Assignment 3: Kickstarter Projects

Xuejing Li

2021-03-31

Text Mining Kickstarter Projects

Overview

Kickstarter is an American public-benefit corporation based in Brooklyn, New York, that maintains a global crowd funding platform focused on creativity. The company's stated mission is to "help bring creative projects to life".

Kickstarter has reportedly received almost $6 billion in pledges from 19.4 million backers to fund 200,000 creative projects, such as films, music, stage shows, comics, journalism, video games, technology and food-related projects.

For this assignment, I am asking you to analyze the descriptions of kickstarter projects to identify commonalities of successful (and unsuccessful projects) using the text mining techniques we covered in the past two lectures.

Data

The dataset for this assignment is taken from webroboto.io 's repository. They developed a scrapper robot that crawls all Kickstarter projects monthly since 2009. We will just take data from the most recent crawl on 2021-03-18.

To simplify your task, I have downloaded the files and partially cleaned the scraped data. In particular, I converted several JSON columns, corrected some obvious data issues, and removed some variables that are not of interest (or missing frequently), and removed some duplicated project entries. I have also subsetted the data to only contain projects originating in the United States (to have only English language and USD denominated projects). Some data issues surely remain, so please adjust as you find it necessary to complete the analysis.

The data is contained in the file `kickstarter_projects_2021_03.csv` and contains about 125k projects and about 20 variables.

Tasks for the Assignment

```
kickstarter <- read.csv("kickstarter_projects_2021-03.csv")
```

1. Identifying Successful Projects

a) Success by Category

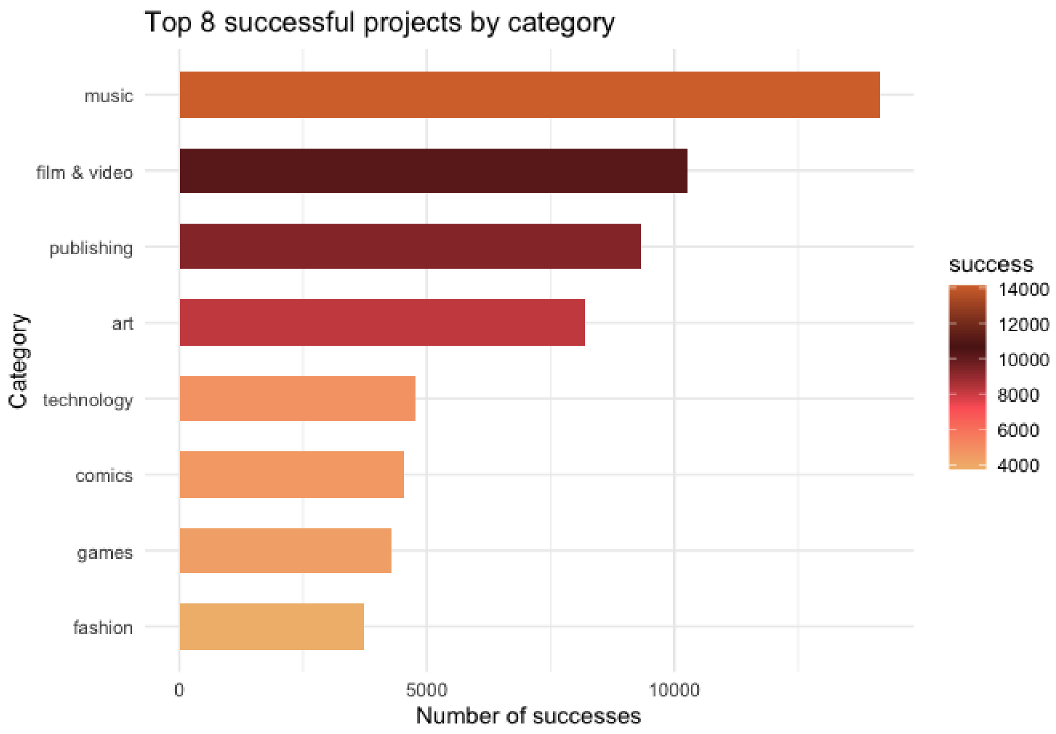
There are several ways to identify success of a project:
- State (`state`): Whether a campaign was successful or not.
- Pledged Amount (`pledged`)
- Achievement Ratio: The variable `achievement_ratio` is calculating the percentage of the original monetary `goal` reached by the actual amount `pledged` (that is `pledged` \ `goal` *100).
- Number of backers (`backers_count`)
- How quickly the goal was reached (difference between `launched_at` and `state_changed_at`) for those campaigns that were successful.

