Final Project

Dennis Rapoport macid: rapopord

6/4/2020

Contents

Introduction **2**

Data Wrangling Plan **3**

Dataset 1 3

Set-up the R enviorment 3

Iteration 1 3

Iteration 2 4

Iteration 3 4

Iteration 4 5

Dataset 2 6

Dataset 3 7

Disussion **8**

Modifications 8

Results 8

Sentiment Score vs. User Score 8

Distribution of User Scores 19

Conclusion **24**

References **24**

Introduction

This project attempts to explore different patterns in the behaviour of Amazon users by analysing their reviews. In order to identify these pattens the *Amazon Product data* dataset by Julian McAuley (He & McAuley (2016)) will be used. This dataset contains categorized product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014. In it each user review entry contains text of the review, user overall score, time of the review, as well as other metrics. For this work only a part of the data will be used for analysis, more specifically only reviews in the categories: *Patio, Lawn and Garden*, *Amazon Instant Video* (today know as *Amazon Prime Video*) and *Musical Instruments.* The reason for picking these lies behind the computational power of my computer, the size of the other review datasets is too large for it to process, mentioned categories are the smallest sets provied, yet contain over 50,000 reviews all together.

One of the things that we are able to do with this dataset is Sentiment Analysis of the reviews written section, it would allow us to compare users attitude towards a product and how it appliees to the numerical quntificaton by the user. Let's define a Comments Sentiment Score to be an average `AFFIN` lexicon score of all the the words in the review that have a sentiment and User Score to be users rating of the product. By calculating Sentiment Score metric we would be able to compare it the User Score of a review. I suspect that there would be diffent results between the categories.

Another metric of interest would be how the users behaviour changes over time. More in detail, I would like to look at the percentage ratio of the 1-5 score reviews over time and how it changes among differnt categories.

Data Wrangling Plan

Dataset 1

Set-up the R enviorment

```
library(knitr)
library(magrittr)
library(tidyverse)
library(jsonlite)
library(tidytext)
library(textdata)
library(wordcloud)
library(lubridate)
library(patchwork)

## Rmd chunk options
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE)
```

Iteration 1

This iteration focuses on converting raw data into `tidy` format.

a. Read the json file into R data.
b. Drop unncessary columns.
c. Rename columns into appropriate format.
d. Convert `r_date` to `<date>` type.

```
## Iteration 1 ------------------------------------------------------------
## a.
tib_data1 <- fromJSON("Instant_Video.json")

tib_data1 %>% glimpse
```

```
## Observations: 37,121
## Variables: 9
## $ reviewerID     <chr> "A11N155CW1UV02", "A3BC8O2KCL29V2", "A60D5HQFOTSOM",...
## $ asin           <chr> "B000H00VBQ", "B000H00VBQ", "B000H00VBQ", "B000H00VB...
## $ reviewerName   <chr> "AdrianaM", "Carol T", "Daniel Cooper \"dancoopermed...
## $ helpful        <list> [<0, 0>, <0, 0>, <0, 1>, <0, 0>, <1, 1>, <12, 12>, ...
## $ reviewText     <chr> "I had big expectations because I love English TV, i...
## $ overall        <dbl> 2, 5, 1, 4, 5, 5, 3, 3, 5, 3, 4, 4, 3, 3, 5, 2, 3, 3...
## $ summary        <chr> "A little bit boring for me", "Excellent Grown Up TV...
## $ unixReviewTime <int> 1399075200, 1346630400, 1381881600, 1383091200, 1234...
## $ reviewTime     <chr> "05 3, 2014", "09 3, 2012", "10 16, 2013", "10 30, 2...
```

```
tib_data1 <- tib_data1 %>%
  ## b.
  select(reviewTime, overall, reviewText) %>%
  ## c.
```

```
  rename(r_date = reviewTime, r_score = overall, r_text = reviewText) %>%
  ## d.
  mutate(r_date = parse_date(r_date, "%m %d, %Y"))
```

Iteration 2

This iteration focuses on removing inappropriate data

a. Check for NA values and inapropriate dates.
b. Add `r_id` column.

- Dates should be between May 1996 and July 2014.

```
## Iteration 2 -------------------------------------------------------------

## a.
tib_data1 %>% summary
```

```
##      r_date              r_score         r_text
##  Min.   :2000-11-29   Min.   :1.00   Length:37121
##  1st Qu.:2013-05-06   1st Qu.:4.00   Class :character
##  Median :2013-11-21   Median :5.00   Mode  :character
##  Mean   :2013-08-18   Mean   :4.21
##  3rd Qu.:2014-03-07   3rd Qu.:5.00
##  Max.   :2014-07-23   Max.   :5.00
```

```
## b.
tib_data1 <- tib_data1 %>%
  mutate(r_id = 1:nrow(tib_data1))
```

Iteration 3

This iteration focuses on the Sentiment Analysis.

a. Create a tibble in one-token-per-row format.
b. Remove stop words and numbers.
c. Perform Sentiment Analysis. Irizarry (2019)
d. Remove tokens with NA sentiment values.
e. Convert `affin` sentiment values into 0-5 scale.

```
## Iteration 3 ------------------------------------------------------------

## a.
tib_data_tokens1 <- tib_data1 %>%
  unnest_tokens(word, r_text)

## Start of citation
## b.
re_digits <- "^\\d+(?![:alnum:])"
data("stop_words")
```

```r
tib_data_tokens1 <- tib_data_tokens1 %>%
  anti_join(stop_words) %>%
  filter(!str_detect(word, re_digits))

## The following chunk of code is taken from the course textbook Ch. 26 {
## c.
tib_afinn <- get_sentiments("afinn")
tib_data_tokens1 <- left_join(tib_data_tokens1, tib_afinn, by = "word") %>%
  ## d.
  ## }
  filter(!is.na(value)) %>%
  ## e.
  mutate(value = (value + 5) / 2)
```

Iteration 4

This iteratoin focuses on finalzing data and creating additional tibbles for visualizing data.

