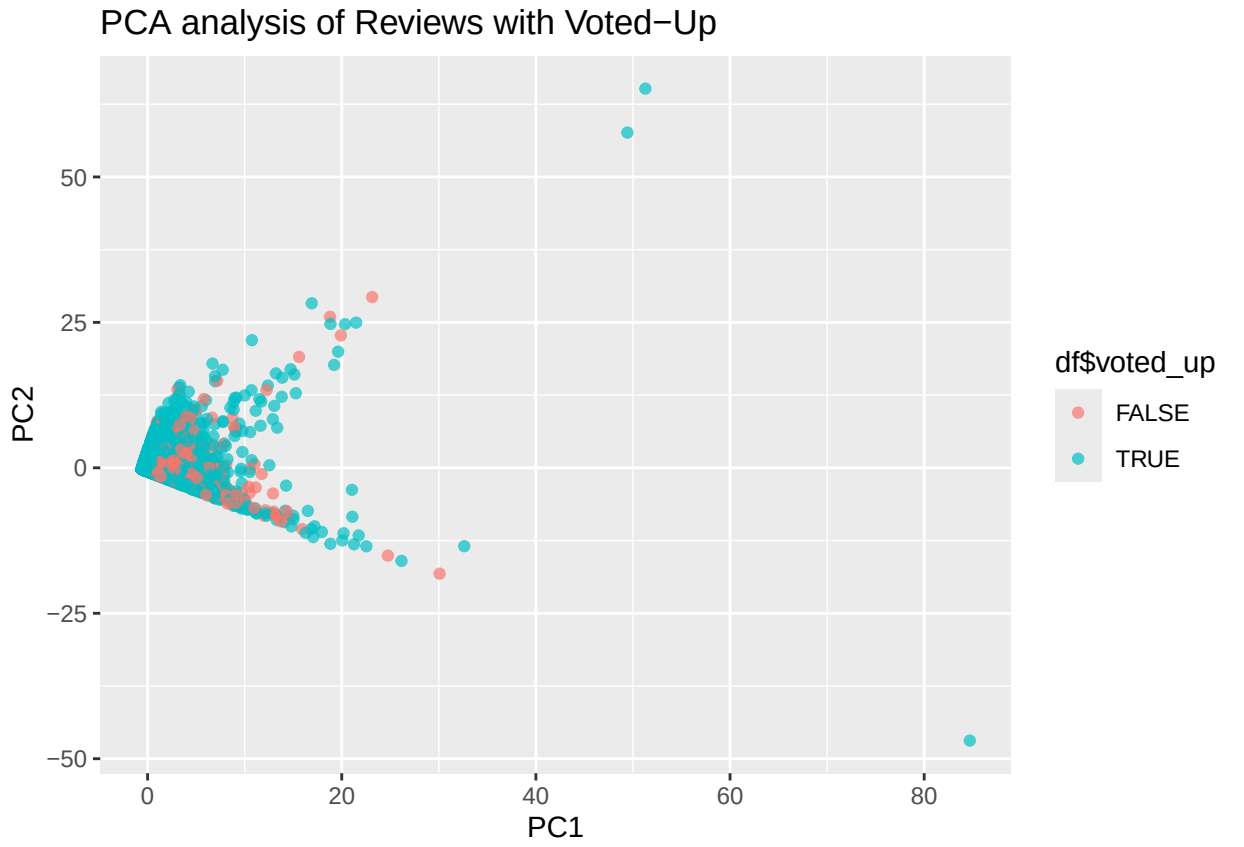



```r
#percentage variance for each PC
print((variance_df[1]+variance_df[2])/sum(variance_df))
```

```
## [1] 0.383603
```

Ploting PC1 and PC2

```
ggplot(pca_df$x, aes(x = PC1, y = PC2)) +
  geom_point(aes(colour = df$voted_up),alpha=0.7)+
  labs(title="PCA analysis of Reviews with Voted-Up")
```




```
#5 variables most strongly contributing to the positive values in PC1?
loadings_pc1 <- pca_df$rotation[, "PC1"]
sorted_loadings_pc1 <- sort(abs(loadings_pc1), decreasing = TRUE)
print(head(sorted_loadings_pc1,n=5))
```

```
##     author_num_reviews author_num_games_owned      unique_word_count
##              0.5413257              0.5381813              0.4322307
##          comment_count               votes_up
##              0.2958335              0.2739969
```

PCA plot gives a fan like structure, from this plot we can determine that PC1 explains the majority of the variance and PC2 explains some or very little of the variance.The variables most variables most strongly contributing to the positive values in PC1 are: author_num_reviews, author_num_games_owned, unique_word_count,comment_count, and votes_up. Since there was no clear seperation between groups, we can assume that the data is not likely linear.

UMAP analysis

```
umap_res <- umap(df_short, n_neighbors = 30, min_dist = 0.1, metric = "cosine")

umap_df <- as.data.frame(umap_res)
```

```r
umap_df$word_count <- df$word_count
umap_df$unique_word_count <- df$unique_word_count
umap_df$group <- df$group
umap_df$voted_up <- df$voted_up

ggplot(umap_df, aes(V1, V2)) +
  geom_point(alpha = 0.6, aes (color = group))
```