Landscape of Funding for Kickstarter Campaigns

Ibrahim Taher Anant Jain Sarthak Kothari Forrest Hooton

Summary

Kickstarter.com is one of the iconic embodiments of a crowdsourced economy. It was one of the first platforms that created an environment for entrepreneurs to share their ideas, and have their ideas funded. This platform operates through campaigns, which are pages where an entrepreneur motivates his/her project and describes the needs the project attempts to fulfill. The specific funding goal, total number of “backers” (people who give money to projects), and quantity of funding already pledged are described on the campaign page along with the descriptions. Each campaign lasts over a period of 30 - 60 days, and backers’ pledges are only collected if the project is successfully funded. To date, almost 15 million backers have participated to successfully fund over 140,000 projects [1].

Kickstarter’s historical precedent for online crowdsourcing and abundance of data from 300,000+ campaigns makes it an ideal case study to explore what factors affect whether or not the crowd economy funds a project. Specifically, the goal for this project is to understand the landscape of funding for kickstarter, i.e. how different factors relate to a project’s success. Additionally, it would be interesting to know what can be modeled and attempt predicting the outcome of project campaigns. The joint dataset used for this project contains 321,623 rows, with 21 columns (features). These features focused on aspects of campaign timeframes, countries + currencies, funding, backers, and textual description.

Methods

The goal understanding the landscape of kickstarter funding was split into three separate stages: data cleaning, exploration, and modeling. This design intuitively built on the preceding step, as clean data is necessary for exploration, and good exploration expedited modeling. After these three stages, the findings were integrated into a Shiny app as a visualization for interactive communication with the audience during the presentation of the findings.

Cleaning

The core of the dataset was the Kickstarter dataset from Kaggle [2], but additional features that could be useful were added from scraped JSON files [3]. For example, the Kaggle kickstarter dataset did not include much text from each project and the JSON files contained the entire text description for each project. The usefulness of access to both these data sources was maximized by joining the already organized and clean Kaggle dataset with the interesting features from the JSON dataset. Fortunately, all projects have a unique campaign key, even if a single project had multiple campaigns, so the join was straightforward. The last aspect of data cleaning was converting the project start dates and deadline dates to datetime objects using lubridate, and adding features to specify whether or not a project was successful and what percent both successful and unsuccessful projects were funded.

Exploration

The exploration stage consumed the most time, as it was the core to at least understanding more about the landscape of funding. It began with an exploration of general Kaggle dataset, then pushed deeper by exploring funding specifically. To explore funding, the relationships of funding variables to the other features in the dataset were observed. Because of all the different countries which use kickstarter, features that were uniform to all areas of the world, such as main category, were the center focus. The dataset was divided into an exploratory set and a modeling set during the process of digging into the funding landscape to preserve modeling integrity later in the project.

While exploring the metadata was an interesting task on its own, understanding the language used by entrepreneurs in the provided summaries seemed to be an interesting point to understand how language might have affected the success of the campaign. The initiative to understand this aspect of kickstarter stemmed from common advertising practice. It is well known that advertisers on television, radio, billboards, etc. attempt to coerce consumers into buying their products using combinations of words that might appeal to the user. The goal of entrepreneurs on kickstarter

seemed to align with that of companies that advertise on other forms of media: convince a consumer/backer to buy into an idea and potentially receive a great product. In understanding the text, it was hypothesized that heterogeneity could be found between success and failure based on words used.

Modeling

Once the important features that directly affect the outcome of the Kickstarter campaign being successful or not were identified, they were built into a few model to attempt to predict the outcome of the campaign.

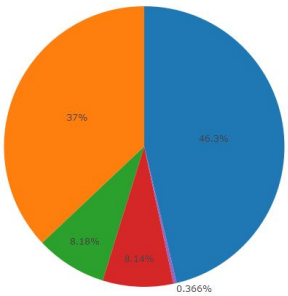
Results

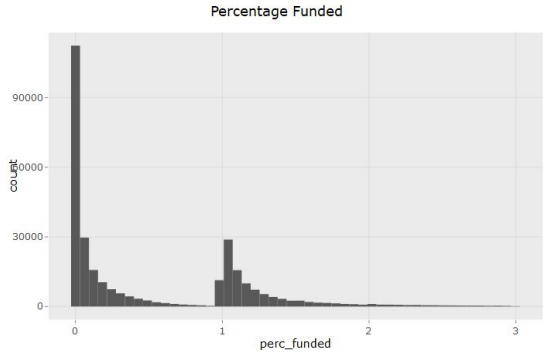
The examination of is divided into three parts of exploratory results: the preliminary data exploration, percent of funding, and sentiment analysis. These three exploratory journeys are followed by observed modeling results. The exploratory results consist of graphs, while the modeling walkthrough displays both graphs and tables of model error.

Exploratory Results

Introductory Exploration

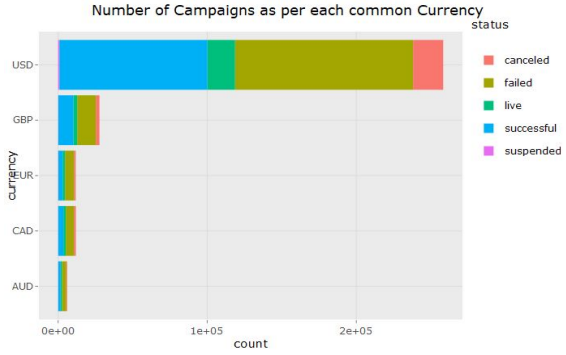
Somewhat surprisingly, only 37% of projects have actually been successful since kickstarter began. As shown by the first peak of “Percentage Funded” below., most campaigns didn't get funded at all, while the raise before the second peak indicates that some almost did. The second peak itself shows the number of campaigns which successfully got funded.



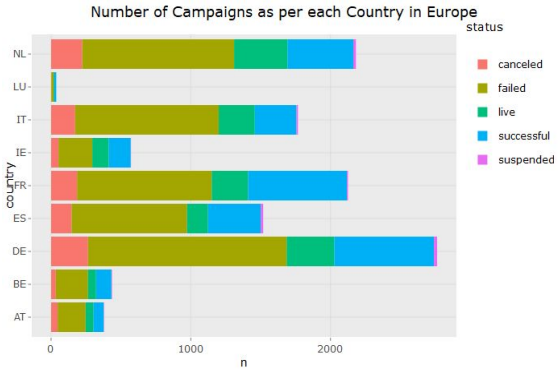


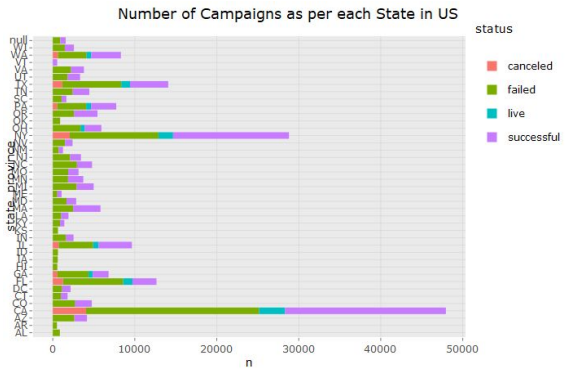
Currency and Region

The plot below shows the fraction of each campaign state/status prevalent under each currency in the kickstarter dataset. Judging from the plot, type of currency does not affect successes or failures.



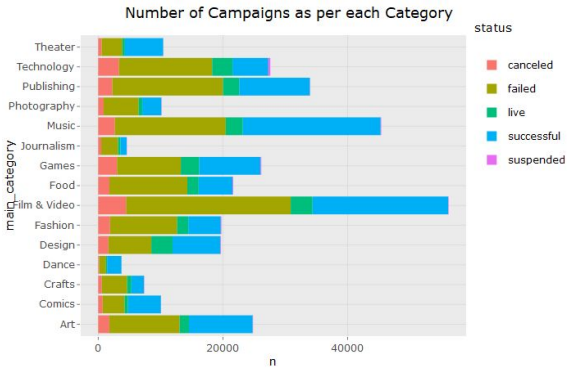
The region of origin effects were also explored. In the plots below, Germany is shown to have the highest number of campaigns in Europe, but it seems the trend of higher failure rates continues here. Also, California has the highest number of campaigns in the US, but the trend of higher failure rates occurs here too; New York which is an exception here, supporting the claim that dreams do come true in New York.