a. Determine sentiment score.
b. Join all data.
c. Create factors.
d. Add `s_dif` metric.
e. Create an additional tibble that will store the distribution of the user scores.

```r
## Iteration 4 -------------------------------------------------------------
## a.
tib_sentiment_score1 <- tib_data_tokens1 %>%
  group_by(r_id) %>%
  summarise(avg_sentiment = mean(value))

## b.
tib_data1 <- left_join(tib_data1, tib_sentiment_score1, by="r_id") %>%
  filter(!is.na(avg_sentiment)) %>%
  ## c.
  mutate(r_date = as_factor(year(r_date)),
         r_score_fct = as_factor(r_score),
         ## d.
         s_dif = r_score - avg_sentiment)
## e.
data_plot1 <- tibble()
for (year in 2007:2014){
  tib_stat_tmp <- tib_data1 %>%
    filter(r_date == year) %>%
    .$r_score_fct %>%
    fct_count() %>%
    add_column(year = year) %>%
    mutate(prc = n/sum(n))
  data_plot1 <- bind_rows(data_plot1, tib_stat_tmp)
}
data_plot1 <- data_plot1 %>%
  mutate(year = as_factor(year))
```

```r
tib_data1 %>% glimpse
```

```
## Observations: 33,627
## Variables: 7
## $ r_date        <fct> 2014, 2012, 2013, 2013, 2009, 2011, 2013, 2013, 2014,...
## $ r_score       <dbl> 2, 5, 1, 4, 5, 5, 3, 3, 5, 3, 4, 4, 3, 3, 5, 2, 3, 3,...
## $ r_text        <chr> "I had big expectations because I love English TV, in...
## $ r_id          <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16...
## $ avg_sentiment <dbl> 2.500000, 3.250000, 1.000000, 1.750000, 2.846154, 2.0...
## $ r_score_fct   <fct> 2, 5, 1, 4, 5, 5, 3, 3, 5, 3, 4, 4, 3, 3, 5, 2, 3, 3,...
## $ s_dif         <dbl> -0.5000000, 1.7500000, 0.0000000, 2.2500000, 2.153846...
```

Dataset 2

```r
tib_data2 <- fromJSON("Musical_Instruments.json")
tib_data2 <- tib_data2 %>%
  select(reviewTime, overall, reviewText) %>%
  rename(r_date = reviewTime, r_score = overall, r_text = reviewText) %>%
  mutate(r_date = parse_date(r_date, "%m %d, %Y"),
         r_id = 1:nrow(tib_data2))

tib_data_tokens2 <- tib_data2 %>%
  unnest_tokens(word, r_text)

tib_data_tokens2 <- tib_data_tokens2 %>%
  anti_join(stop_words)

tib_data_tokens2 <- left_join(tib_data_tokens2, tib_afinn, by = "word") %>%
  filter(!is.na(value)) %>%
  mutate(value = (value + 5) / 2)

tib_sentiment_score2 <- tib_data_tokens2 %>%
  group_by(r_id) %>%
  summarise(avg_sentiment = mean(value))

tib_data2 <- left_join(tib_data2, tib_sentiment_score2, by="r_id") %>%
  filter(!is.na(avg_sentiment)) %>%
  mutate(r_date = as_factor(year(r_date)),
         r_score_fct = as_factor(r_score),
         s_dif = r_score - avg_sentiment)

data_plot2 <- tibble()
for (year in 2009:2014){
  tib_stat_tmp <- tib_data2 %>%
    filter(r_date == year) %>%
    .$r_score_fct %>%
    fct_count() %>%
    add_column(year = year) %>%
    mutate(prc = n/sum(n))
  data_plot2 <- bind_rows(data_plot2, tib_stat_tmp)
}
```

```
data_plot2 <- data_plot2 %>%
  mutate(year = as_factor(year))
```

Dataset 3

```
tib_data3 <- fromJSON("Patio_Lawn_and_Garden.json")
tib_data3 <- tib_data3 %>%
  select(reviewTime, overall, reviewText) %>%
  rename(r_date = reviewTime, r_score = overall, r_text = reviewText) %>%
  mutate(r_date = parse_date(r_date, "%m %d, %Y"),
         r_id = 1:nrow(tib_data3))

tib_data_tokens3 <- tib_data3 %>%
  unnest_tokens(word, r_text)

tib_data_tokens3 <- tib_data_tokens3 %>%
  anti_join(stop_words)

tib_data_tokens3 <- left_join(tib_data_tokens3, tib_afinn, by = "word") %>%
  filter(!is.na(value)) %>%
  mutate(value = (value + 5) / 2)

tib_sentiment_score3 <- tib_data_tokens3 %>%
  group_by(r_id) %>%
  summarise(avg_sentiment = mean(value))

tib_data3 <- left_join(tib_data3, tib_sentiment_score3, by="r_id") %>%
  filter(!is.na(avg_sentiment)) %>%
  mutate(r_date = as_factor(year(r_date)),
         r_score_fct = as_factor(r_score),
         s_dif = r_score - avg_sentiment)

data_plot3 <- tibble()
for (year in 2006:2014){
  tib_stat_tmp <- tib_data3 %>%
    filter(r_date == year) %>%
    .$r_score_fct %>%
    fct_count() %>%
    add_column(year = year) %>%
    mutate(prc = n/sum(n))
  data_plot3 <- bind_rows(data_plot3, tib_stat_tmp)
}
data_plot3 <- data_plot3 %>%
  mutate(year = as_factor(year))
```

Disussion

Modifications

`I1d.` Converted `r_date` in `tib_data` to `<date>` type to follow `tidy` format and in order to check if all the review dates fall within desired range.
`I2b.` I added `r_id` column to `tib_data` so I could uniqley identify each comment after I breakdown the dataset into one-token-pre-row format in `I3a` and summarize the data in `I4b`.
`I3a`. Creating a new tibble `tibble_data_tokens` to perfom Sentiment Analysis.
`I3c`. Removing stopword tokens from `tibble_data_tokens` for Sentiment Analysis.
`I3e`. I convert `AFFIN` sentiment values into 0-5 scale in order to match `r_score` scale.
`I4a`. I created `tib_sentiment_score` to store comment's sentiment score. Sentiment score of a comment is the mean value of all token sentiment values in a comment.
`I4c`. I add additional `r_score_fct` and converter `r_date` to factor to simplify use with `ggplot2`.
`I4d`. I created `s_diff` metric to reflect the differnce between Sentiment Score and User Score.