Use one or more of these measures to visually summarize which categories were most successful in attracting funding on kickstarter. Briefly summarize your findings.

```
success_state <- kickstarter %>%
  group_by(top_category) %>%
  filter(state == "successful") %>%
  summarize(success = n()) %>%
  arrange(desc(success)) %>%
  top_n(8) %>%
  ggplot(.,aes(x=reorder(top_category, success), success,fill=success)) +
  geom_bar(stat="identity",width=0.6,alpha=1) +
  coord_flip() +
  scale_fill_gradientn(colors=wes_palette(name="GrandBudapest1")) +
  theme_minimal()+
  labs(title="Top 8 successful projects by category",x="Category", y= "Number of successes",caption = "Sourc
e: https://webrobots.io/kickstarter-datasets/")
```

```
## Selecting by success
```

```
success_state
```



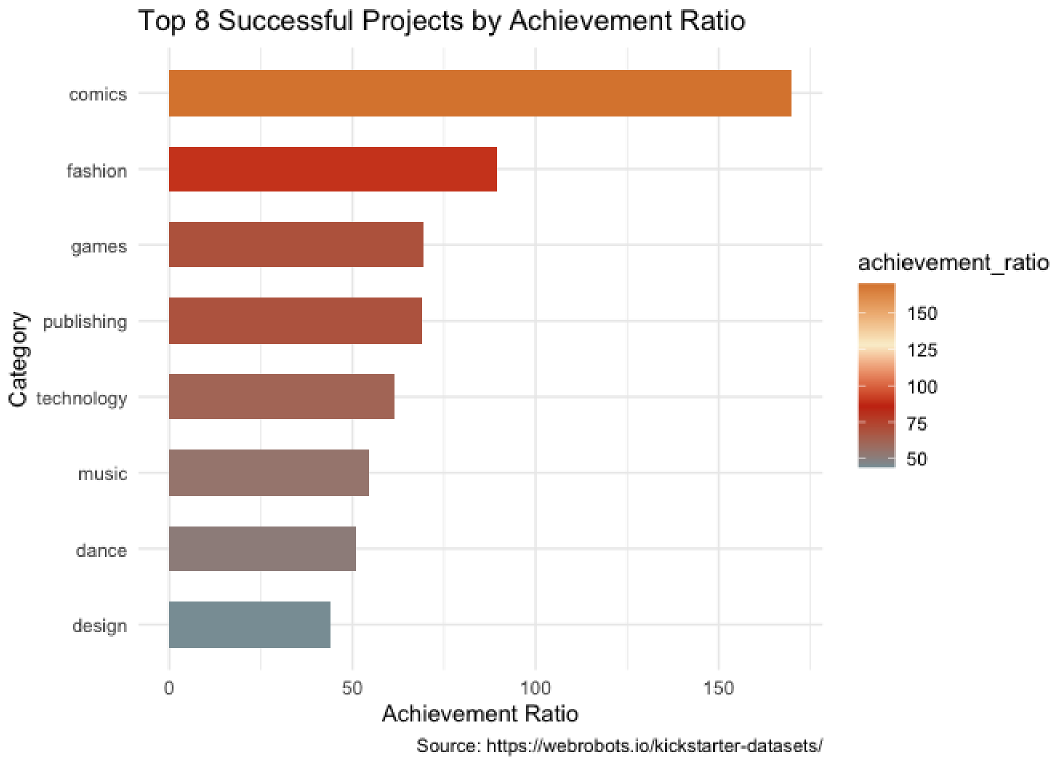
Source: https://webrobots.io/kickstarter-datasets/

- Achievement Ratio: The variable `achievement_ratio` is calculating the percentage of the original monetary `goal` reached by the actual amount `pledged` (that is `pledged` \ `goal` *100).

```
AchievementRatio <- kickstarter %>%
  group_by(top_category) %>%
  summarise_at(c("pledged", "goal"), sum, na.rm = TRUE) %>%
  mutate(achievement_ratio = round(pledged/goal * 100, 2)) %>%
  arrange(desc(achievement_ratio)) %>%
  top_n(8) %>%
  ggplot(.,aes(x=reorder(top_category, achievement_ratio), achievement_ratio,fill=achievement_ratio)) +
  geom_bar(stat="identity",width=0.6,alpha=1) +
  coord_flip() +
  scale_fill_gradientn(colors=wes_palette(name="Royal1")) +
  theme_minimal()+
  labs(title="Top 8 Successful Projects by Achievement Ratio",x="Category", y= "Achievement Ratio",caption =
"Source: https://webrobots.io/kickstarter-datasets/")
```

```
## Selecting by achievement_ratio
```

```
AchievementRatio
```