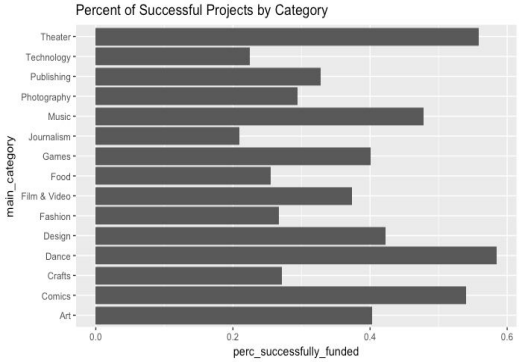




Category

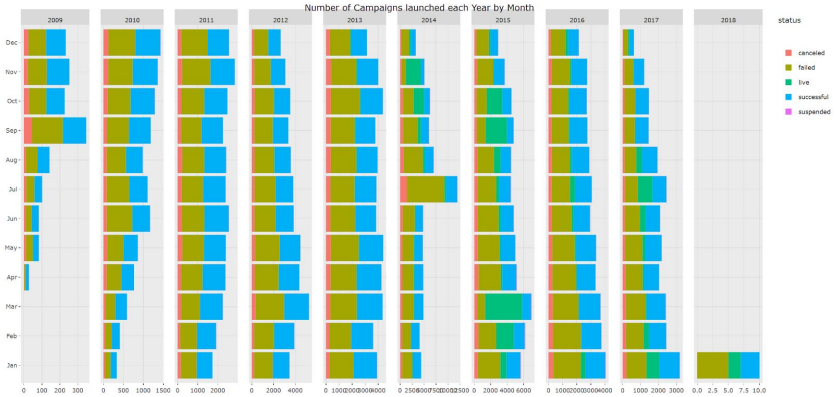
Film and Music are shown in the graph below to have the highest number of campaigns. Furthermore, Dance, Theatre and Comics seem to be successful the most frequently. Journalism, Food, and Fashion seem to fail the most frequently.





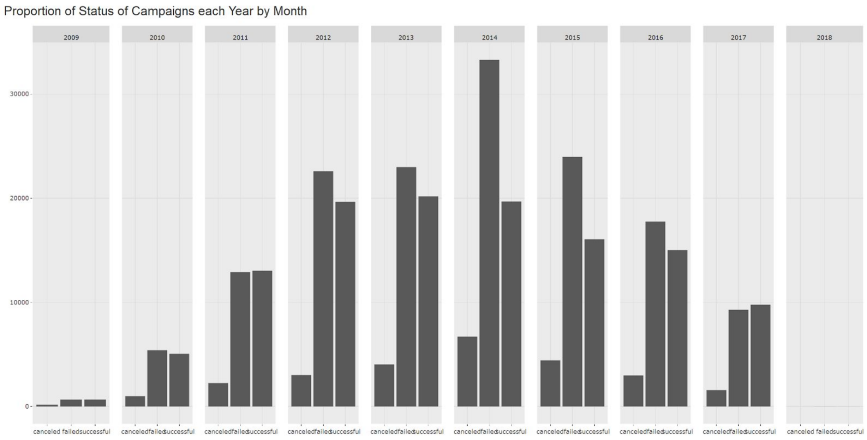
Time Period

The temporal aspect of whether or not a project is funded does not seem to show particularly evident seasonality. It looks like earlier the number of campaigns increased as the year progressed, but in the recent years the most number of campaigns are at the start and then decreases. There seems to be slight spike in July, maybe because of summer projects.



However, the graphs do show some interesting funding behavior by year. A much higher percent of projects were funded in 2009, and did not seem to reach a steady rate until later 2010. This might be due to the fact that Kickstarter began in 2009, and more enthusiasm for funders or more interesting projects drove up the percentage of successfully

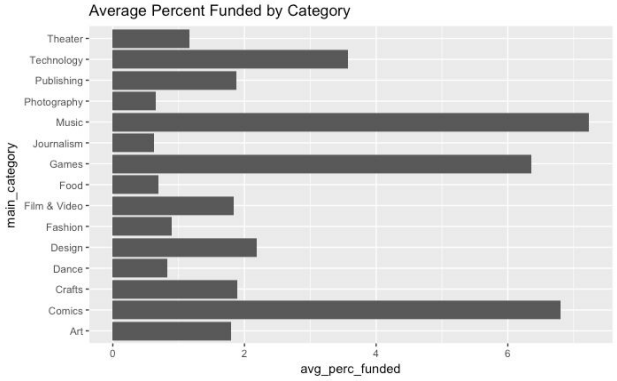
funded campaigns. In late 2010, the site was probably more well known, less “new”, and funding enthusiasm became more consistent. Additionally, the number of projects grew to a higher quantity, as examined before, so naturally a smaller percentage of projects would be funded. In 2017, the percent of successful projects rose again. Therefore, the landscape of funding does depend on the time of year according to the data, but is not highly explanatory.

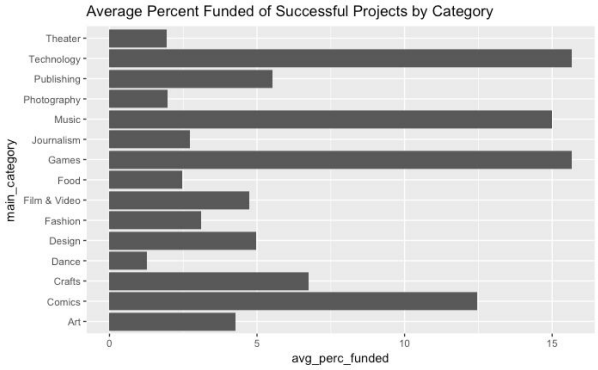


Percent of Funding

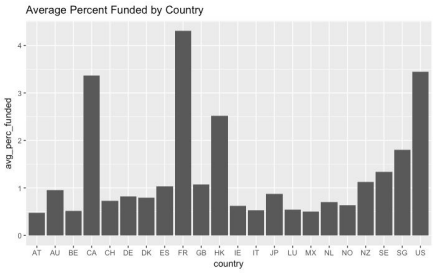
The average percentage of funding for kickstarter projects receive gives a more in depth perspective of a projects success. At times, a projects funding can far exceed its goal. For instance, while the median funding percentage for all projects is 16%, the 3rd quartile is 107.88%. The funding percentage increases even more dramatically for the top; the highest funding percentage of all time was 68,764%.

The main categories average funding percentage across all projects reveal some very interesting insights. The average percent funding is very high at 1500% for some categories, and very low for others. This is likely due to a few large outliers pulling up the rest of the data. But even more interesting, the categories that had the highest percentage of funded projects have two of the lower averages the the percent of a project funded, and the two lowest averages for the average percentage of successfully funded projects. However, while numerous categories are inverse, the comics category are dominant both in terms of percent of projects funded and average percentage funded per project.



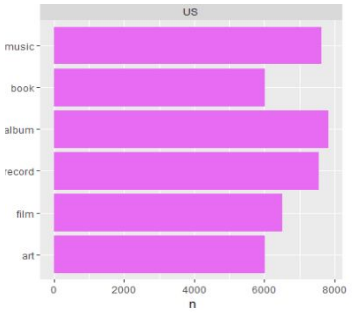


The graphs for the average funding percentage look very different than the percent of funded projects in relation to countries, so much so the average for all projects is only worth a glance. France has the highest average for funding percentage, but the United States trails in second. This was not exactly what was expected due to such a large volume of projects being in the US, but it it turns out so is the total quantity of funding.

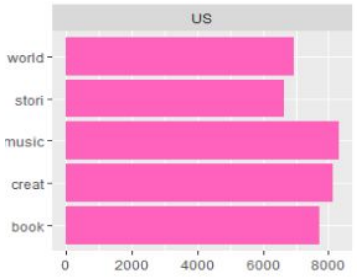


Text Analysis

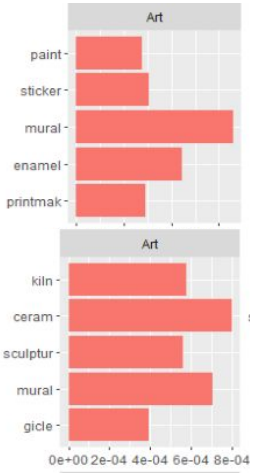
The language used by entrepreneurs proved to be an interesting aspect of the project. The text was stripped of stop words then stemmed and then explored using a unigram or bigram approach and then importance was casted using simple counts or TF-IDF. Heterogeneity between success and failed campaigns was explored by looking at campaigns for each country or category. This was motivated by the idea that what might make a successful and/or failed campaign in the United States might be different than in Great Britain, and (more obviously) what makes a successful and/or failed campaign in Music is much different than Technology.