Results

Sentiment Score vs. User Score

```
p1 <- tib_data1 %>%
  ggplot(aes(x = r_score_fct, y = s_dif)) +
  geom_boxplot() +
  geom_hline(yintercept = 0.0, linetype="dotdash") +
  ylim(-3.5, 5.5) +
  scale_y_continuous(breaks = seq(-4, 5)) +
  theme_classic() +
  labs( x = "User Score",
        y = "Score Difference",
        title = "Instant Video")

p2 <- tib_data2 %>%
  ggplot(aes(x = r_score_fct, y = s_dif)) +
  geom_boxplot() +
  geom_hline(yintercept = 0.0, linetype="dotdash") +
  ylim(-4, 6) +
  scale_y_continuous(breaks = seq(-4, 5)) +
  theme_classic() +
  labs( x = "User Score",
        y = " Score Difference",
        title = "Musical Instruments")

p3 <- tib_data3 %>%
  ggplot(aes(x = r_score_fct, y = s_dif)) +
  geom_boxplot() +
  geom_hline(yintercept = 0.0, linetype="dotdash") +
  ylim(-3.5, 5.5) +
  scale_y_continuous(breaks = seq(-4, 5)) +
  theme_classic() +
  labs( x = "User Score",
        y = "Score Difference",
```

```
        title = "Lawn and Garden")

p1 | p2 | p3
```

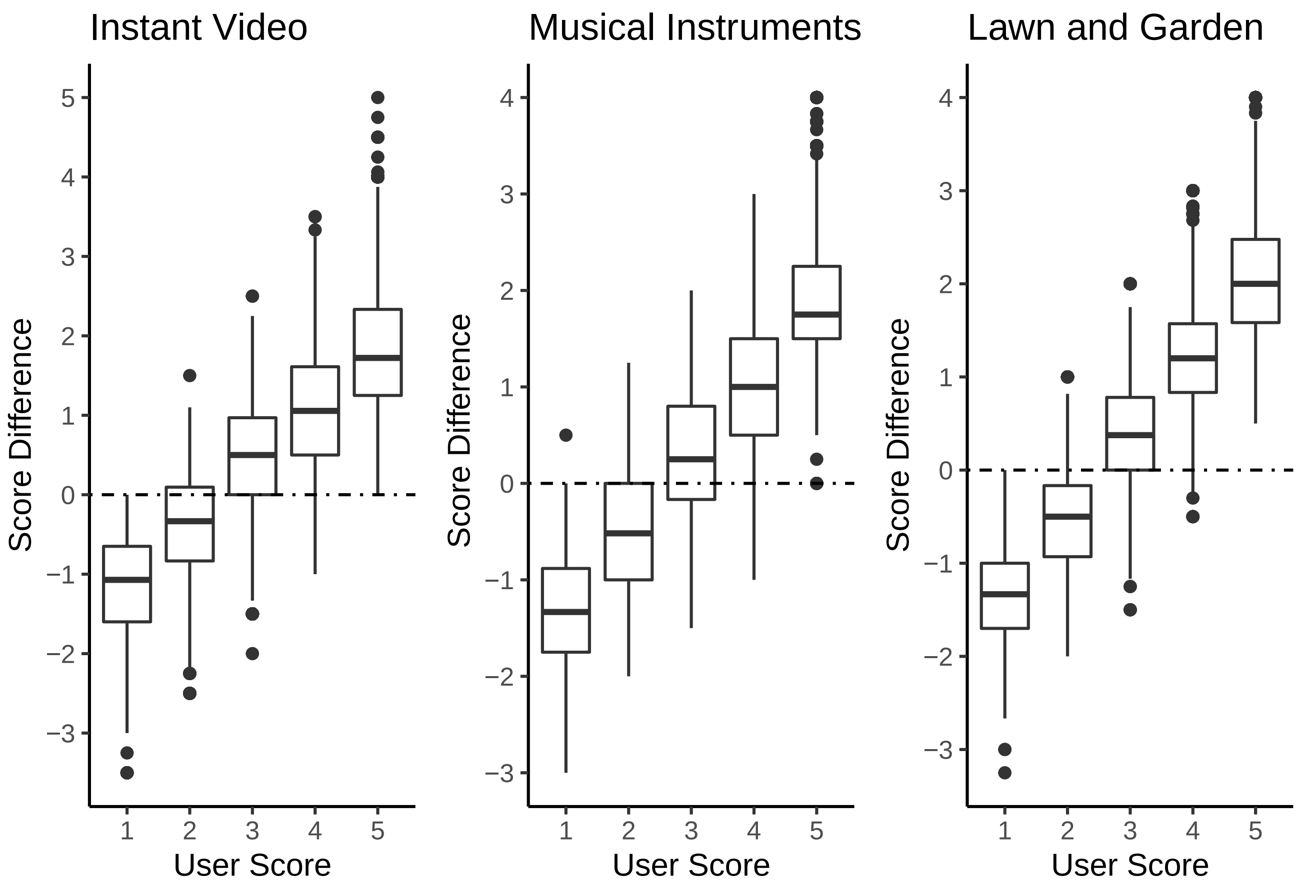

Figure 1: Quartile Distribution of the difference between Sentiment Score and User Score

Figure 1 shows the quartile distribution of the score difference between review Sentiment Score and its User Score. The graph suggests that in the majority of cases Sentiment Score does not reflect the User Score accurately. I believe that this is due to the fact that Sentimenet Analys takes into consideration only individual words and does not account for context of the expressed opinion.

However, something of interest is that quartile distribution varies within each of the three categories. To examine why this might occur, we may investigate words in a review which sentiment varies from the User Score the most. It makes sense to only look at the words that deviate more than by a factor two, since they have the most effect on the spread of the distrubition. For sake of simplicity let's denote these words as *"bad"* words.

```
tib_data_tokens1 %>%
  mutate(token_diff = abs(r_score - value)) %>%
  filter(token_diff > 2) %>%
  count(word, sort = TRUE)  %>%
  with(wordcloud(word, n, random.order = FALSE, min.freq=20))
```