I looked at number of successes and calculated the achievement ratio of each projects, and then I grouped all projects by categories. The bar graphs above showcase how the ranking of “most successful” projects different by category if we use different measures of success. Interestingly, comics, fashion, and games have the most counts of successful projects, but have the lowest achievement ratio. My hypothesis is that the total amount raised for these three categories are high, but there are fewer total projects in these categories.

BONUS ONLY: b) Success by Location

Now, use the location information to calculate the total number of successful projects by state (if you are ambitious, normalize by population). Also, identify the Top 50 “innovative” cities in the U.S. (by whatever measure you find plausible). Provide a leaflet map showing the most innovative states and cities in the U.S. on a single map based on these information.

2. Writing your success story

Each project contains a `blurb` – a short description of the project. While not the full description of the project, the short headline is arguably important for inducing interest in the project (and ultimately popularity and success). Let’s analyze the text.

a) Cleaning the Text and Word Cloud

To reduce the time for analysis, select the 1000 most successful projects and a sample of 1000 unsuccessful projects. Use the cleaning functions introduced in lecture (or write your own in addition) to remove unnecessary words (stop words), syntax, punctuation, numbers, white space etc. Note, that many projects use their own unique brand names in upper cases, so try to remove these fully capitalized words as well (since we are aiming to identify common words across descriptions). Create a document-term-matrix.

Provide a word cloud of the most frequent or important words (your choice which frequency measure you choose) among the most successful projects.

```
#get 1000 most successful projects and 1000 least successful projects
Suc1000 <- kickstarter %>%
  select(blurb,pledged, goal, top_category) %>%
  mutate(achievement_ratio = round(log(pledged/goal * 100), 2)) %>%  #log transformation on achievement rati
o, and then round it two two digits
  arrange(desc(achievement_ratio)) %>%
  top_n(1000)
```

```
## Selecting by achievement_ratio
```

```
Unsuc1000 <- kickstarter %>%
  select(blurb,pledged, goal, top_category) %>%
  mutate(achievement_ratio = round(pledged/goal * 100, 2)) %>%
  arrange(achievement_ratio, desc(goal)) %>%
  slice_head(n = 1000)
```

```
#Get 1000 most successful projects, clean the corpus, and create the document-term-matrix

# Method 1: Use tidytext to clean the text
Success <- Suc1000 %>%
  select(blurb)  %>%
  unnest_tokens(word, blurb) %>%
  anti_join(get_stopwords(source = "smart")) %>%
  count(word, sort=TRUE)
```

```
## Joining, by = "word"
```

```
# Method 2: Use tm package convert dataframe to corpus
doc_id = c(1:1007)
example_text <- data.frame(doc_id, text = Suc1000$blurb, stringsAsFactors = FALSE)

# Convert example_text to a corpus: Success_corpus
Success_corpus <- VCorpus(DataframeSource(example_text))

corpus <- tm_map(Success_corpus, content_transformer(tolower))
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removeWords, c(stopwords("en")))
corpus <- tm_map(corpus, content_transformer(function(x) gsub("[[:upper:]]", "", x))) #Remove brand name in
upper cases
corpus <- tm_map(corpus, stripWhitespace)
corpus <- tm_map(corpus, removePunctuation)

success_dtm <- DocumentTermMatrix(corpus)
#Document Term Matrix of Successful Projects
success_dtm
```

```
## <<DocumentTermMatrix (documents: 1007, terms: 4376)>>
## Non-/sparse entries: 10144/4396488
## Sparsity           : 100%
## Maximal term length: 36
## Weighting          : term frequency (tf)
```

