SMI610: Social Analytics & Visualisation

Can content and video characteristics be used to predict the success of online educational videos? – An exploratory analysis of TED Talk data

Student Number: 180283033

Word Count: 3823 Words

Executive Summary

Despite educational content becoming increasingly accessible online, what determines the long-term success of online content is not well understood. Organisations such as 'Technology, Entertainment, Design' provide freely available videos which aim to be both informative and entertaining. Exploring which factors may contribute to views provides three related but separate insights. Firstly, educational content providers would be able to make their content more desirable. Secondly, insight about social issues, interests, values, and trends can be gained. Thirdly, existing studies focus on early viewing trends to predict the long-term success of online content, leaving the extent to which factors other than early viewing trends can predict long-term success of online content unclear. The current report, therefore, investigates if views on TED talks can be predicted using meta-variables related to the video, such as duration, title length, and number of translations, as well as viewer reactions and content related groupings.

It is found that a random forest regression is unable to accurately capture relationships between views and other variables. The extent to which the poor model fit of the random forest regression is methodologically or contextually based is discussed. Despite being unable to predict views accurately, some interesting patterns emerge in the current analysis. For example, tag keywords are found to be useful in clustering talks by themes and appear to have a strong overlap with talk descriptions. Moreover, viewer reactions seem to differ by talk themes, such that talks about the arts receive more 'beautiful' and 'inspiring' ratings than talks about different topics. Thus, although the current analysis is unable to build a model that can accurately predict views, some trends in the data emerge in the exploratory analysis. Nonetheless, the question of how views can accurately be predicted without using early viewing trends remains.

Introduction

Literature Review

Educational and academic materials are increasingly available online. One example of such materials are the talks hosted by ‘Technology, Entertainment, Design’ (TED), an organisation aiming to provide talks which are both educational and entertaining. These talks have a wide reach beyond the academic community and are freely available online. While the online availability of this content allows for a new way of sharing knowledge, little is understood about what type of knowledge or content people seek. Thus, understanding what determines long-term success of online content can improve design academic online materials. Moreover, the type of content people seek can provide insight into social trends, values, and behaviours.

Existing research identifies patterns and biases linked to ratings and comments. For example, Sugimoto and Thelwall (2013) argue that talks on science and technology held by non-academics are less liked than those presented by academics. In addition, gender biases can be observed within audience reactions, such that comments on female speakers’ talks are more likely to be emotional than those left on male speakers’ talks (Tsou et al., 2014). These findings provide insight not only into socio-cultural perceptions of gender and status, but also into how online educational materials are received.

Despite this attention to comments, no study, to date, addresses differences in views. An understanding of which talks receive views can, firstly, contribute to improving the impact of educational materials, and secondly – and more importantly – provide insight into trends, interests, and how people engage with educational content online. However, as with other online content, predicting how much attention these materials receive is difficult. Using early views, Szabo and Huberman (2010) are able to predict long term successes of content on the online platforms Digg and YouTube (see also Pinto et al., 2013). Expanding on this research and using TED talks as an example, the current analysis aims to expand on this research to assess if views can also be predicted prior to access to early view data. In addition to looking at video-related characteristics, this analysis explores ways in which talk content and reactions to talks may be incorporated into a predictive model of views. This could help in planning for educational content to reach wider audiences, as well as highlight which variables and content itself impact what type of online content people seek.

The current report, therefore, investigates if views on TED talks can be predicted using variables such as talk duration and number of translations. First, the dataset and method are outlined. Thereafter, findings are reported. This section is split into various parts; firstly, existing variables are explored and pre-processed. Secondly, viewer reactions are explored and grouped to investigate if the relationship between views and audience reaction. Thirdly, textual and sentiment analyses are used to group talks by topic and assess potential impacts of sentiment on views. Moreover, differences between topic groups are explored. Fourthly, a random forest regression model is used to assess whether video characteristics and talk content can predict views. Fifth, findings and limitations are discussed.

Aims and Objectives

This research aims to bridge some of the aforementioned gaps and analyse viewing behaviour of TED talks. Secondly, the current analysis explores different characteristics of talks and videos to address if these result in differences in success or views. Moreover, the current analysis aims to explore textual aspects of TED talks, including transcripts and descriptions, and investigate links with views.

Methods

Data Collection

The dataset used for the current analysis can be downloaded via Kaggle (https://www.kaggle.com/; Banik, 2017), and is publicly available with the creation of a Kaggle-username. The dataset contains data on TED talks uploaded to the TED website (https://www.ted.com/) before September 21st, 2017. All information included in the current dataset is openly available on the TED website.