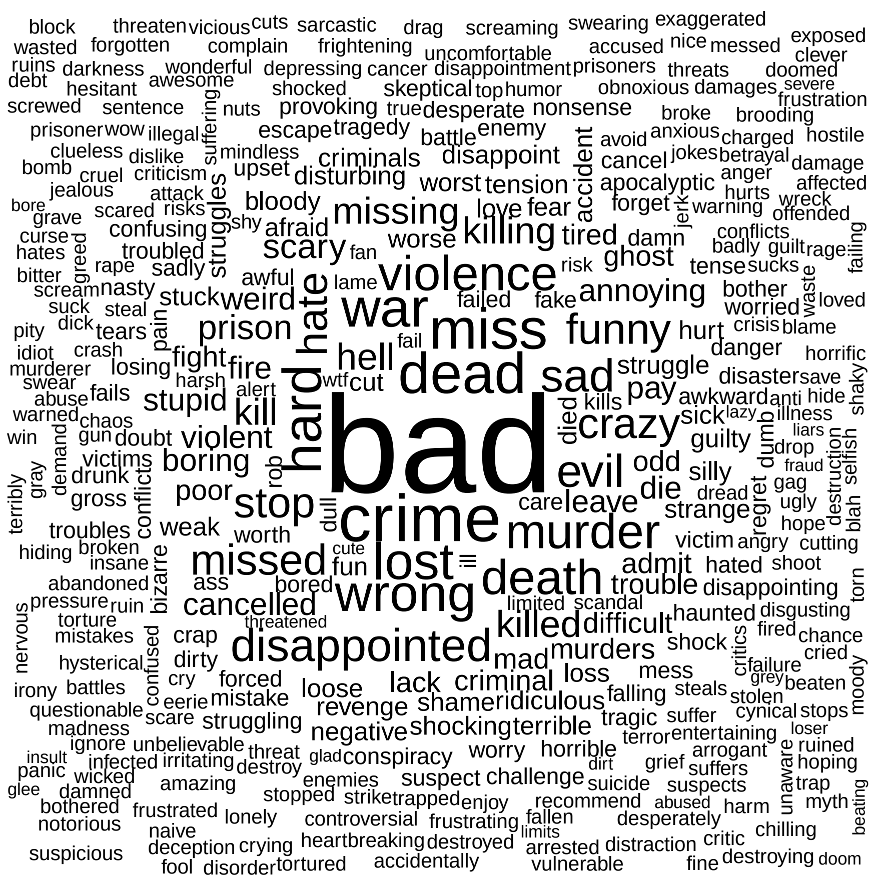

Figure 2: Amazon Instant Video Wordcloud

Figure 2 is a Wordcloud of *"bad"* words in the *Amazon Instant Video* review category. After visual inspection, it seems that inaccuracy in the Sentiment Score compared to the User Score is mostly causes by the words that are describing the events in the video, not by the words which express an opion of it.

```
tib_data_tokens1 %>%
  mutate(token_diff = abs(r_score - value)) %>%
  filter(token_diff > 2) %>%
  count(word, sort = TRUE)   %>%
  left_join(., tib_afinn, by = "word") %>%
  mutate(value = (value + 5) / 2) %>%
  head(n = 10) %>%
  kable(caption = "Top 10 words that imapct Sentiment Score in *Instant Video*",
        col.names = c("Words", "Insatnces", "`affin` score"))
```

Table 1: Top 10 words that imapct Sentiment Score in *Instant Video*

Words	Insatnces	`affin` score
bad	2013	1.0
crime	810	1.0
dead	687	1.0
miss	675	1.5
war	640	1.5
hard	613	2.0
lost	596	1.0
wrong	579	1.5
death	550	1.5
murder	538	1.5

Table 1 confirms that the differnce between the scores is caused by words such as: *war*, *crime*, *death*. These nouns appear to describe events that happen in the video, and do not carry any meaning that might describe users feeling about the video. To confrim such let's try to find an example.

```
display_txt <- . %>% str_c(collapse = "\n") %>% str_wrap(width = 60) %>% cat

r1 <- tib_data1 %>%
  mutate(is_war = str_detect(r_text, " war ")) %>%
  filter(is_war == TRUE & r_score == 5) %>%
  arrange(desc(s_dif)) %>%
  filter(row_number() == 1)

r1 %>%
  .$r_text %>%
  display_txt()
```

```
## so good!! and i am so mad they couldnt make more of them
## due to the death of the writer. im wondering if there's
## a project in the works to bring in new talent for another
## season. just as i was getting into it, poof,,like the war
## itself, good folks dying too early.
```

```r
## User Score
r1 %>%
  .$r_score
```

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

```r
## Sentiment Score
r1 %>%
  .$avg_sentiment
```

```
## [1] 1.333333
```

This output demonstrates a user review which has high User Score and low Sentiment Score. This comment comments expresses high appreciaton for the product. Its low Sentiment score is caused by words: *war*, *death*; they do not show users opinion about the movie, but decribe the events that took place.

```r
tib_data_tokens2 %>%
  mutate(token_diff = abs(r_score - value)) %>%
  filter(token_diff > 2) %>%
  count(word, sort = TRUE)  %>%
  with(wordcloud(word, n, random.order = FALSE, min.freq=10))
```

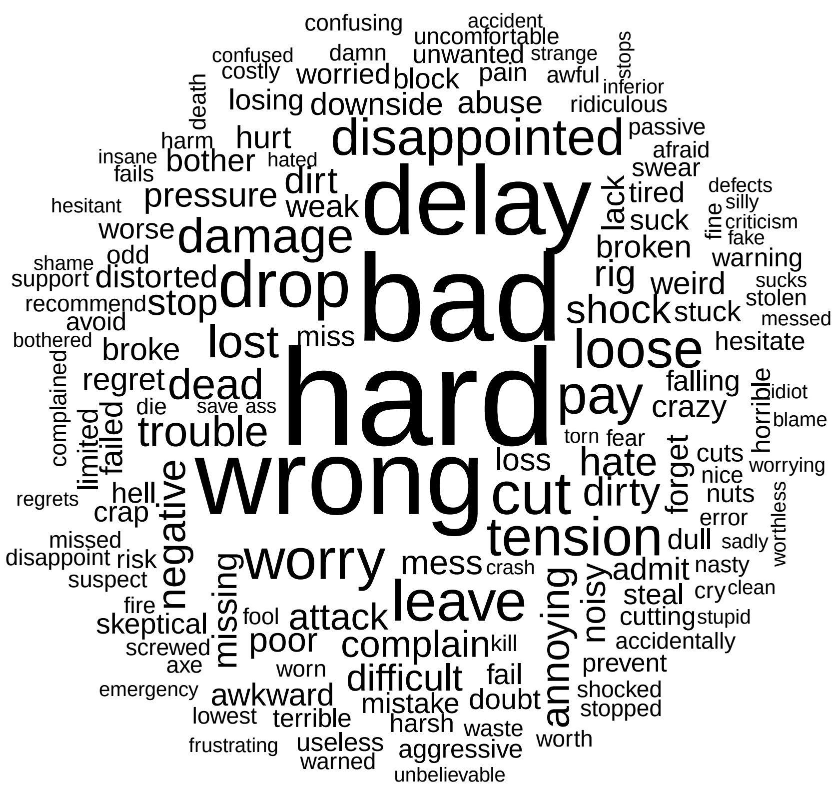

Figure 3: Music Instruments Wordcloud

Figure 3 is a Wordcloud of *"bad"* words in the *Music Instruments* categroy. The wordcloud suggests that the words which might be specific to music instruments and have multiple meanings are one of the causes to deviation of scores.

```
tib_data_tokens2 %>%
  mutate(token_diff = abs(r_score - value)) %>%
  filter(token_diff > 2) %>%
  count(word, sort = TRUE)   %>%
  left_join(., tib_afinn, by = "word") %>%
  mutate(value = (value + 5) / 2) %>%
  head(n = 10) %>%
  kable(caption = "Top 10 words that imapct Sentiment Score in *Music Instruments*",
        col.names = c("Words", "Insatnces", "`affin` score"))
```