A unigram approach with counts was used to attempt to disambiguate the successful and failed campaigns. There was not much difference between successful and failed campaigns for both countries and categories. This makes sense: popular words would appear in both types of campaigns. For example, Music is one of the more popular types of campaigns in the United States, so it is sensible that music showed up both as one of the top 5 most common words in successful (right top) and failed campaigns (right bottom) in the US.



The same was present in successful and failed campaigns by category. Words that helped really define the category appeared as the most popular words in the categories. I.e. in the music category, words such as music, album and song appeared both in the successful and failed campaigns and this type of pattern continued for other categories. There was no heterogeneity observed because these words are the most popular words in each category/country and they simply just appear too much. However, these words might not be the most important words for each country or category so a metric for word importance, known as TF-IDF was used to understand the importance of words by country and category.



Using TF-IDF with unigrams showed better results in terms of heterogeneity, but it still was not enough to disambiguate categories. For example, with the United States, words that were associated with music were still the most popular and among other nations, words associated with nationality were popular among both successful and failed categories (i.e. Australia and Sydney for Australia.) Unigrams with TF-IDF showed more heterogeneity for categories. For example looking at the figures on the left, success (top) and failure (bottom) were extremely different in terms of the words that were considered the most important by the TF-IDF metric. The TF-IDF metric and disambiguating by categories was producing better result for heterogeneity between successful and failed campaigns. In order to increase heterogeneity, the notion of bigrams were introduced as the tokens instead of unigrams. Also, using categories was producing better results for campaigns so it would just be bigrams with TF-IDF broken down by category. While certain words might appear more in each bigram, the other word in the bigram would give more context as to its reason. For example, music may be really popular as a unigram in both successful and failed campaigns but rock music might show up more in successful campaigns and pop music might show up more in failed campaigns.

What begins to arise when the bigrams are examined by categories are important distinctions between what might make a successful/failed campaign. For example, in design if you create a campaign that has to do with an Apple product, it seems like those would fail; whereas things with more utility like multi-tooled objects, wallets, and stainless steel objects are more important among successful projects. In photography it is observed that nudes and adult content seem to be important for successful categories whereas books of photos that might go on your coffee table are likely to fail. While true heterogeneity appears in many categories, identifying bigrams that were associated with success or failure in certain categories, such as music, was unfruitful. These results might have to do with the landscape of individual categories. Music, as a category, mainly involves individuals trying to promote their albums where as design has more built in variety of products.

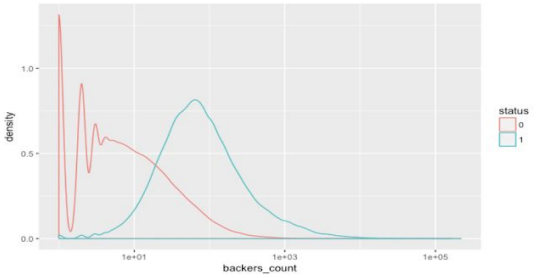
Overall, it seems as though what makes a successful campaign are products that seem much more interesting or appealing to the potential backer. iPhone 4 products are very common already in the real world, which might make a backer less likely to back their product; whereas, minimalist wallets have not truly broken into the main industry and

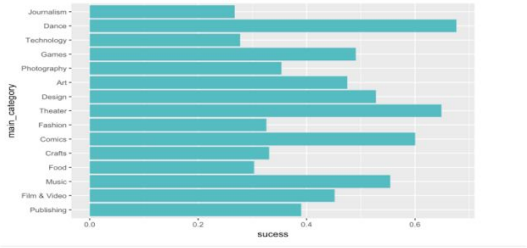
generally find themselves as product ideas on kickstarter. More obviously, photo albums are sold regularly at department stores; whereas, tasteful adult content is not as mainstream and therefore might be more likely to gather pledges.

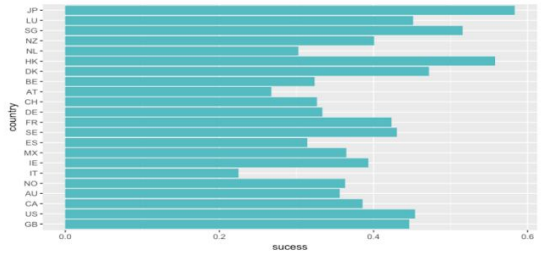
Modeling Results

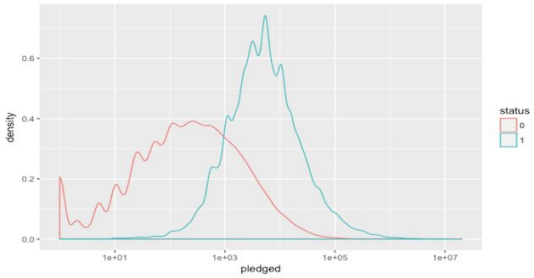
Three different models were built for the data.The primary logistic regression model which would decide the outcome, a Linear model which would predict what percent of the goal got funded, and lastly a random forest model which would predict the outcome also. The result and scores were of the logistic model were then compared to the random forest model.

Since the target variable has a binomial distribution, continuous variable and categorical variable as density plots bar plots, respectively, indicating the ratio of success to failure. The models were tweaked incrementally to validate that the model did not overfit. Additionally, a K-fold cross validation was implemented, which gave similar results across all folds.









In the graphs, the feature ‘backers count’ has different density for success and failure; when the ‘backers count’ is smaller, there are more projects that have been unsuccessful. Similarly, the variable pledge has different densities over the success and failed outcomes.

Different categories have different success rates. Categories like Dance, Theater, and Comics have almost 70% success ratio compared to Technology and Journalism which have only 30% success ratio. It is also observed that countries like Japan, Hong Kong, Singapore have 50% success ratio whereas countries like Italy, Austria have less than 30% success ratio.

As stated previously, three models were created. The logistic model was the focus for this project, but the errors of the random forest and linear regression are shown below. The random forest actually had higher accuracy, showing that it would actually be the better model to use for prediction. In regards to the linear model, the variables for the fit were chosen by simply using the best variables for logistic regression. While this is no the most technically accurate method, it was previously known that the RMSE would be terrible with any model due to the wild outliers in regards to the percent funded in some of the projects; the linear regression was run simply to note how bad the prediction would be. It is also worth noting that all of these models use variables that could only be gathered after a campaign had concluding, thus limiting their effectiveness as predictive (as opposed to descriptive) models.

Predictive Method	Model	Accuracy
Logistic Regression	status ~ main_category + backers_count + country + pledged + Month + Year + days_live	83%
Random Forest	status ~ main_category + backers_count + spotlight + Month + Year + days_live	88%
Linear Regression	perc_funded ~ main_category + backers_count + country + pledged + Month + Year + days_live	133 RMSE

Discussion

Overall, deeper insight was gained through exploratory analysis and modeling. The exploratory analysis revealed some key insights. For instance, the highly disproportionate volume of projects in the United States makes it the primary focus for exploratory analysis and modeling with plentiful data; the insight from other nations is more limited and eliminates the significant of features like currency. Additionally, this paper explored the general growth patterns of kickstarter funding, the potential for outliers to far exceed the norm, and how different categories affect the chances of being funded. Within categories, using the TF-IDF metric combined with bigram tokens; a relationship between certain words in categories and success was developed (i.e. adult content in photography.)

This project was even able to model, with reasonable accuracy, whether or not a project was funded. However, while there was success in modeling campaigns historically, the models failed to perform well predictively. The models in this paper used features that could only be determined after a project concluded, such as number of backers for the project. While the importance of the “number of backers” feature does illustrate that “number of backers” is an important variable, this value would not be known before a campaign ended. Therefore, “number of backers” would not help actually predicting whether or not a project is successful. In future iterations of this project, another feature that might have predictive power would be bigrams in each campaign. As discussed in text analysis, bigrams discovered with TF-IDF showed some level of heterogeneity between success and failure so it is possible that these bigrams could increase the accuracy of the models solely for prediction.