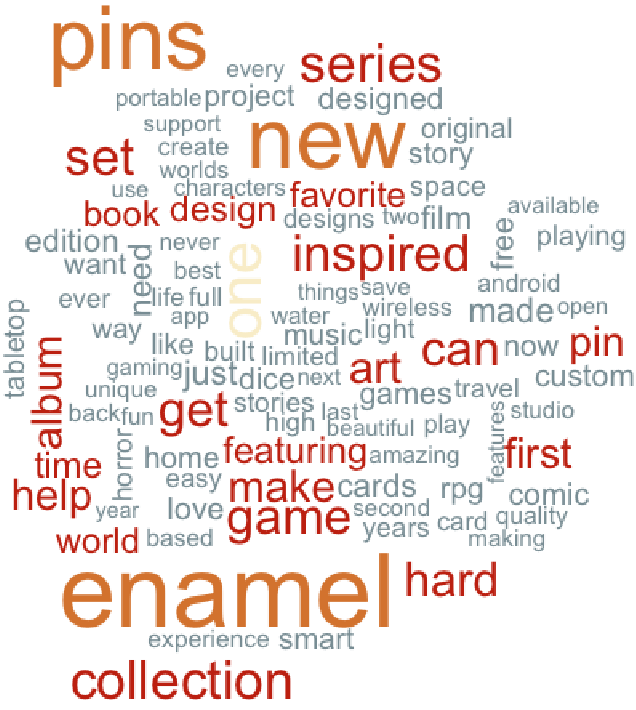
```
##Get 1000 most successful projects, clean the corpus, and create the document-term-matrix

# Method 1: Use tidytext to clean the text
Unsuccess <- Unsuc1000 %>%
  select(blurb)  %>%
  unnest_tokens(word, blurb) %>%
  anti_join(get_stopwords(source = "smart")) %>%
  count(word, sort=TRUE)
```

```
## Joining, by = "word"
```

```
# Method 2: Use tm package to convert dataframe to corpus
doc_id = c(1:1000)
example_text_unsuc <- data.frame(doc_id, text = Unsuc1000$blurb, stringsAsFactors = FALSE)

# Convert example_text_unsuc to a corpus: Success_corpus
Unsuccess_corpus <- VCorpus(DataframeSource(example_text_unsuc))

#Clean the corpus text
un_corpus <- tm_map(Unsuccess_corpus, content_transformer(tolower))
un_corpus <- tm_map(un_corpus, removeNumbers)
un_corpus <- tm_map(un_corpus, removeWords, c(stopwords("en")))
un_corpus <- tm_map(un_corpus, content_transformer(function(x) gsub("[[:upper:]]", "", x))) #Remove brand na
me in upper cases
un_corpus <- tm_map(un_corpus, stripWhitespace)
un_corpus <- tm_map(un_corpus, removePunctuation)

Unsuccess_dtm <- DocumentTermMatrix(un_corpus)
#Document Term Matrix of Successful Projects
Unsuccess_dtm
```

```
## <<DocumentTermMatrix (documents: 1000, terms: 4496)>>
## Non-/sparse entries: 10623/4485377
## Sparsity           : 100%
## Maximal term length: 71
## Weighting          : term frequency (tf)
```

```
#convert dtm matrix into    dataframe
success_td <- tidy(success_dtm) %>%
  group_by(term) %>%
  summarize(n = sum(count)) %>%
  arrange(desc(n))

# Bind the TF,DF, and IDF frequency  of a tidy text dataset to the dataset
# success_tf_idf <-  success_td %>%
#                 bind_tf_idf(term, document, count) %>%
#                 arrange(desc(tf_idf))
# success_tf_idf

# Create a wordcloud with wesanderson palette
wordcloud(success_td$term, success_td$n,
      max.words = 100, colors = wes_palette(name="Royal1"))
```