Table 2: Top 10 words that imapct Sentiment Score in *Music Instruments*

Words	Insatnces	`affin` score
hard	360	2.0
bad	318	1.0
wrong	269	1.5
delay	239	2.0
drop	143	2.0
cut	125	2.0
worry	124	1.0
leave	121	2.0
pay	115	2.0
tension	112	2.0

Table 2 shows that words with negative sentiment, such as: *hard*, *bad*, *wrong* are the most influential to the Sentiment Socre in the category. However, the table also contains words such as: *delay* and *tension*, which have a meaning realted to music instruments, as well as a meaning with a negative sentiment. To illustatre this, lets find an insatnce of such.

```
r2 <- tib_data2 %>%
  mutate(is_delay = str_detect(r_text, " delay ")) %>%
  filter(is_delay == TRUE & r_score == 5) %>%
  arrange(desc(s_dif)) %>%
  filter(row_number() == 1)

r2 %>%
  .$r_text %>%
  display_txt()
```

```
## The Danelectro FAB series of pedals offer a great sound
## without digging into your budget. The D-8 600Ms Delay
## effects pedal works great and gives me as much delay as I
## need . There's no fancy bells and whistles but just good old
## manual adjustments that lets you dial in the delay you want
## with no effort at all.. This series of pedals ( FAB ) have
## hard plastic cases and less expensive hardware than the more
## costly units but works quit well. Danelectro offers more
```

```
## expensive units but I don't think the sound is much better,
## just the hardware is better, but that's my view!!!. You
## can't go wrong with this product, Danelectro has been around
## for years and makes a great product...I have six Danelectro
## effects pedals and they all work great...
```

```r
## User Score
r2 %>%
  .$r_score
```

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

```r
## Sentiment Score
r2 %>%
  .$avg_sentiment
```

```
## [1] 1.833333
```

This review has a User Score of 5 and Sentiment Score of 1.8. The consumer satified with the product, however the Sentiment Score is still low. This is caused by a word: *delay*. In this context *delay* refers to a *delay pedal* and has no negative sentiment.

```r
tib_data_tokens3 %>%
  mutate(token_diff = abs(r_score - value)) %>%
  filter(token_diff > 2) %>%
  count(word, sort = TRUE)  %>%
  with(wordcloud(word, n, random.order = FALSE, min.freq=10))
```

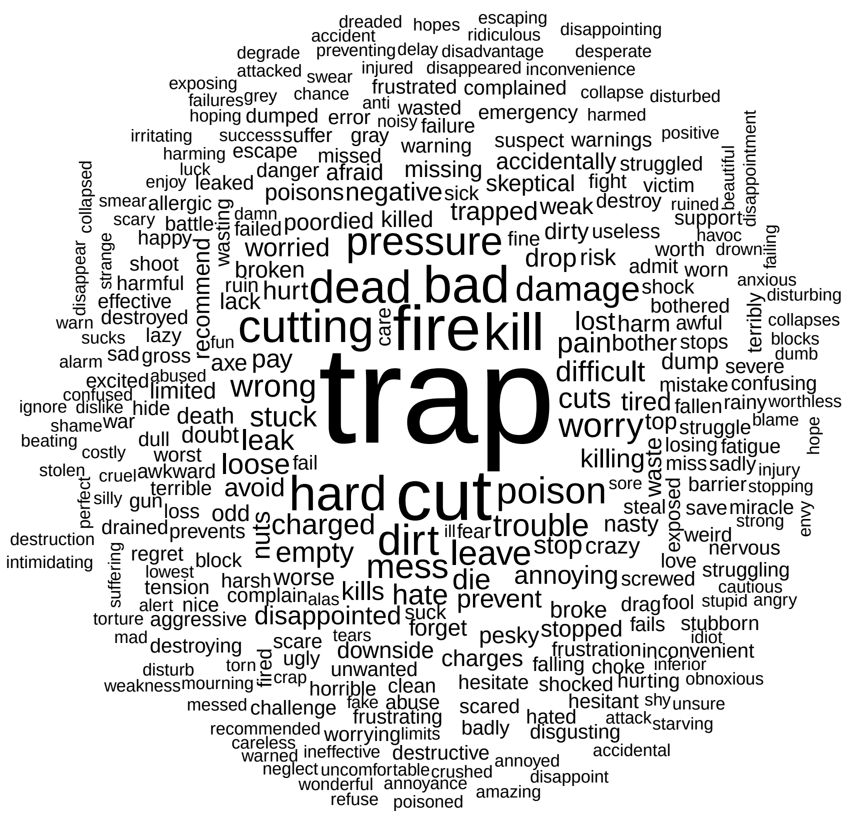

Figure 4: Patio, Lawn and Garden Wordcloud

Figure 4 is a wordcloud of *bad* words in the *Patio, Lawn and Garden* categroy. It shows that the discrepancy in the scores was mostly caused by words related to *Patio, Lawn and Garden* and are homonyms.

```
tib_data_tokens3 %>%
  mutate(token_diff = abs(r_score - value)) %>%
  filter(token_diff > 2) %>%
  count(word, sort = TRUE)   %>%
  left_join(., tib_afinn, by = "word") %>%
  mutate(value = (value + 5) / 2) %>%
  head(n = 10) %>%
  kable(caption = "Top 10 words that imapct Sentiment Score in *Patio, Lawn and Garden*",
        col.names = c("Words", "Insatnces", "`affin` score"))
```

Table 3: Top 10 words that imapct Sentiment Score in *Patio, Lawn and Garden*

Words	Insatnces	`affin` score
trap	1746	2.0
cut	771	2.0
fire	649	1.5
kill	497	1.0
bad	486	1.0
hard	477	2.0
dead	432	1.0
cutting	401	2.0
dirt	369	1.5
pressure	320	2.0

Table 3 indeed shows that the inaccuracy of the Sentiment Score compare to User Score is caused by the words which are specific to the products in the category. To show this explicity, let's find an example.

```
r3 <- tib_data3 %>%
  mutate(is_trap = str_detect(r_text, " trap ")) %>%
  filter(is_trap == TRUE & r_score == 5) %>%
  arrange(desc(s_dif)) %>%
  filter(row_number() == 1)

r3 %>%
  .$r_text %>%
  display_txt()
```

```
## I hate mice, so when I found some droppings in the corner of
## my kitchen, I was really upset. It was time to do something
## and to get rid of these guys once and for all. I found the
## Victor Tri-Kill mouse trap and ordered it, it was the best
## thing that I ever did. This will kill 3 mice at once and
## that is something that you definitely want to check out.
```

```
## User Score
r3 %>%
  .$r_score
```

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

```
## Sentiment Score
r3 %>%
  .$avg_sentiment
```

```
## [1] 1.3
```

This review has a maximum User Score and a very low Sentiment Score. In it user describes the cause for buying the product and his good experince with it. Such Sentiment Score is caused by words: *kill*, *trap*. This example confirms that the discrepancy between the scores was caused by words specific to this shopping category.