Variables in the dataset include number of views, full transcripts, dates of filming and publishing, number of translations, ratings, related talks, and other meta-information about the talks. A full list and description of variables available and whether they are used in the current analysis can be found in Table 1. While comments include only first-level comments made on the TED website, it is unclear whether the number of views includes only views made directly on the TED website or also those views made on external platforms, including on YouTube. Most variables are recorded for all 2,550 talks in the dataset, however, transcripts are only available for 2,464 of these talks.

Table 1. Variables included in the TED dataset and current analysis.

Variable	Variable Description	Included in Analysis
Comments	*Number of first level comments on TED website*	Yes
Description	*Short summary of talk*	Yes
Duration	*Length of talk in seconds*	Yes
Event	*TED / TEDx event*	No
Film Date	*Date of talk*	Yes
Languages	*Number of languages, into which talk is translated*	Yes
Main Speaker	*Name of the main speaker*	No
Name	*Official name of talk, including name of speaker and title of the talk*	No
Number of Speakers	*Number of speakers*	Yes
Published Date	*Date on which the talk is published on the TED website*	Yes
Ratings	*Word ratings and frequencies given to talks**	Yes
Related Talks	*List of talks recommended to watch next by the TED website*	No
Speaker's Occupation	*Main occupation of main speaker*	No
Tags	*Themes of talk in keywords*	Yes
Title	*Title of talk*	Yes
Transcript	*Full transcript of talk in English*	Yes
URL	*URL of talk*	No
Views	*Number of views*	Yes

** Ratings include funny, beautiful, ingenious, courageous, longwinded, confusing, informative, fascinating, unconvincing, persuasive, jaw-dropping, ok, obnoxious, and inspiring.*

Data Analysis

The analysis is divided into an exploratory analysis of the data and a prediction of number of views. The first part of the analysis investigates patterns in the data and explore common themes and topics across the talks. The second part of the analysis uses these findings to attempt to predict number of views. The full analysis is done in R and can be found in Appendix A.

Findings

Variable Exploration

First, some variables are pre-processed and added to the dataset. To standardise comments, comments per views are calculated. Moreover, title lengths, in characters, and length of talk descriptions, in words, are calculated. Thereafter, distributions are explored. Views and comments are found to be positively skewed, though log-transformations (base e) normalise the distributions. Moreover, although the oldest talk in the dataset is filmed in the year 1972, talks are only published on the website after 2006. In addition, there is a steep increase in number of talks filmed from 1972 until 2015, and then a slow decrease. All these findings are visualised in Figure 1.

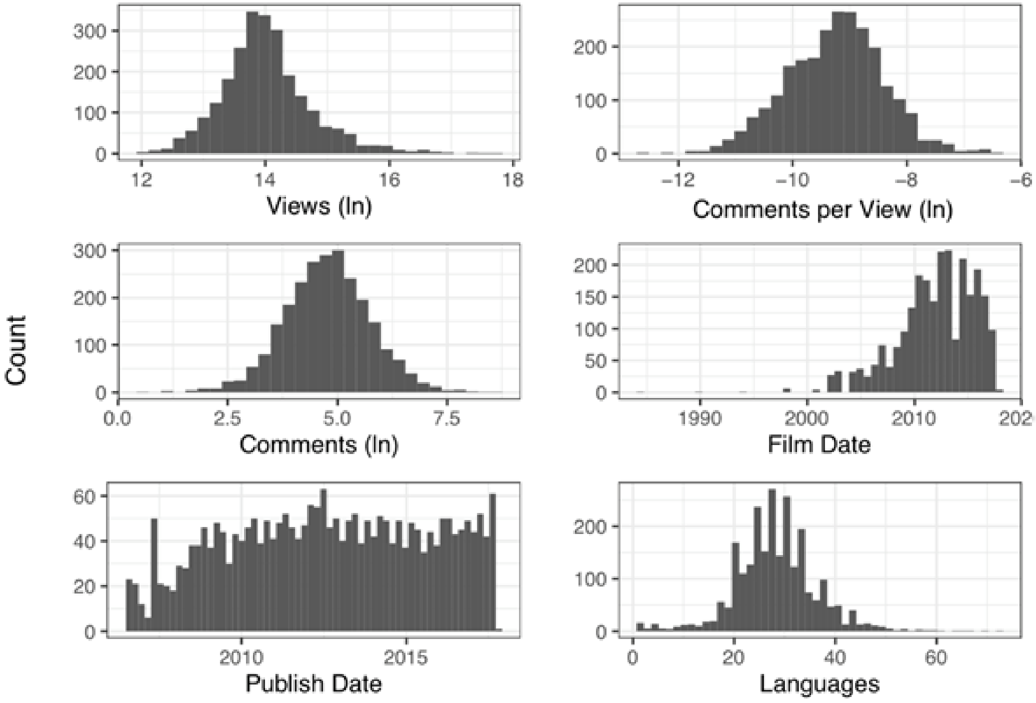


Figure 1. Variable distributions

Secondly, relationships between views and other continuous variables, including comments, comments per view, description length, duration, title length, languages, and number of speakers are explored (see Figure 2). Relationships appear weak, although there may be a linear relationship between log-transformed views and the number of languages, in which a talk is available. Moreover, there may be a weak negative linear relationship between views and comments per view. Lastly, only few talks appear to have more than one speaker, making it difficult to assess the relationship between number of speakers and views. Duration, description length, and title length seem to have no linear relationship with views.