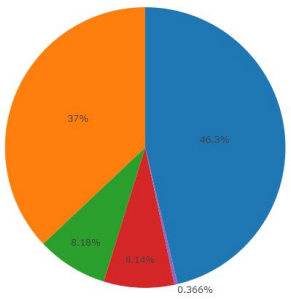
In conclusion, import factors in whether or not a project is funded could be understood and interpreted, but the models could not make accurate predictions in a real world scenario. This is especially true for any type of linear prediction, which makes since due to extreme outliers. In reality, the dataset does not include important factors that dramatically affect the outcome of a campaign. People back campaigns that they find novel and useful, and it takes complex understanding of entire industries to identify if a single campaign embodies these values. This project may be able to evaluate certain aspects with the used dataset, but the true heart of Kickstarter that drives funding evades us for now.

References

[1] “Kickstarter.” *Kickstarter*, www.kickstarter.com/.
[2] Kemical. “Kaggle.” *Kickstarter Projects* , 8 Feb. 2018, www.kaggle.com/kemical/kickstarter-projects/data.
[3] “Kickstarter Datasets.” Web Scraping Service, webrobots.io/kickstarter-datasets/.

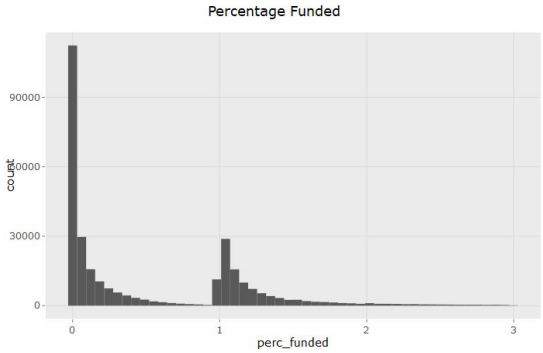
Appendix

(1)



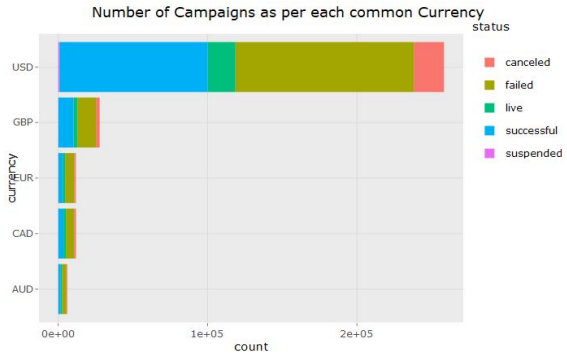

```
ks %>%
        group_by(status) %>%
        summarise(n = n()) %>%
        plot_ly(labels = ~status, values = ~n, type = 'pie')
```

(2)



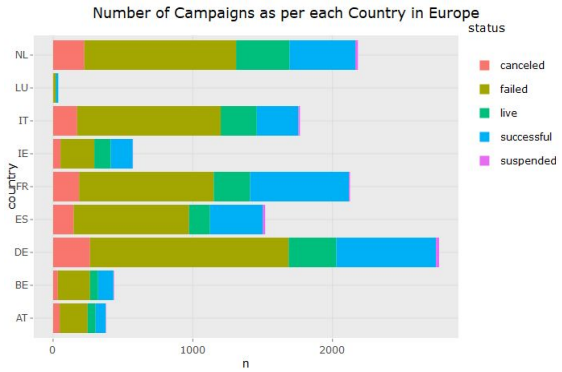

```
ks  %>%
        filter(perc_funded < 3) %>%
        ggplot() + geom_histogram(aes(perc_funded), bins =50) +
        ggtitle("Percentage Funded")
```

(3)



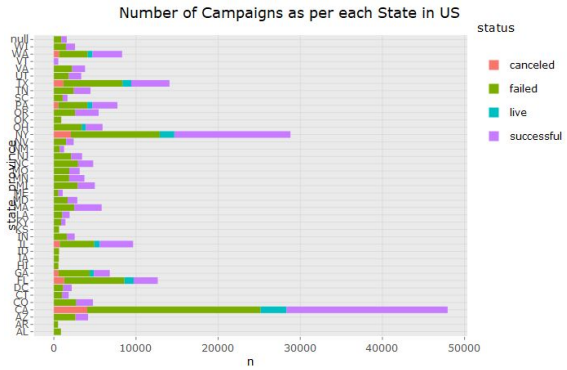

```
ks <- ks %>%
        filter(currency == "USD" | currency == "GBP" | currency == "EUR" | currency == "CAD" | currency == "AUD")
ggplotly(ggplot(ks, aes(x = currency, fill = status)) +
        geom_bar() +
        coord_flip() +
        ggtitle("Number of Campaigns as per each common Currency"))
```

(4)




```
ks %>%
        group_by(country,currency,status) %>%
        summarise(n = n()) %>%
        filter(currency == "EUR") %>%
        ggplot(aes(x=country, y=n, fill=status)) +
        geom_col() +
        coord_flip()+           ggtitle("Number of Campaigns as per each Country in Europe")
```

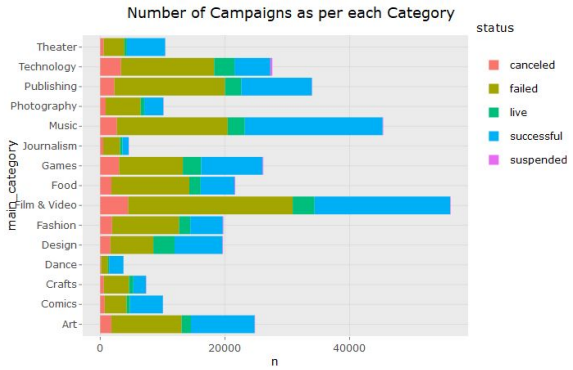
(5)




```
ggplotly(ks %>%
        filter(country == "US") %>%
        group_by(state_province, status) %>%
        summarise(n = n()) %>%
        filter(n>500) %>%
        ggplot(aes(x=state_province, y=n, fill=status)) +
```

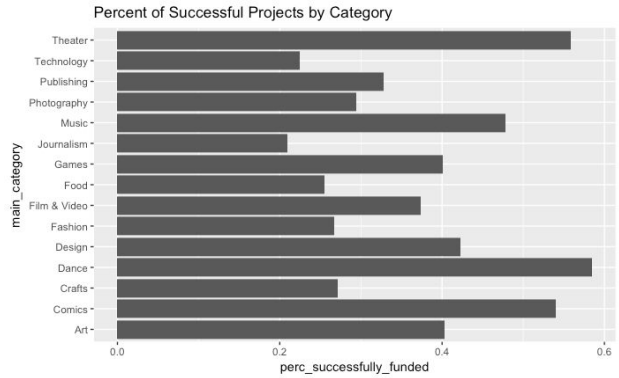
```
        geom_col() +
        coord_flip()+          ggtitle("Number of Campaigns as per each State in US")
```

(6)



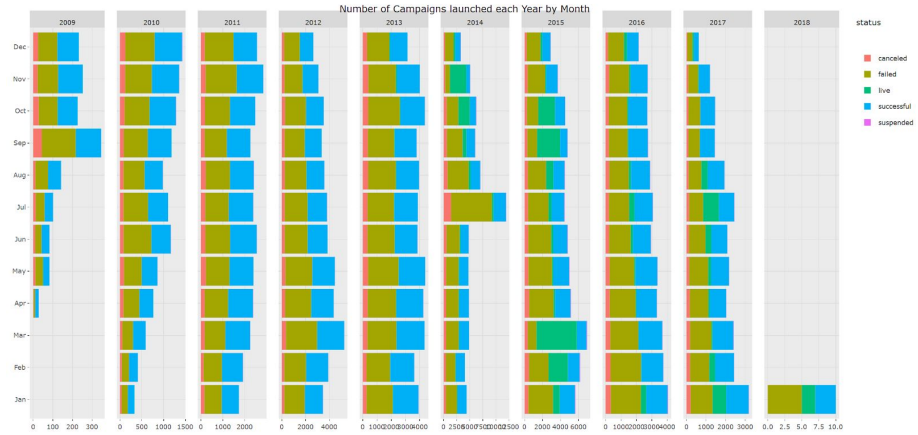

```
ggplotly(ks %>%
        group_by(main_category, status) %>%
        summarise(n = n()) %>%
        ggplot(aes(x=main_category, y=n, fill=status)) +
        geom_col() +
        coord_flip()+          ggtitle("Number of Campaigns as per each Category")
```