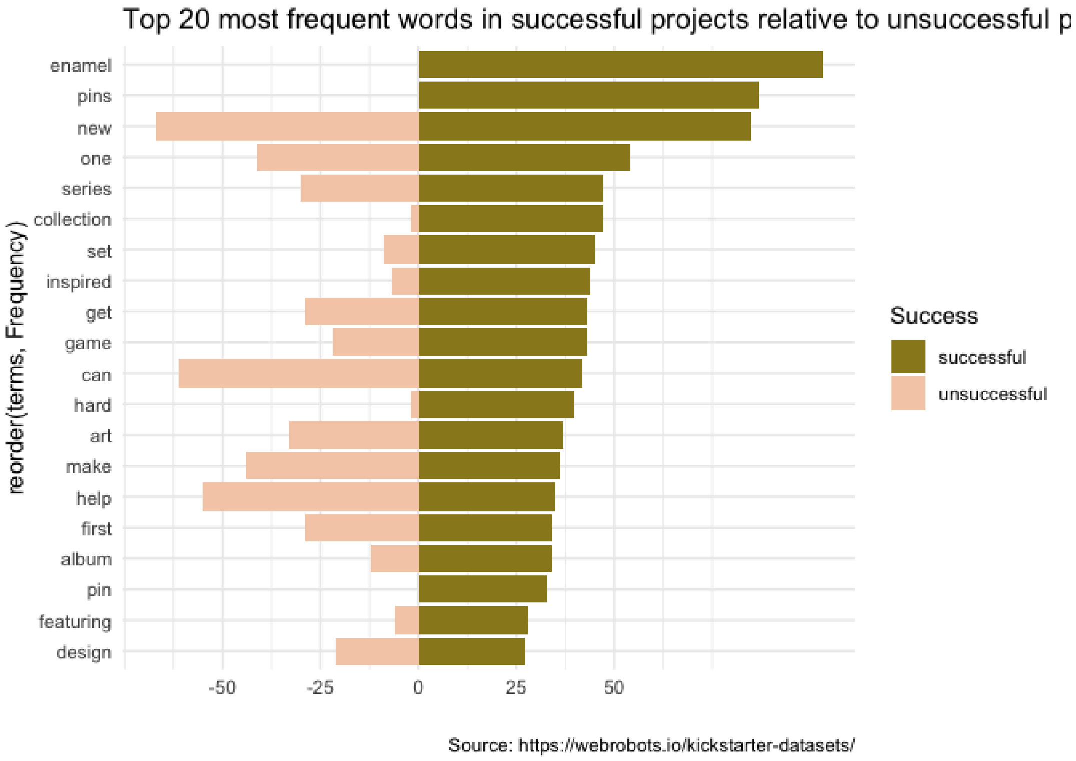
b) Success in words

Provide a pyramid plot to show how the words between successful and unsuccessful projects differ in frequency. A selection of 10 - 20 top words is sufficient here.

```
#join dataframes
Unsuccess_td <- tidy(Unsuccess_dtm) %>%
  group_by(term) %>%
  summarize(n = sum(count)) %>%
  arrange(desc(n))

joined <- success_td %>%
  left_join(Unsuccess_td, by = c("term"), suffix = c("_successful", "_unsuccessful")) %>%
  arrange(desc(n_successful)) %>%
  slice_head(n = 20)
```

Pyramid Plot

```
ggplot(joined, aes(x = reorder(terms, Frequency),
                  y = Frequency, fill = Success)) +
  geom_bar(data = filter(joined, Success == "successful"), stat = "identity") +
  geom_bar(data = filter(joined, Success == "unsuccessful"), stat = "identity") +
  scale_fill_manual(values=wes_palette(name="Royal2")) + coord_flip() +
  scale_y_continuous(breaks = seq(-50, 50, 25)) + ylab("") +
  theme_minimal()+
  labs(title="Top 20 most frequent words in successful projects relative to unsuccessful projects",caption =
"Source: https://webrobots.io/kickstarter-datasets/")
```



#### c) Simplicity as a virtue

These blurbs are short in length (max. 150 characters) but let's see whether brevity and simplicity still matters. Calculate a readability measure (Flesh Reading Ease, Flesh Kincaid or any other comparable measure) for the texts. Visualize the relationship between the readability measure and one of the measures of success. Briefly comment on your finding.

```
require(quanteda)
require(dplyr)
suc_corpus <- corpus(corpus)  # convert to quanteda corpus
FRE_success <- textstat_readability(suc_corpus,
               measure=c('Flesch.Kincaid'))
```

I created an interactive plot that showcases the relationship between readability and success categorized by project category. The relationship makes sense especially if we look at how comics has a higher readability level compared to all other categories. The readability level of comic projects is skewed upwards and that of music projects has the widest range.