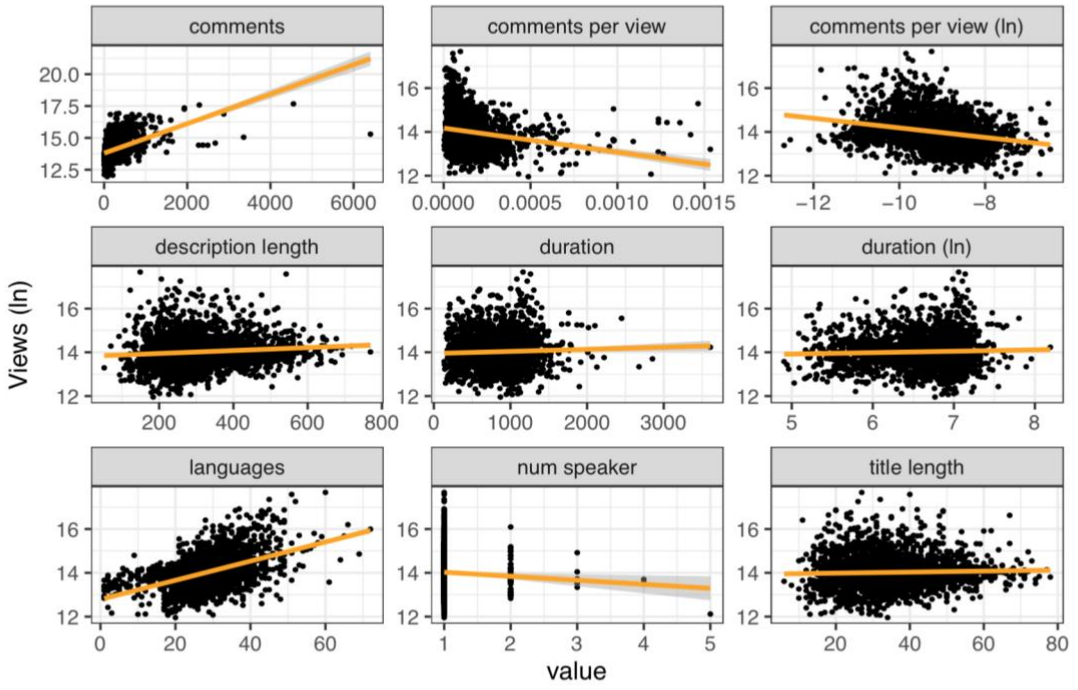


Figure 2. Relationships between views and other continuous variables.
** *Note: Orange line shows a linear model fit.*

Talk Ratings

Ratings of the talks are also investigated, to assess how these differ between talks and if they are linked with views. Ratings are converted into percentages, such that a value shows what percentage of all ratings a talk received are in a certain category. As shown in Figure 3, positive ratings, including inspiring, informative and fascinating more frequently have high percentages than negative ratings, such as longwinded, obnoxious, and confusing.