(7)



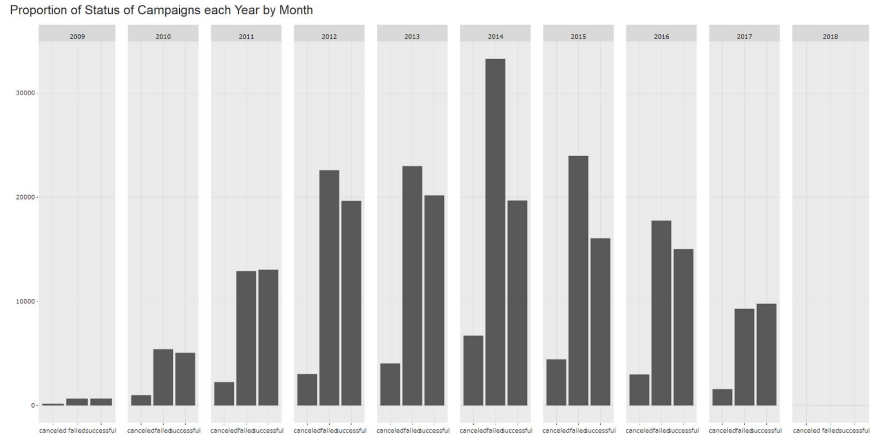

```
# Total number of projects per category
category_count <- exp_data %>%
  group_by(main_category) %>%
  summarise(
    num = n()
  )
# Get the percent of successful projects per category
exp_data %>%
  filter(is_funded == TRUE) %>%
  left_join(category_count, by = c('main_category' = 'main_category')) %>%
  group_by(main_category) %>%
  summarise(
    perc_successfully_funded = n() / mean(num)
  ) %>%
  ggplot() + geom_col(aes(x = main_category, y = perc_successfully_funded)) + ggtitle("Percent of Successful Projects by Category") + coord_flip()
```

(8)



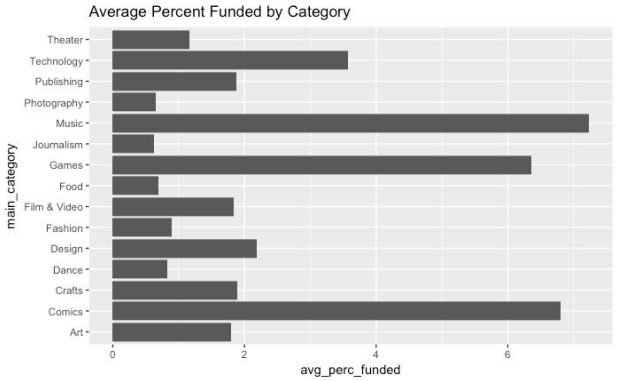

```
ks %>%
        group_by(Year, Month, status) %>%
        summarise(n = n()) %>%
        ggplot(aes(x=Month, y=n, fill=status)) +
        geom_col() +
        coord_flip() +
        ggtitle("Number of Campaigns launched each Year by Month") +
        facet_grid(~Year, scales = "free")
```

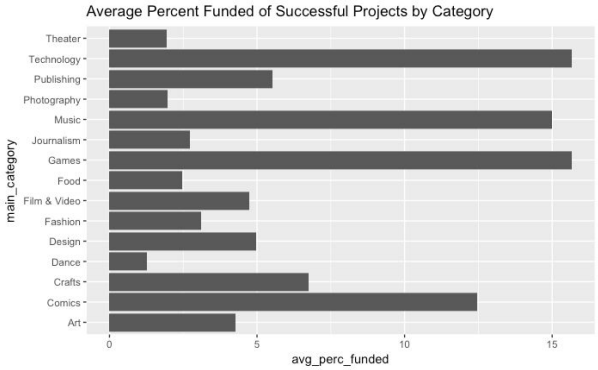
(8)




```
ggplotly(ks %>%
        filter(status == "canceled" | status == "failed" | status == "successful" ) %>%
        group_by(status, Year) %>%
        summarise(n = n()) %>%
        ggplot() +
        geom_col(aes(status,n)) +
        facet_grid(~Year)
```

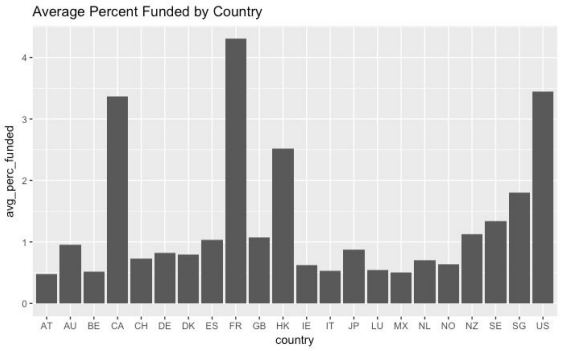
(9)





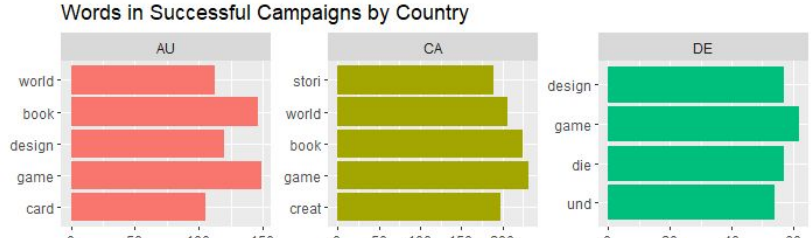

```
# Get the average percent funded of projects per category
exp_data %>%
  group_by(main_category) %>%
  summarise(
    avg_perc_funded = mean(perc_funded)
  ) %>%
  ggplot() + geom_col(aes(x = main_category, y = avg_perc_funded)) + ggtitle("Average Percent Funded by Category") + coord_flip()

# Get the average percent funded of successfully funded projects per category
exp_data %>%
  filter(is_funded == TRUE) %>%
  group_by(main_category) %>%
  summarise(
    avg_perc_funded = mean(perc_funded)
  ) %>%
  ggplot() + geom_col(aes(x = main_category, y = avg_perc_funded)) + ggtitle("Average Percent Funded of Successful Projects by Category") +
coord_flip()
```

(10)



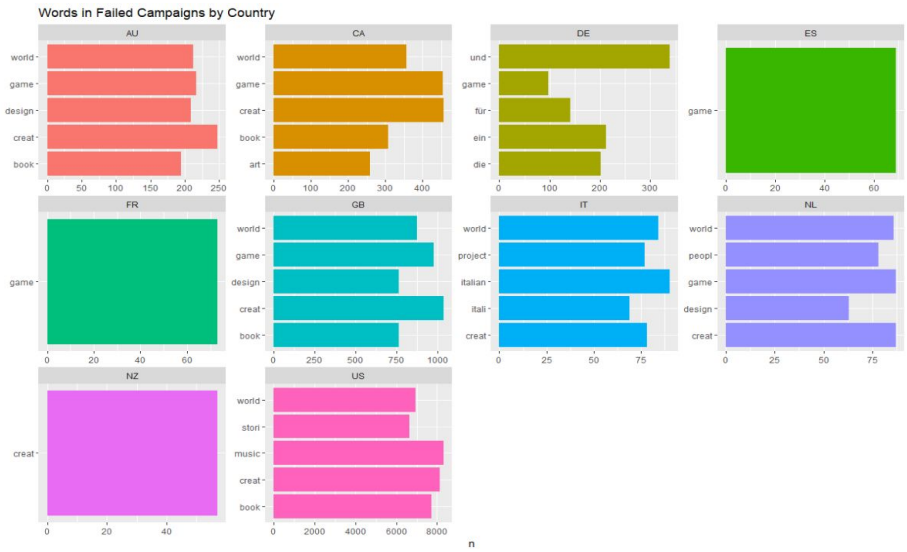

```
# Get the average percent funded of successfully funded projects per country
exp_data %>%
  filter(is_funded == TRUE) %>%
  group_by(country) %>%
  summarise(
    avg_perc_funded = mean(perc_funded)
  ) %>%
  ggplot() + geom_col(aes(x = country, y = avg_perc_funded)) + ggtitle("Average Percent Funded of Successful Projects by Country") + coord_flip()
```