Distribution of User Scores

```
data_plot1 %>%
  ggplot(aes(x = f, y = prc)) +
  geom_col(fill = "darksalmon") +
  scale_y_continuous(labels=scales::percent) +
  facet_wrap(year ~ .) +
  theme_bw() +
  labs( x = "User Score",
        y = "Percentage Presence")
```

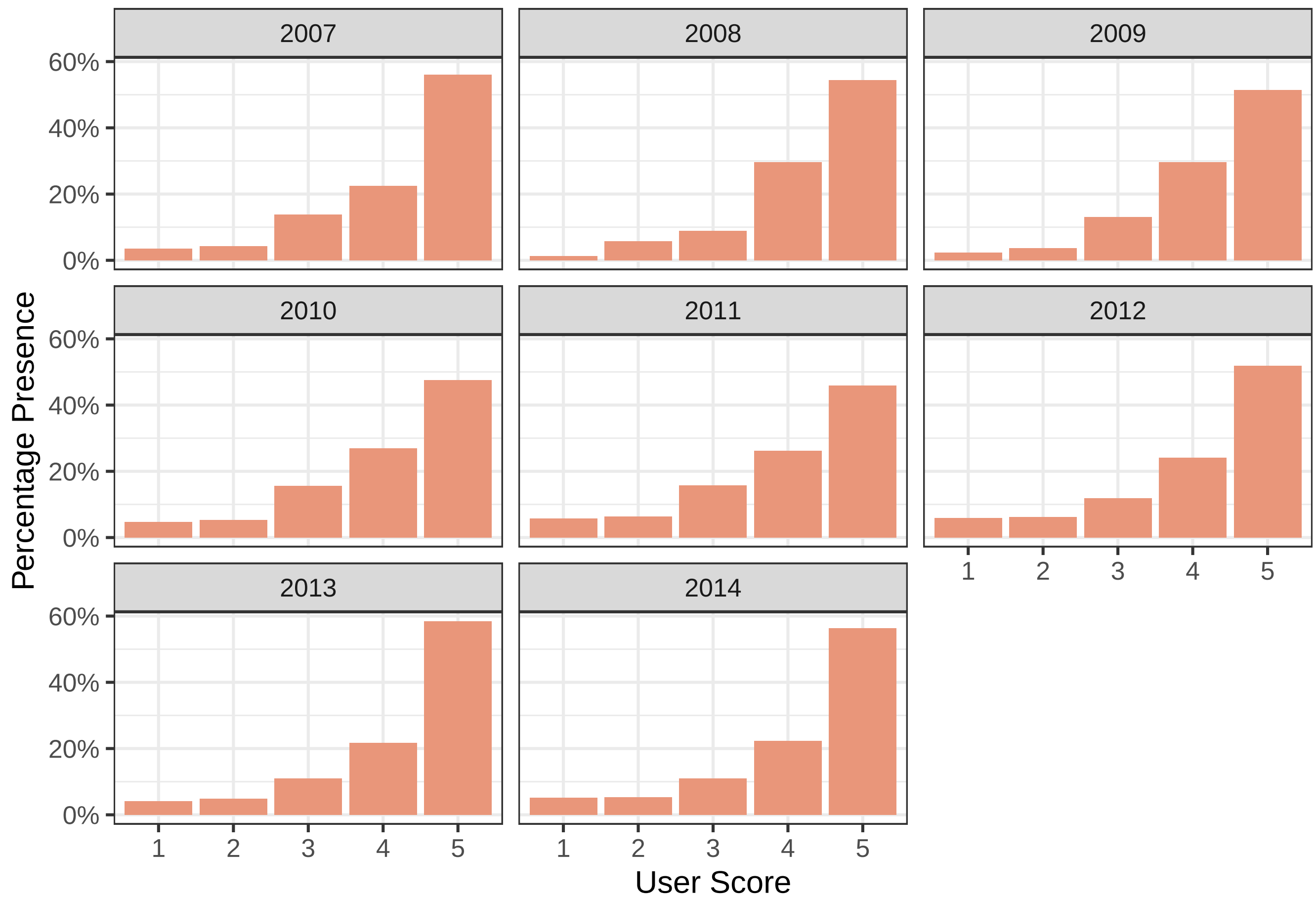

Figure 5: Amazon Instant Video User Score distribution

Figure 5 shows the distribution of User Scores and how it changes over time within *Amazon Instant Video* category. It appears that no change in trend of distribtuion has occured, the higher the user score of a review the more presense there is of reviews with such score.

```
data_plot2 %>%
  ggplot(aes(x = f, y = prc)) +
  geom_col(fill = "lightgreen") +
  scale_y_continuous(labels=scales::percent) +
  facet_wrap(year ~ .) +
```

```
  theme_bw() +
  labs( x = "User Score",
        y = "Percentage Presence")
```