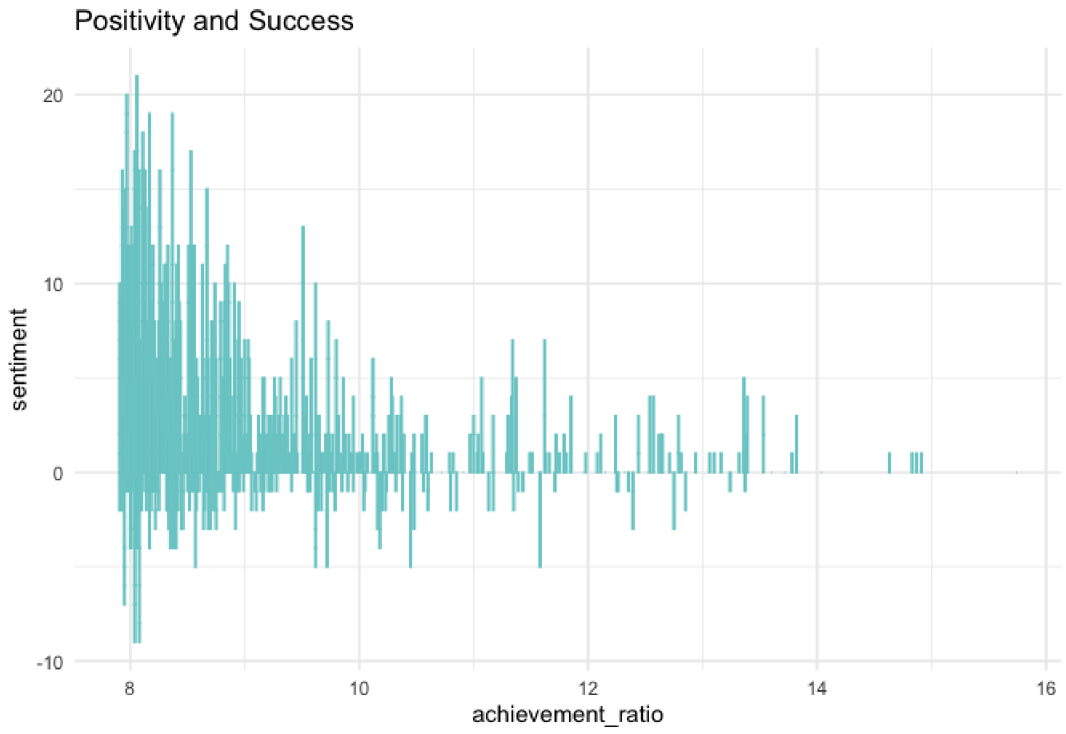
3. Sentiment

Now, let's check whether the use of positive / negative words or specific emotions helps a project to be successful.

a) Stay positive

Calculate the tone of each text based on the positive and negative words that are being used. You can rely on the Hu & Liu dictionary provided in lecture or use the Bing dictionary contained in the tidytext package (`tidytext::sentiments`). Visualize the relationship between tone of the document and success. Briefly comment.

```
#get dataframe of 1000 most successful projects and 1000 least successful projects
text2000 <- bind_rows(Suc1000,Unsuc1000) %>%
  mutate(id=row_number())

#conbine DTM of success and unsuccessful projects
blurb_dtm <- c(success_dtm, Unsuccess_dtm)
#Convert DTM to tidy dataframe
blurb_tidy <- tidy(blurb_dtm) %>%
  mutate(index = as.numeric(document)) %>%
  left_join(text2000, by=c("index"="id"))

# Get Bing lexicon
bing <- get_sentiments("bing")

# Join text to lexicon
blurb_bing <- inner_join(blurb_tidy, bing, by = c("term" = "word")) %>%
#   mutate(index = as.numeric(document))
   # Count by sentiment, index, document
  count(sentiment,index,document, blurb,achievement_ratio,top_category) %>%
   # Spread sentiments
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive-negative) %>%
  ggplot(.,aes(x=achievement_ratio, y=sentiment)) +
  geom_col(color="darkslategray3")+
  theme_minimal()+
  labs(title="Positivity and Success",caption = "Source: https://webrobots.io/kickstarter-datasets/")

# Review the spread data
blurb_bing
```



Source: https://webrobots.io/kickstarter-datasets/

From this graph, we can tell the relationship between the sentiment of the project description and it's achievement ratio. While projects with relatively lower achievement ratio tend to have more variations in description sentiment and a majority of them have a positive sentiment, the higher the achievement ratio, the more likely the descriptions' sentiment is neutral.