(11)



```
tidyTextKS %>%
  select(country,stemmed,status) %>%
  filter(status=='successful') %>%
  mutate(stemmed =factor(stemmed, levels =rev(unique(stemmed)))) %>%
  group_by(country,stemmed) %>%
  summarise(count=n()) %>%
  arrange(desc(count)) %>%
  filter(count>51) %>%
  top_n(5) %>%
  ungroup() %>%
  ggplot() + geom_col(mapping = aes(x=stemmed, y=count, fill=country),
                      show.legend = FALSE) +
  ggtitle("Words in Successful Campaigns by Country") +
  labs(x = NULL, y = "n") + facet_wrap(~country, scales='free') + coord_flip()
```

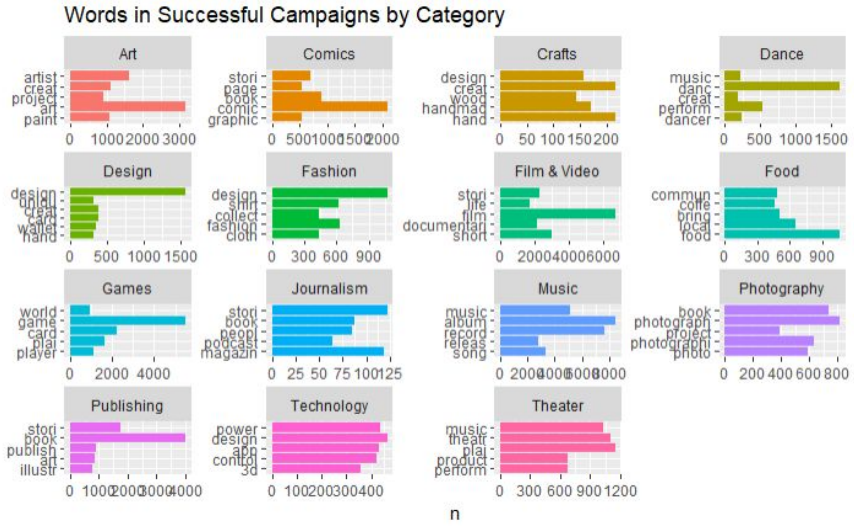
(12)




```
tidyTextKS %>%
  select(country,stemmed,status) %>%
  filter(status=='failed') %>%
  mutate(word =factor(stemmed, levels =rev(unique(stemmed)))) %>%
  group_by(country,stemmed) %>%
  summarise(count=n()) %>%
  arrange(desc(count)) %>%
  filter(count>51) %>%
  top_n(5) %>%
```

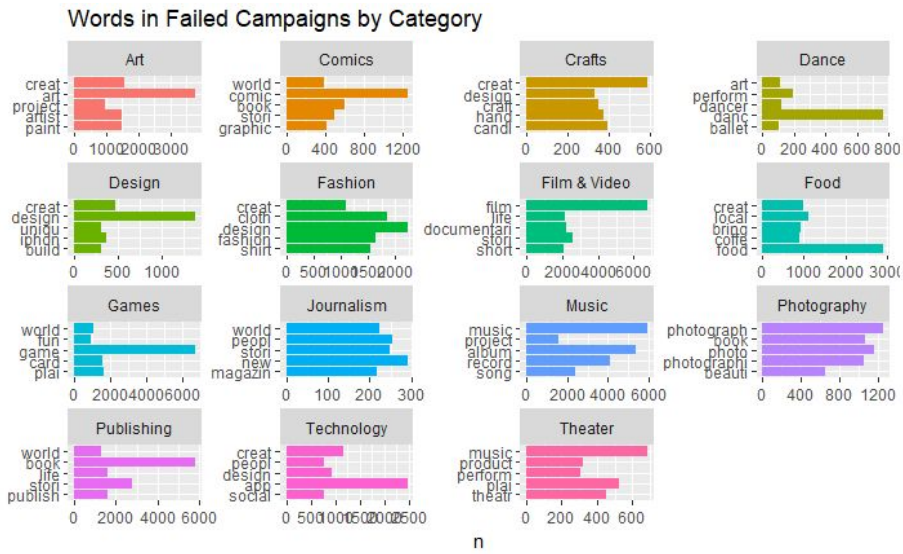
```
  ungroup() %>%
  ggplot() + geom_col(mapping = aes(x=stemmed, y=count, fill=country),
                      show.legend = FALSE) +
  ggtitle("Words in Failed Campaigns by Country") +
  labs(x = NULL, y = "n") + facet_wrap(~country, scales='free') + coord_flip()
```

(13)



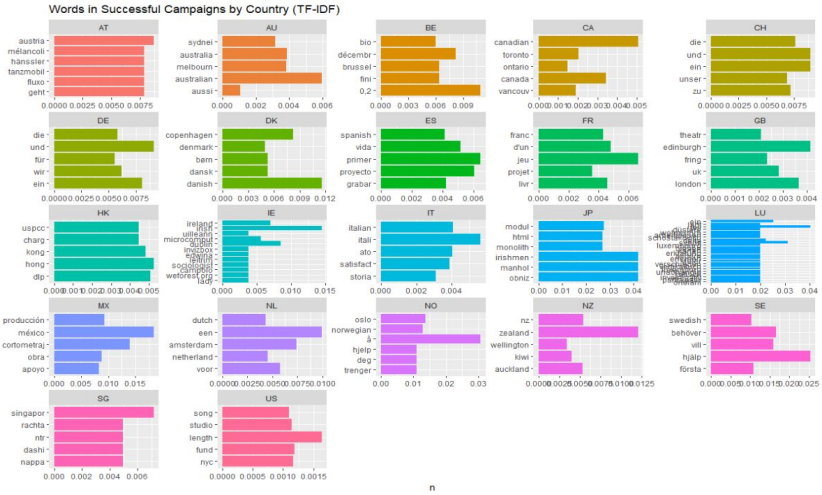

```
tidyTextKS %>%
  select(main_category,stemmed,status) %>%
  filter(status=='successful') %>%
  mutate(stemmed =factor(stemmed, levels =rev(unique(stemmed)))) %>%
  group_by(main_category,stemmed) %>%
  summarise(count=n()) %>%
  arrange(desc(count)) %>%
  filter(count>50) %>%
  top_n(5) %>%
  ungroup() %>%
  ggplot() + geom_col(mapping = aes(x=stemmed, y=count, fill=main_category),
                      show.legend = FALSE) +
  ggtitle("Words in Successful Campaigns by Category") +
  labs(x = NULL, y = "n") + facet_wrap(~main_category, scales='free') + coord_flip()
```

(14)




```
tidyTextKS %>%
  select(country,stemmed,status) %>%
  filter(status=='failed') %>%
  mutate(word =factor(stemmed, levels =rev(unique(stemmed)))) %>%
  group_by(country,stemmed) %>%
  summarise(count=n()) %>%
  arrange(desc(count)) %>%
  filter(count>51) %>%
  top_n(5) %>%
  ungroup() %>%
  ggplot() + geom_col(mapping = aes(x=stemmed, y=count, fill=country),
                      show.legend = FALSE) +
  ggtitle("Words in Failed Campaigns by Country") +
  labs(x = NULL, y = "n") + facet_wrap(~country, scales='free') + coord_flip()
```

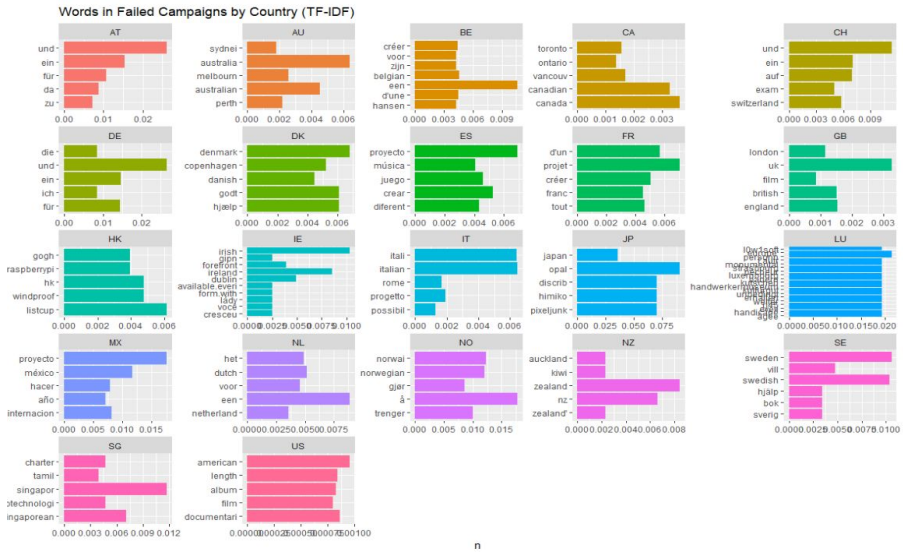
(15)




```
tidyTextKS %>%
  select(country,stemmed,status) %>%
  filter(status=='successful') %>%
  mutate(stemmed =factor(stemmed, levels
=rev(unique(stemmed)))) %>%
  group_by(country,stemmed) %>%
  summarise(count=n()) %>%
  bind_tf_idf(stemmed, country, count) %>%
  arrange(desc(tf_idf)) %>%
  top_n(5) %>%
  ungroup() %>%
  ggplot() + geom_col(mapping = aes(x=stemmed, y=tf_idf, fill=country),
```