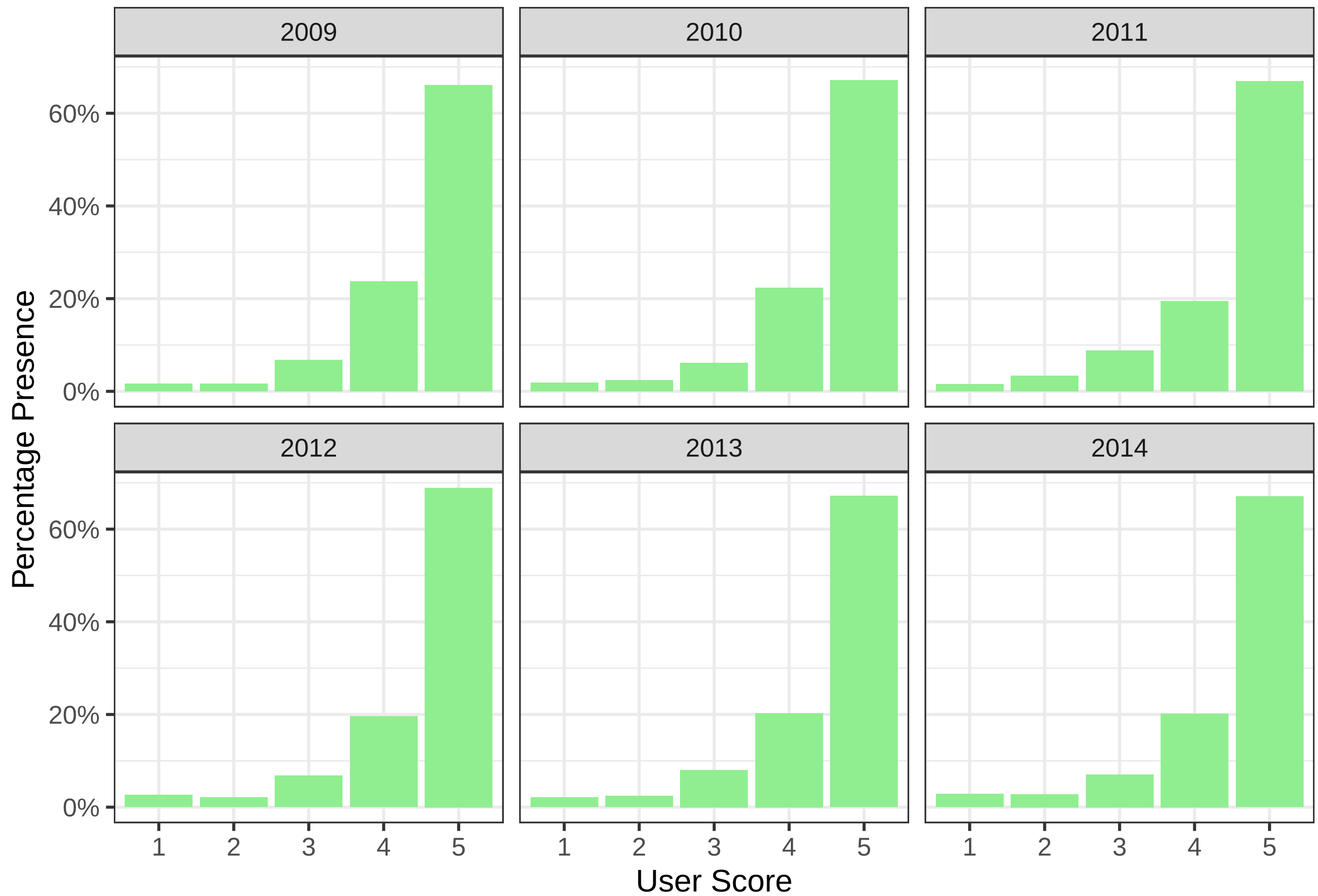

Figure 6: Muisc Instruments User Score distribution

Figure 6 shows the distribution of User Scores in *Music Instruments* category. The graph shows the same trend as seen in *Amazon Instant Video* category.

```
data_plot3 %>%
  ggplot(aes(x = f, y = prc)) +
  geom_col(fill = "lightblue") +
  scale_y_continuous(labels=scales::percent) +
  facet_wrap(year ~ .) +
  theme_bw() +
  labs( x = "User Score",
        y = "Percentage Presence")
```

Figure 7 shows the distribution of User Scores in *Music Instruments* category. The graph shows the same trend as seen in previous categories, however during the years 2006-2010 the User Score of 1 had more presense than User Score of 2. This disrepentecy might be caused by small sample set of the years.

Note that range of each graph differs, this is done to only display meaningful results.