b) Positive vs negative

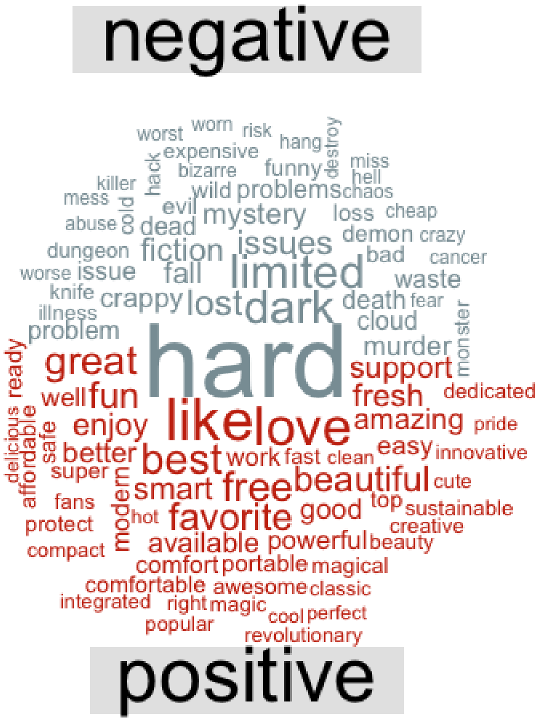
Segregate all 2,000 blurbs into positive and negative texts based on their polarity score calculated in step (a). Now, collapse the positive and negative texts into two larger documents. Create a document-term-matrix based on this collapsed set of two documents. Generate a comparison cloud showing the most-frequent positive and negative words.

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
blurb_compare <- inner_join(blurb_tidy, bing, by = c("term" = "word")) %>%
  count(term, sentiment) %>%
  acast(term ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = wes_palette(name="Royal1"),
                   max.words = 100)
```



```
blurb_compare
```

```
## NULL
```

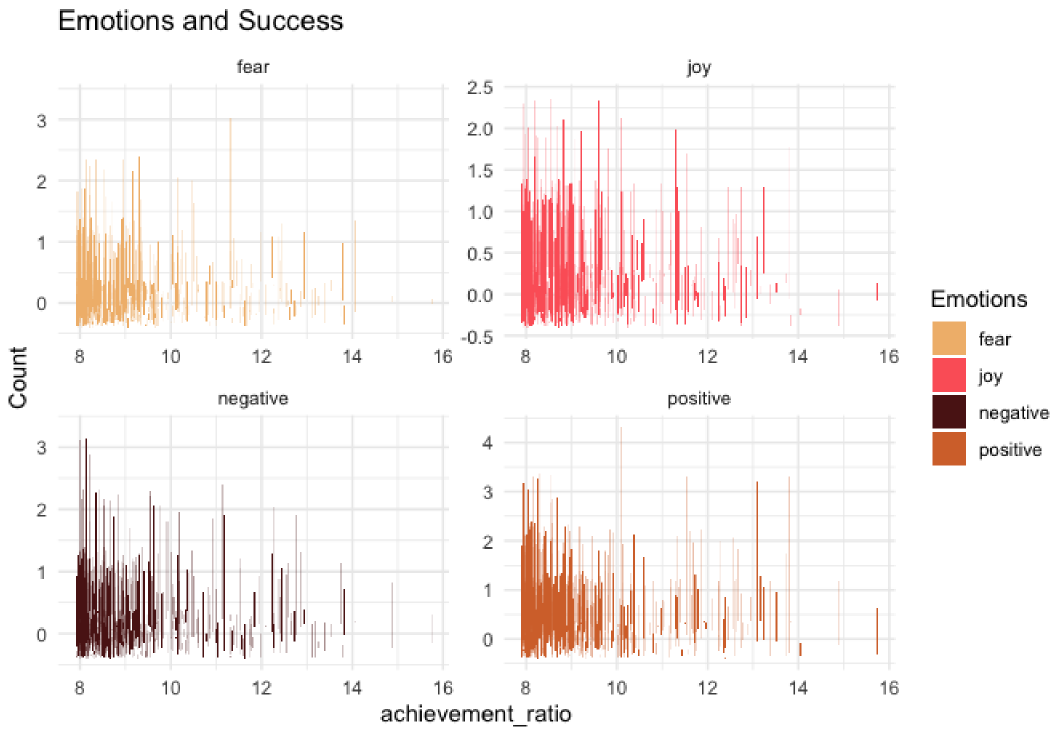
c) Get in their mind

Now, use the NRC Word-Emotion Association Lexicon in the `tidytext` package to identify a larger set of emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust). Again, visualize the relationship between the use of words from these categories and success. What is your finding?