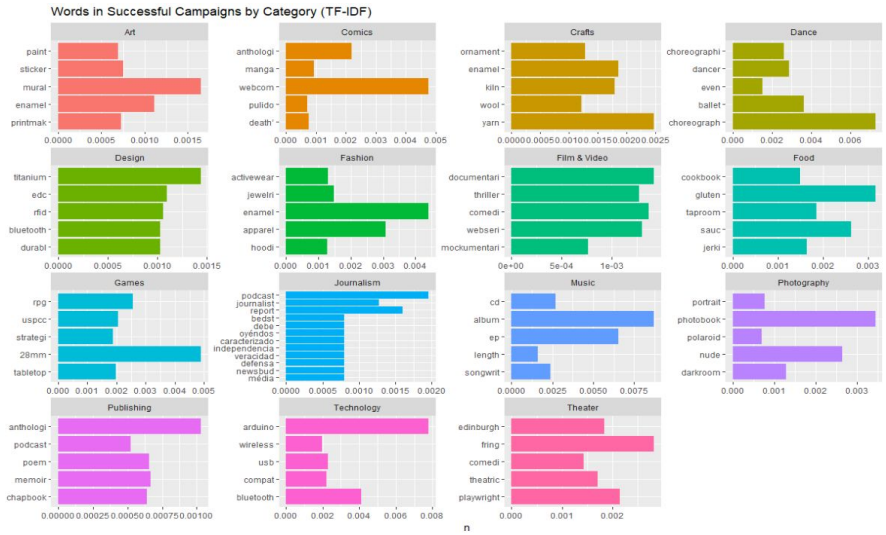
```
                    show.legend = FALSE) +
  ggtitle("Words in Successful Campaigns by Country (TF-IDF)") +
  labs(x = NULL, y = "n") + facet_wrap(~country, scales='free') + coord_flip()
```

(16)




```
tidyTextKS %>%
  select(country,stemmed,status) %>%
  filter(status=='failed') %>%
  mutate(stemmed =factor(stemmed, levels =rev(unique(stemmed)))) %>%
  group_by(country,stemmed) %>%
  summarise(count=n()) %>%
  bind_tf_idf(stemmed, country, count) %>%
  arrange(desc(tf_idf)) %>%
  top_n(5) %>%
  ungroup() %>%
  ggplot() + geom_col(mapping = aes(x=stemmed, y=tf_idf, fill=country),
                      show.legend = FALSE) +
  ggtitle("Words in Failed Campaigns by Country (TF-IDF)") +
  labs(x = NULL, y = "n") + facet_wrap(~country, scales='free') + coord_flip()
```

(17)




```
tidyTextKS %>%
  select(main_category,stemmed,status) %>%
  filter(status=='successful') %>%
  mutate(stemmed =factor(stemmed, levels =rev(unique(stemmed)))) %>%
  group_by(main_category,stemmed) %>%
  summarise(count=n()) %>%
  bind_tf_idf(stemmed, main_category, count) %>%
  arrange(desc(tf_idf)) %>%
  top_n(5) %>%
  ungroup() %>%
  ggplot() + geom_col(mapping = aes(x=stemmed, y=tf_idf, fill=main_category),
                      show.legend = FALSE) +
  ggtitle("Words in Successful Campaigns by Category (TF-IDF)") +
  labs(x = NULL, y = "n") + facet_wrap(~main_category, scales='free') + coord_flip()
```

(18)

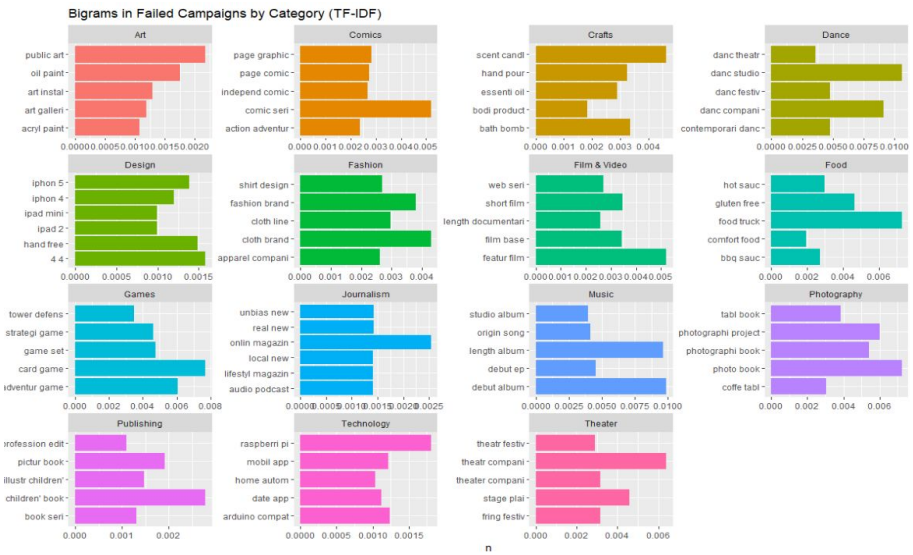

```
tidyTextKS %>%
  select(main_category,stemmed,status) %>%
  filter(status=='failed') %>%
  mutate(stemmed =factor(stemmed, levels =rev(unique(stemmed)))) %>%
  group_by(main_category,stemmed) %>%
  summarise(count=n()) %>%
```

```
  bind_tf_idf(stemmed, main_category, count) %>%
  arrange(desc(tf_idf)) %>%
  top_n(5) %>%
  ungroup() %>%
  ggplot() + geom_col(mapping = aes(x=stemmed, y=tf_idf, fill=main_category),
                      show.legend = FALSE) +
  ggtitle("Words in Failed Campaigns by Category (TF-IDF)") +
  labs(x = NULL, y = "n") + facet_wrap(~main_category, scales='free') + coord_flip()
```

(19) Code to develop bigram tibble

```
tidyTextKSBigrams<-kickstarter %>%
  select(Year,id,country,blurb,main_category,category_name,status) %>%
  unnest_tokens(word,blurb, token="ngrams", n=2)

tidyTextKSBigramsNoStop <- tidyTextKSBigrams %>%
  separate(word,into = c("word1","word2"), sep = " ") %>%
  filter(!word1 %in% custom_stop_words$word,
         !word2 %in% custom_stop_words$word) %>%
  mutate(stem1=wordStem(word1),stem2=wordStem(word2))
```