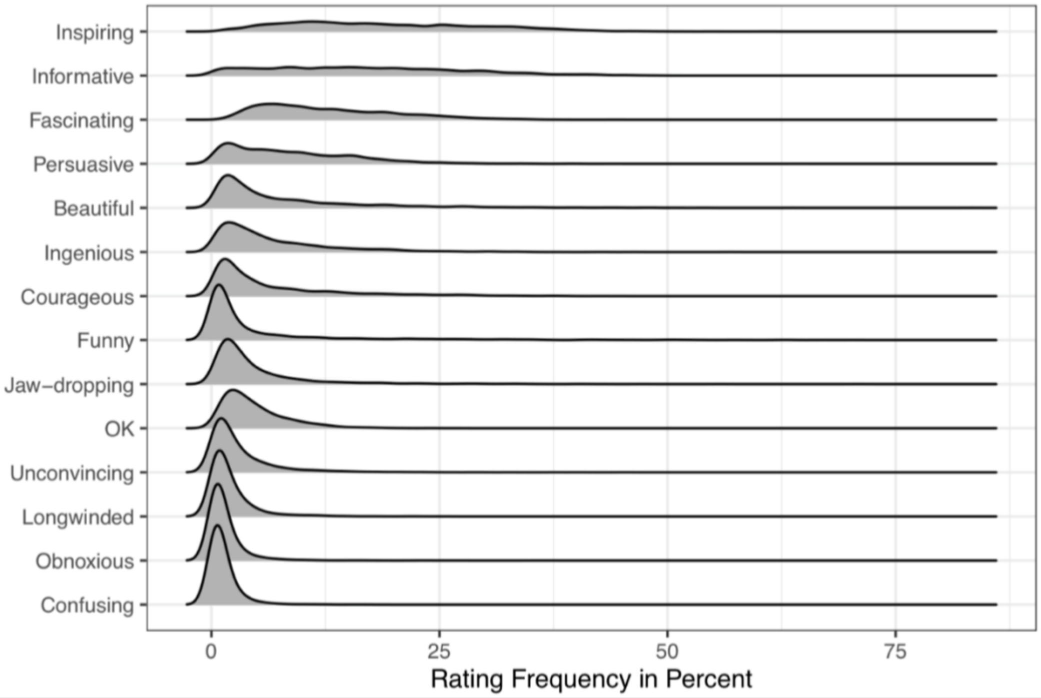


Figure 3. Distributions of ratings.

Negative ratings being similarly less frequent suggests that ratings may be clustered. For this, ratings are split into principle components. As ratings are mostly uncorrelated or only weakly correlated with one another this type of analysis is possible (James et al., 2017; see Appendix B). Using a k-means clustering algorithm, the data are clustered on the raw data and on the top three principle components, which together explain over 80% of variance in the data. The resulting clusters are visualised in Figure 4. Notably, clusters do not differ between the PCA-based k-means and the raw data k-means approaches, suggesting that variance in the data due to components not included in the PCA-based analysis has little impact. Moreover, these findings show that negative ratings (confusing, longwinded, ok, obnoxious, unconvincing) are indeed clustered, suggesting that talks receive more similar scores across these ratings than other ratings.

Relationships between rating characteristic and these characteristic clusters with views are further explored visually in **Error! Reference source not found.**. For this, each talk is labelled with one characteristic only; the characteristic a talk most commonly received in its ratings. As can be seen, talks receiving mostly negative ratings are associated with fewer views than talks receiving other ratings. All talks in the negative characteristic cluster have the lowest mean ratings. There appears to be no trend between a talk's views and having the most common rating in a different characteristic cluster. Thus, having mainly negative ratings may be a predictor of views.