```
#install.packages("textdata", dependencies = TRUE)
# Get NRC Word-Emotion lexicon
nrc1 <- get_sentiments("nrc") %>%
  filter(sentiment == c("joy","positive","negative","fear"))
```

```
## Warning in sentiment == c("joy", "positive", "negative", "fear"): longer object
## length is not a multiple of shorter object length
```

```
blurb_NRC_4 <- inner_join(blurb_tidy, nrc1, by = c("term" = "word")) %>%
  count(sentiment,index,document, blurb,achievement_ratio) %>%
  spread(sentiment, n, fill = 0) %>%
  melt(id.vars=c("index","document","blurb","achievement_ratio"),value.name = "Count", variable.name="Emotio
ns") %>%
  ggplot(.,aes(x=achievement_ratio, y=Count, fill=Emotions)) +
  geom_col( position="jitter")+
  scale_fill_manual(values=wes_palette(name="GrandBudapest1"))+
  #scale_fill_manual(values = c("cadetblue","coral","coral4","darkslategray2"))+
  facet_wrap(~Emotions, ncol = 2, scales = "free")+
  theme_minimal()+
  labs(title="Emotions and Success",caption = "Source: https://webrobots.io/kickstarter-datasets/")

blurb_NRC_4
```

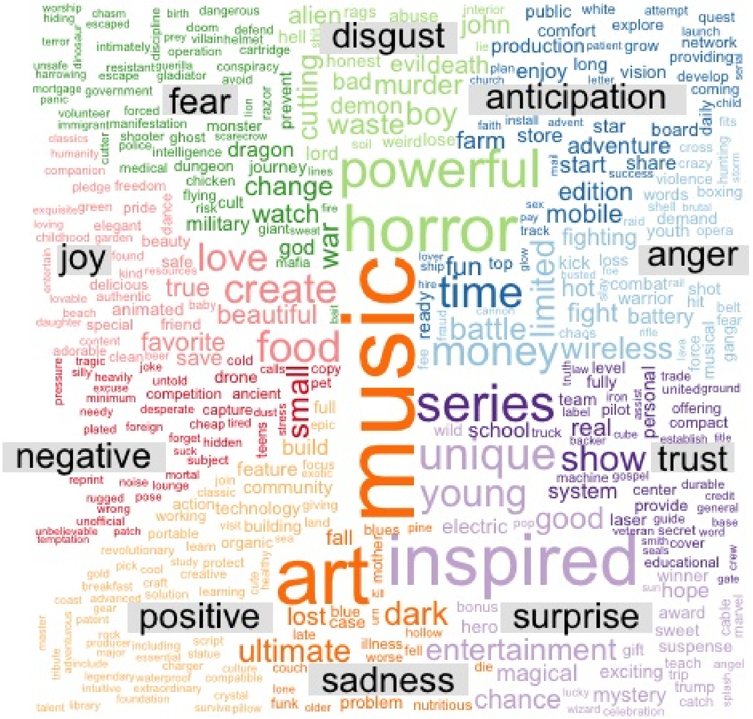


Source: https://webrobots.io/kickstarter-datasets/

I picked four emotions from the NRC dictionary to measure each of it's relationship with the success of a project. The relationships all have similar trajectory through different achievement ratio. For joe and positive, We see similar dips and peaks in counts of sentiment at the same achievement ratio, which suggests that there might be a lot of overlaps in these two groups of emotions. Overall, the count for positive emotions is higher than all other emotions by looking at the y coordinate.

Comparison cloud of all emotion types in NRC

```
nrc <- get_sentiments("nrc")

# Create a Comparison Cloud for 10 types of emotions
blurb_NRC <- inner_join(blurb_tidy, nrc, by = c("term" = "word")) %>%
   # Count by sentiment, index, document
  count(term, sentiment) %>%
  acast(term ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = brewer.pal(10,"Paired"), max.words=800,
                   title.size=1.4, random.order=FALSE,c(4,0.4))
```



```
blurb_NRC
```

```
## NULL
```

Submission

Please follow the instructions to submit your homework. The homework is due on Wednesday, March 31.

Please stay honest!

If you do come across something online that provides part of the analysis / code etc., please no wholesale copying of other ideas. We are trying to evaluate your abilities to visualized data not the ability to do internet searches. Also, this is an individually assigned exercise – please keep your solution to yourself.