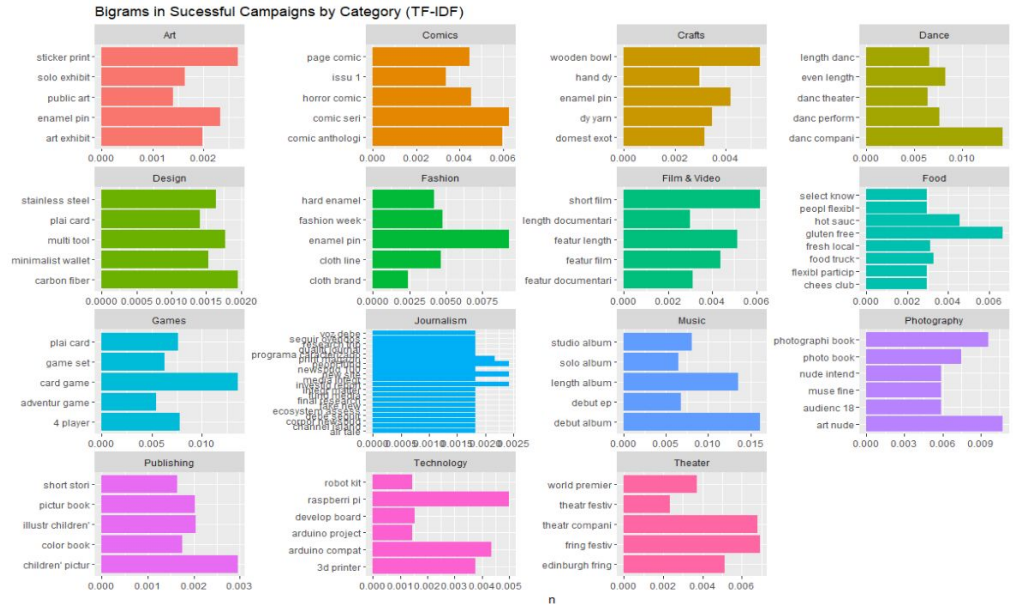
(20)




```
tidyTextKSBigramsNoStop %>%
  unite(stemmed, stem1, stem2, sep=" ") %>%
  filter(status=='failed') %>%
  group_by(main_category,stemmed) %>%
  summarise(count=n()) %>%
```

```
  bind_tf_idf(stemmed, main_category, count) %>%
  arrange(desc(tf_idf)) %>%
  top_n(5) %>%
  ungroup() %>%
  ggplot() + geom_col(mapping = aes(x=stemmed, y=tf_idf, fill=main_category),
                      show.legend = FALSE) +
  ggtitle("Bigrams in Failed Campaigns by Category (TF-IDF)") +
  labs(x = NULL, y = "n") + facet_wrap(~main_category, scales='free') + coord_flip()
```

(21)

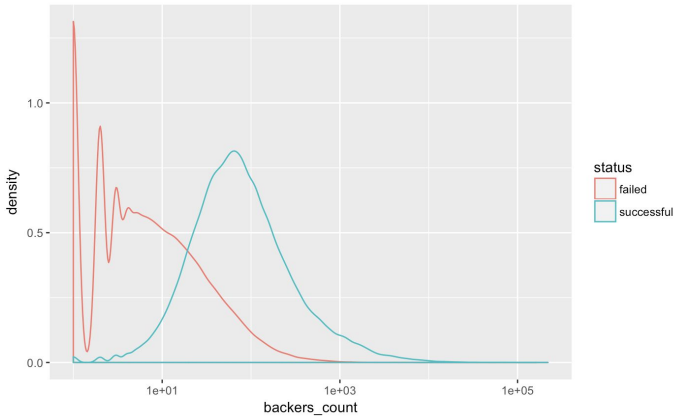



```
tidyTextKSBigramsNoStop %>%
  unite(stemmed, stem1, stem2, sep=" ") %>%
  filter(status=='successful') %>%
  group_by(main_category,stemmed) %>%
  summarise(count=n()) %>%
  bind_tf_idf(stemmed, main_category, count) %>%
  arrange(desc(tf_idf)) %>%
  top_n(5) %>%
  ungroup() %>%
  ggplot() + geom_col(mapping = aes(x=stemmed, y=tf_idf, fill=main_category),
                      show.legend = FALSE) +
  ggtitle("Bigrams in Sucessful Campaigns by Category (TF-IDF)") +
  labs(x = NULL, y = "n") + facet_wrap(~main_category, scales='free') + coord_flip()
```

(22)

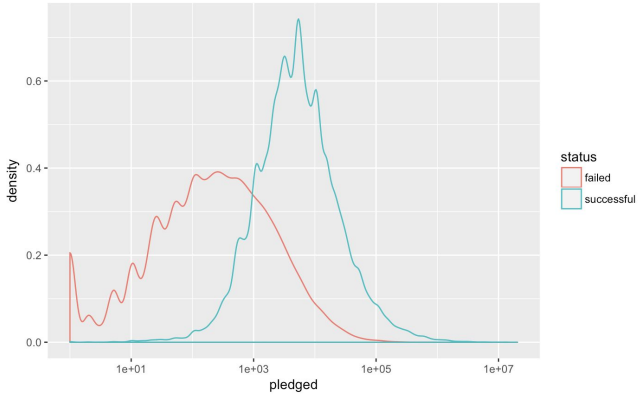
```
#cross-validation module.

cross_validation <- function(data, formula, k) {
  cv <- crossv_kfold(data, k)
  cv <- cv %>%
    mutate(fit = map(train, ~ glm(formula, family = binomial(link='logit') ,data = .)))

  cv <- cv %>% mutate (score  = map2(cv$test, cv$fit, ~ predict( .y, newdata = .x, type="response"))) %>%
    mutate(score = map(score, function(x) ifelse(x > 0.4, 1,0)))  %>%
    mutate(score = map2_dbl(cv$test, score, ~ mean(data.frame(.x)$status == .y)))
  return(cv)
}

cv <- cross_validation(model_data, status ~ main_category + backers_count + country  +
```

pledged + Month + Year + days_live, 8)



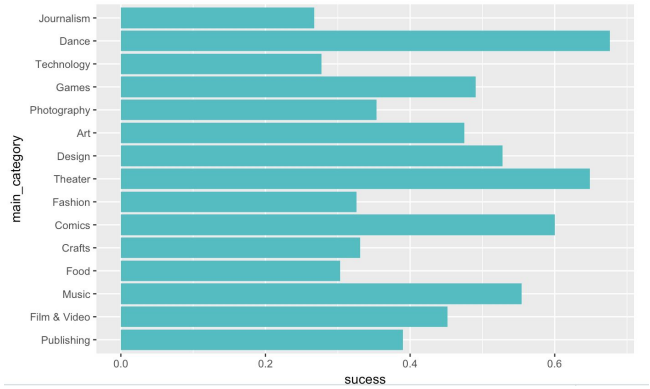
#density plot of backers
ggplot(model_data, aes(x=backers_count, color=status)) + geom_density() + scale_x_log10()

(23)



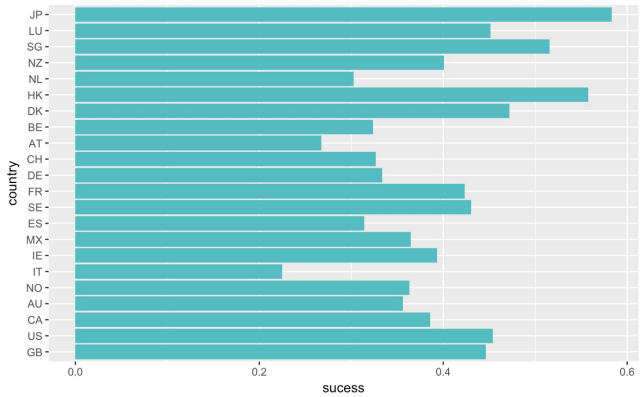
#density plot of pledged
ggplot(model_data, aes(x=pledged, color=status)) + geom_density() + scale_x_log10()

(24)



#success ratio per main_category
model_data %>% group_by(main_category) %>% summarise(sucess = mean(status=='successful')) %>%
  ggplot () + geom_bar(aes(main_category, sucess), fill="#00BFC4", stat="identity") + coord_flip()

(25)




```
#success ratio per country
model_data %>% group_by(country) %>% summarise(sucess = mean(status=='successful')) %>%
  ggplot () + geom_bar(aes(country, sucess), fill="#00BFC4", stat="identity") + coord_flip()
```

(26) Linear regression code

```
# Subset the modeling data further to train and test a model
model_data <- as_tibble(full_data$modeling)
model_data_cv <- crossv_kfold(model_data, 10)

# Template code for cv modeling linear regression
model_data_cv <- model_data_cv %>%
  mutate(fit = map(train,
                   ~ lm(perc_funded ~ main_category + backers_count + country  + pledged + Month + Year + days_live,
                   data = .)))

model_data_cv <- model_data_cv %>%
  mutate(rmse_train = map2_dbl(fit, train, ~ rmse(.x, .y)),
         rmse_test = map2_dbl(fit, test, ~ rmse(.x, .y)))

mean(model_data_cv$rmse_train)
mean(model_data_cv$rmse_test)
```