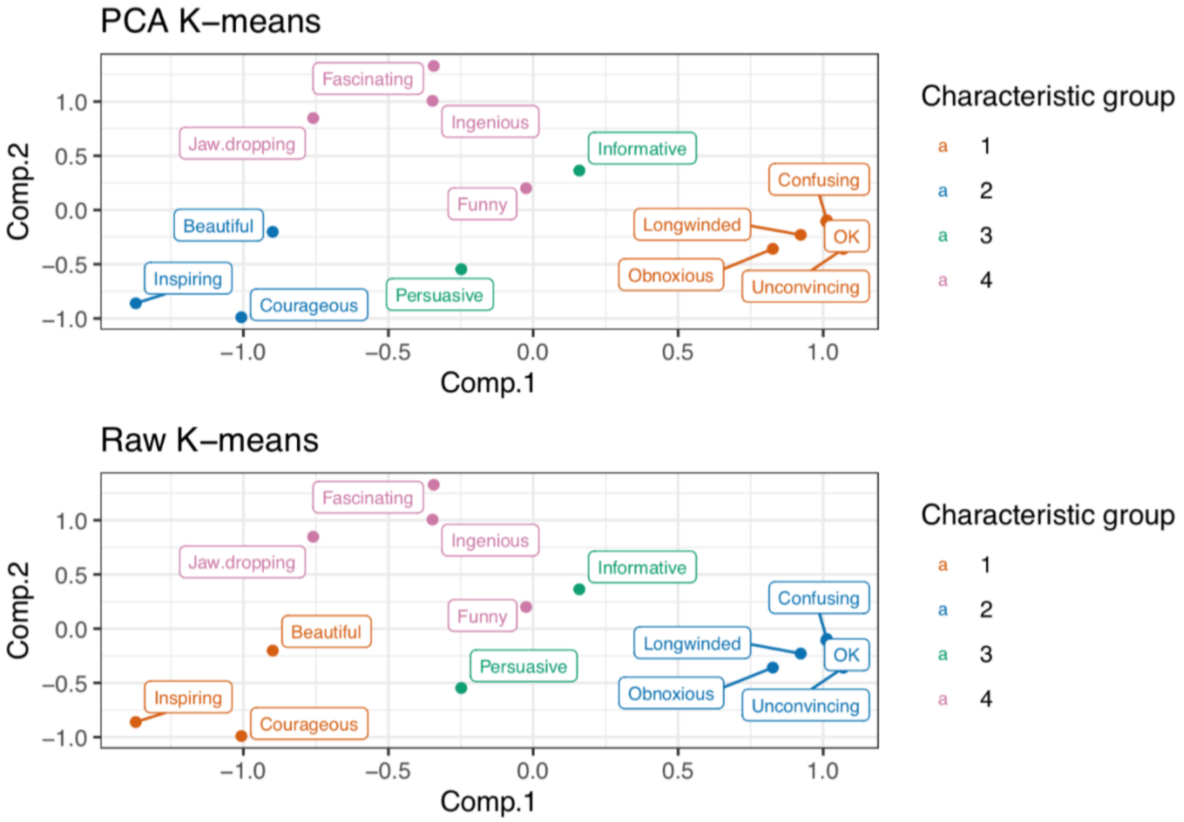


Figure 4. Rating clusters based on (top) the top three components of a principle components analysis (PCA), and (bottom) the raw ratings data.
*** Note: Axes only represent top two principle components, while clustering is done on the top three components.*

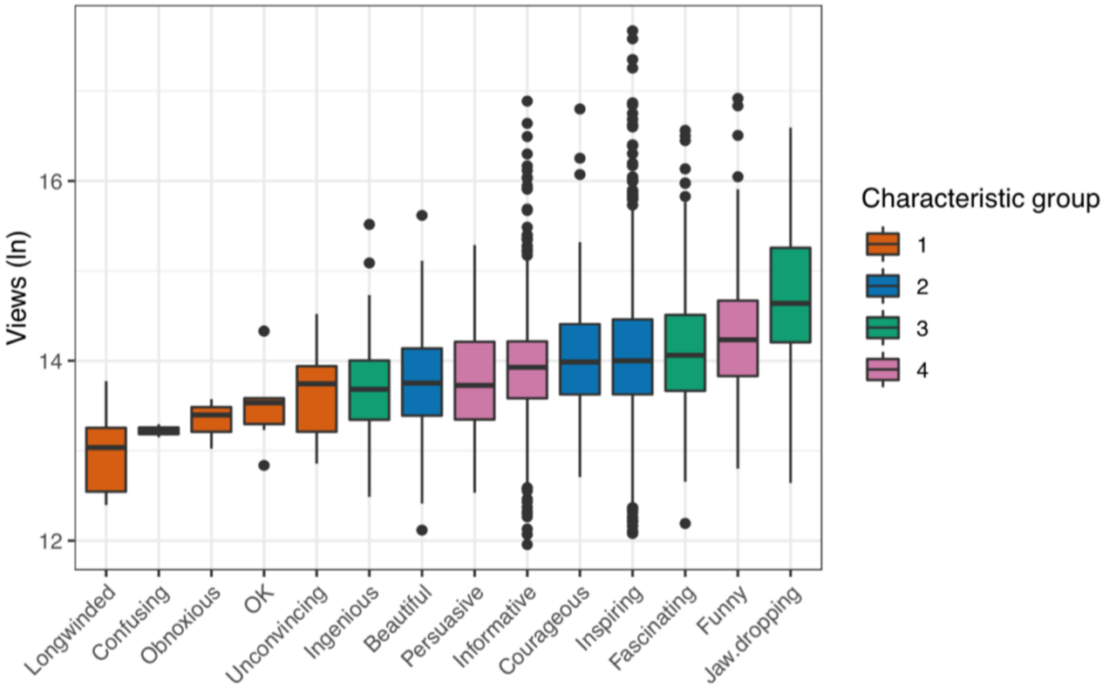


Figure 5. Views and ratings by characteristic groups.
*** Note: Talks are grouped into the characteristic, which they most frequently received.*

Topic Clustering

The second part of the analysis is concerned with clustering the talks by topics and themes, to assess if certain topics are more popular than others; or, in other words, receive more views. As talks are already tagged according to broader themes, and to avoid clustering on common words not included in R's stopword dictionary (tm package), clusters are based on the tags variable. The most common tags across all talks are technology and science, with both linked to over 600 talks (see Figure 6).