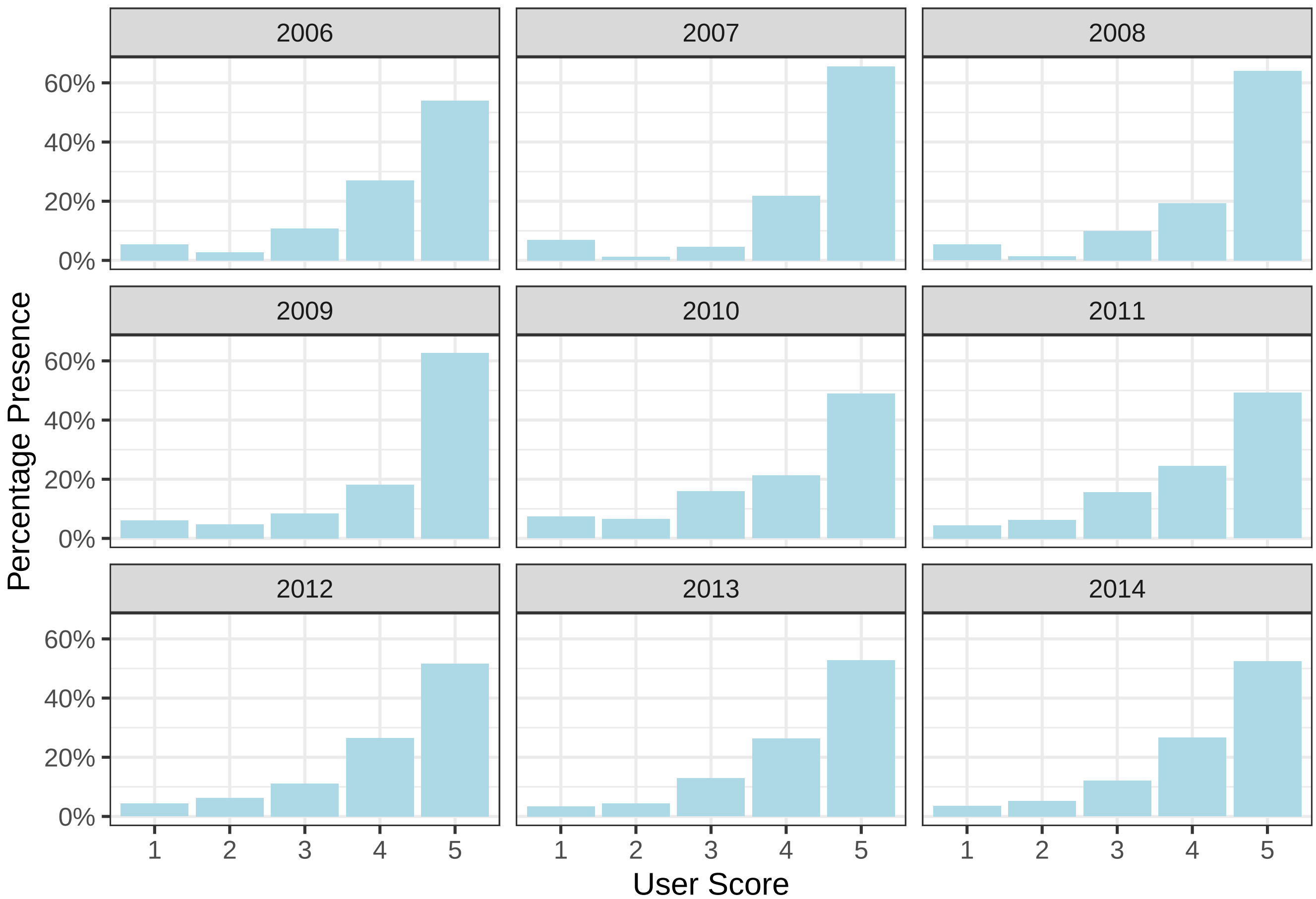

Figure 7: Patio, Lawn and Garden User Score distribution

```r
data_tmp1 <- tib_data1 %>%
  .$r_score_fct %>%
  fct_count() %>%
  add_column(category = "Amazon Instant Video") %>%
  mutate(prc = n/sum(n))

data_tmp2 <- tib_data2 %>%
  .$r_score_fct %>%
  fct_count() %>%
  add_column(category = "Music Instruments") %>%
  mutate(prc = n/sum(n))

data_tmp3 <- tib_data2 %>%
  .$r_score_fct %>%
  fct_count() %>%
  add_column(category = "Patio, Lawn and Graden") %>%
  mutate(prc = n/sum(n))

data_all <- bind_rows(data_tmp1, data_tmp2, data_tmp3)
data_all <- data_all %>% mutate(category = as_factor(category))
```

```r
data_all %>%
  ggplot(aes(x = f, y = prc, fill = category )) +
  geom_col(position = "dodge") +
  scale_y_continuous(breaks = seq(0, 0.7, by = 0.1), labels=scales::percent) +
  scale_fill_manual(values=c("darksalmon", "lightgreen", "lightblue")) +
  theme_bw() +
  labs( x = "User Score",
        y = "Percentage Presence",
        fill = "Category")
```

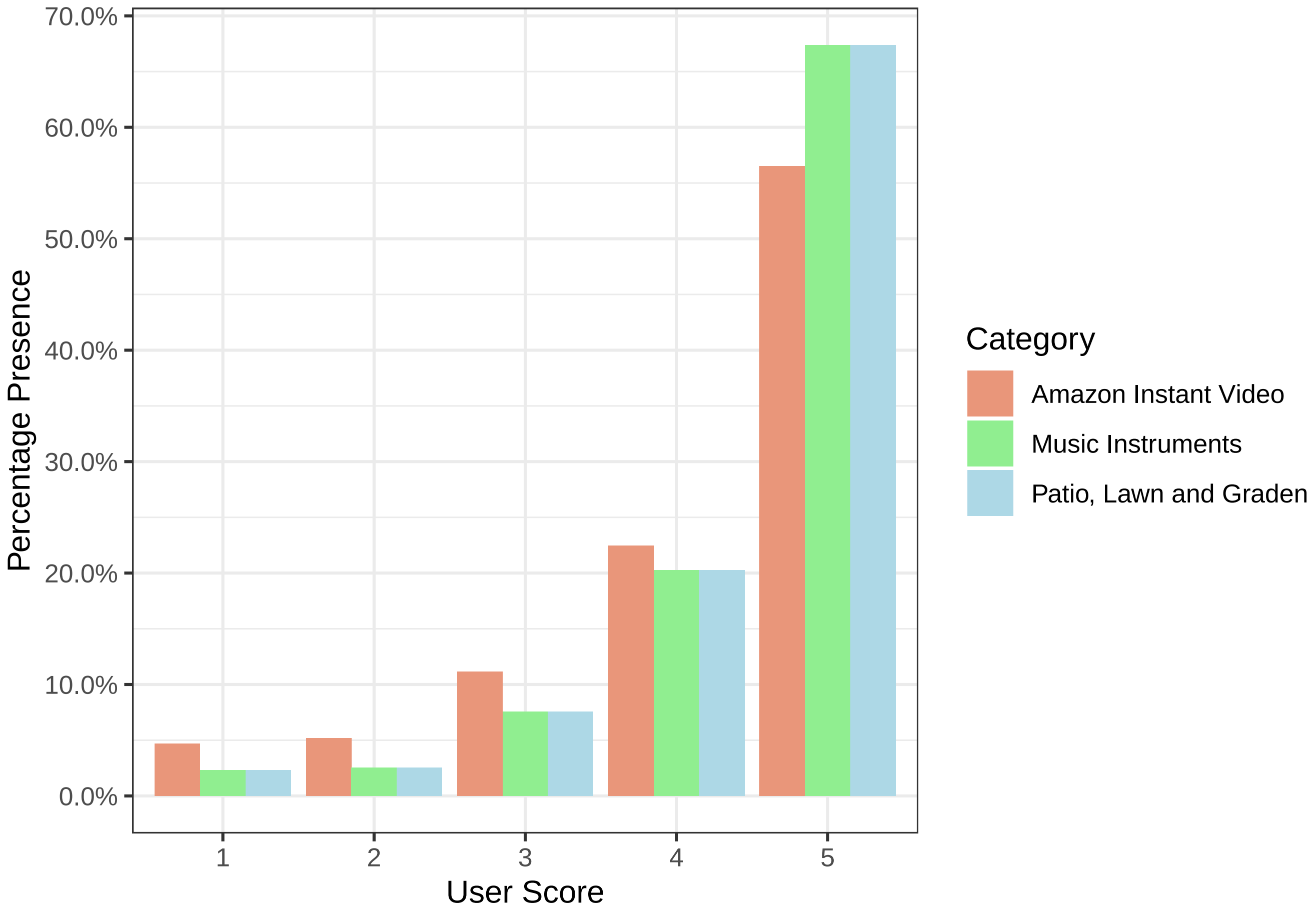

Figure 8: Overall User Score distrubiton

Figure 8 shows the distrubtion of User Scores between different categories, this graph covers all reviews in datasets. It shows that reviews in *Music Instrumetns* category and in *Patio, Lawn and Garden* follow the same distrubtion, but *Amazon Instant Video* reviews slighty deviates from it. This might be due to the fact that reviews in *Music Instrument* and *Patio, Lawn and Garden* focus on physical goods, but reviews in *Amazon Instant Video* focus on media.

Conclusion

In this project I invisitgated the relationship between Sentiment Score and User Score and distribution of User Scores over time. This work showed that Sentiment Score is inaccurate representaion of User Score. I believe this occured is due to the fact that Sentiment Analysis used in this projects only considers individual words and does not take context into account. The variation in the distribution of error between the categories is caused by specifics of each category, which were disscussed. There appears to be no change in the distrubtion of user socres over time, in no categories. However, one ineresting find is that distibution of the User Scores varies for physical goods and media.

References

He, R., & McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. *Proceedings of the 25th International Conference on World Wide Web*, 507–517.

Irizarry, R. A. (2019). *Introduction to data science: Data analysis and prediction algorithms with r*. CRC Press.