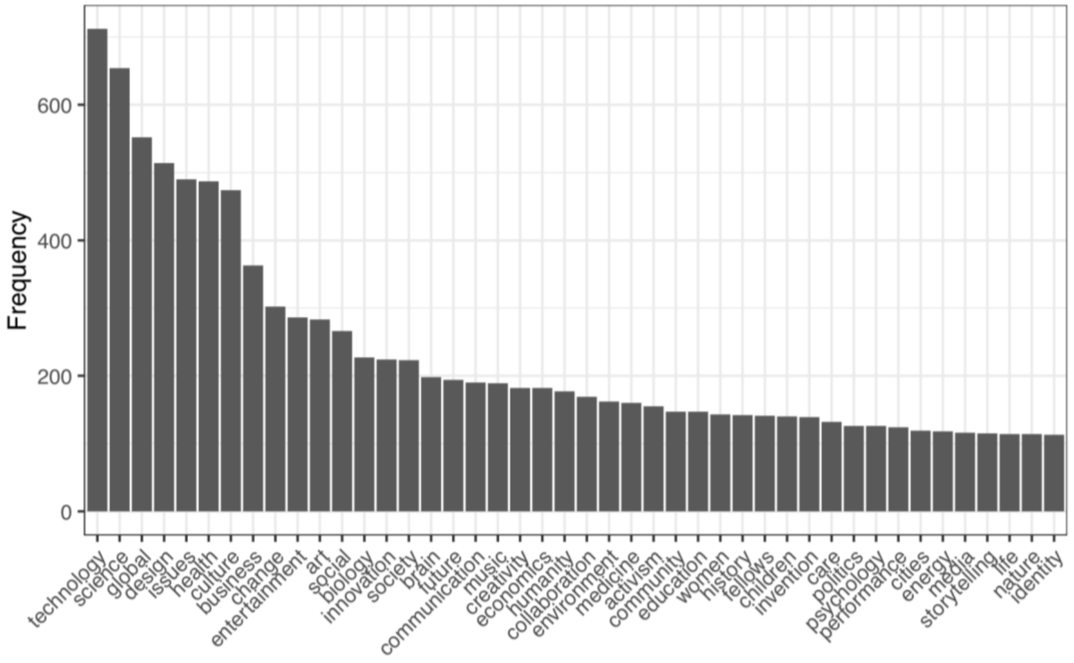


Figure 6. Top tags across all talks.

A hierarchical k-means analysis, using 6 cuts, on talk tags suggests that topics may be distinguishable by clustering tags, as the most frequent tags differ strongly between some clusters. For instance, based on the 10 most frequent tags per cluster (see Figure 7) tag cluster 2 appears to contain talks linked to the arts, whereas clusters 3 and 5 seem to be political, cluster 6 about health and medicine, and 4 about technology and innovation. While this proposes that themes emerge through clustering, there is also some overlap between some clusters. For example, tag clusters 1 and 4 both have the tags ‘technology’, ‘science’, and ‘design’ among their top 10 tags. Moreover, cluster 1 contains considerably more talks than other clusters and its top tags cannot easily be categorised by a single theme. Therefore, cluster 1 was split up, using the same method, into two further clusters (see Figure 8).

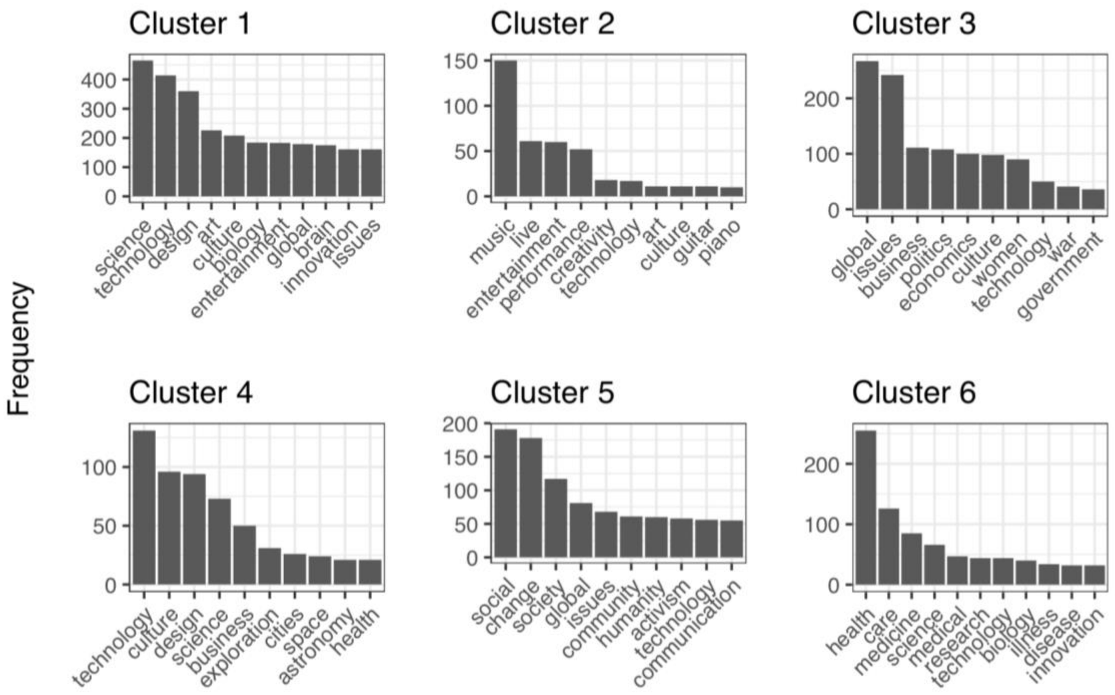


Figure 7. Top 10 tags by tag cluster.

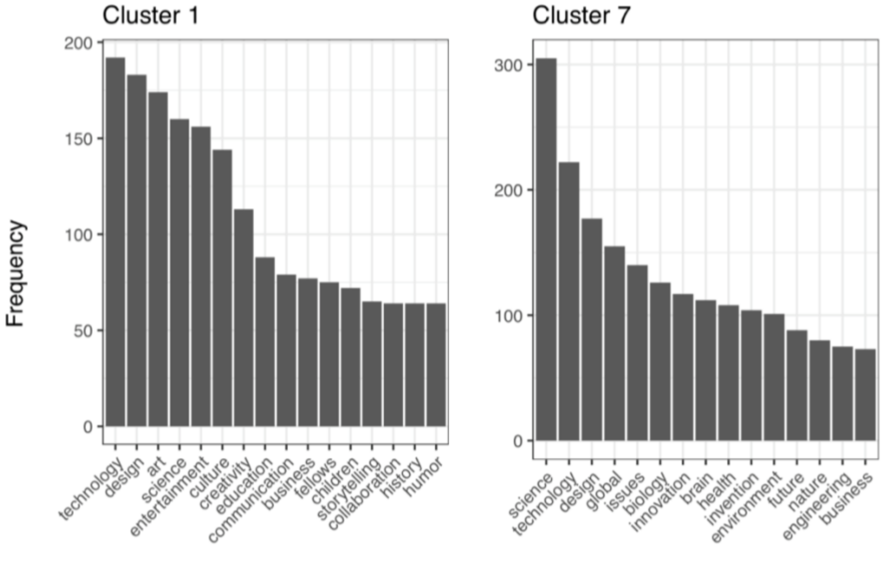


Figure 8. Top 15 tags by cluster – Figure 7, cluster 1 split into 2 clusters.

Both new clusters are more comparable in most frequent tag to the other clusters. Additionally they seem more focussed; while one cluster (Figure 8, cluster 1) contains a high frequency of talks tagged with entertainment and business related keywords, the other one (Figure 8, cluster 7) appears to contain talks related to the natural sciences and medicine.

To validate cluster topics, term frequency–inverse document frequencies (tf–idf) are calculated for talk descriptions. These rate words based on their frequency of occurrence in one document and across documents (Silge and Robinson, 2019). As a result, words appearing frequently in one document only will receive a high rating, while those appearing both infrequently, or frequently but across all documents receive low ratings. As shown in Figure 9, top words for cluster 2 centre around music, cluster 3 around politics, cluster 4 around physics and astronomy, and cluster 6 around health and medicine. Top words for clusters 1, 5, and 7 are more scattered or include many names. Although there is uncertainty around some clusters, this method is able to confirm some themes which emerged from the k-means analysis.

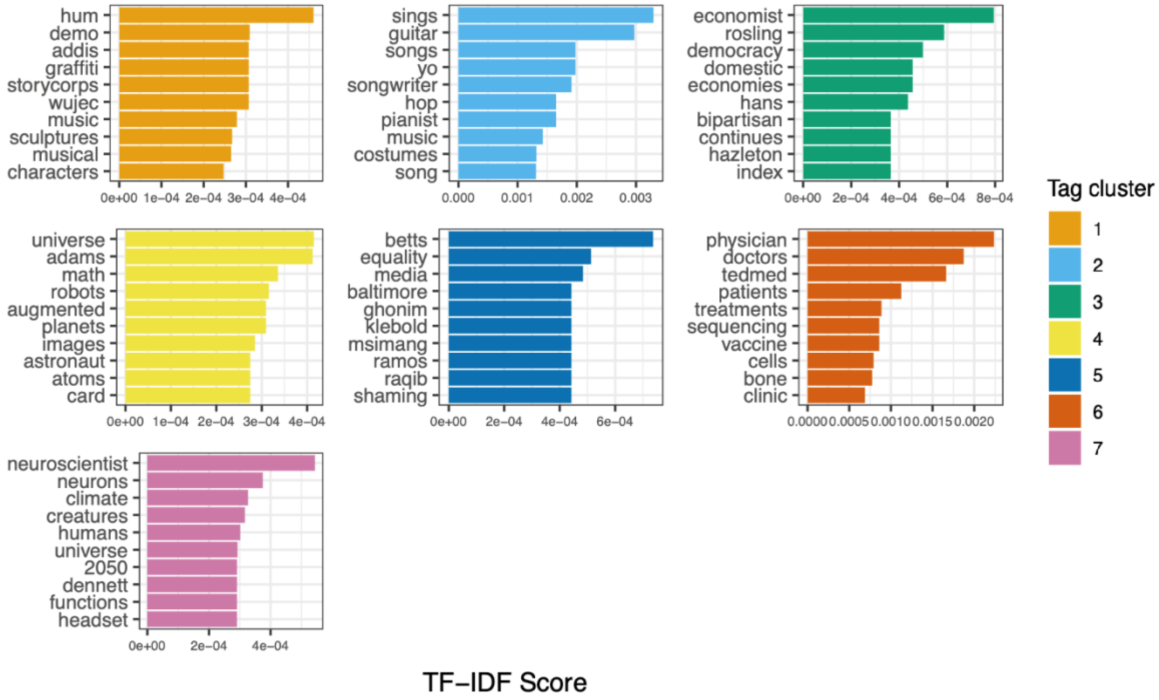


Figure 9. Top 10 words per cluster by tf–idf score.

To assess if topics are linked to other variables, the difference in views, ratings, languages, comments per view, duration and film and publish dates between these tag clusters is analysed. Tag cluster 2 appears to have fewer ratings in the group ‘funny, informative, persuasive’, and more ‘beautiful, inspiring, fascinating’. Moreover, there is a steep increase in the number of cluster 5 talks filmed and published after 2015 (see Appendix C), suggesting that this topic is trending. Finally, while talks in the second cluster more frequently have shorter durations, differences in views, comments per views, and languages are not apparent between the clusters (see Figure 10).

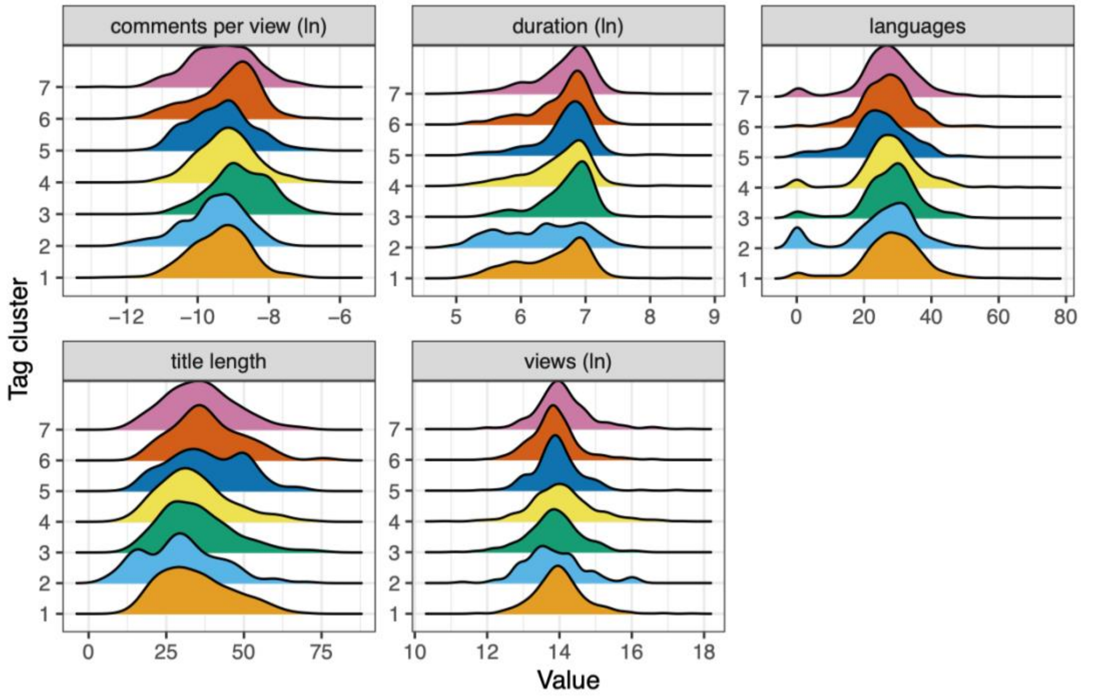


Figure 10. Continuous variables by cluster.

Sentiment Analysis

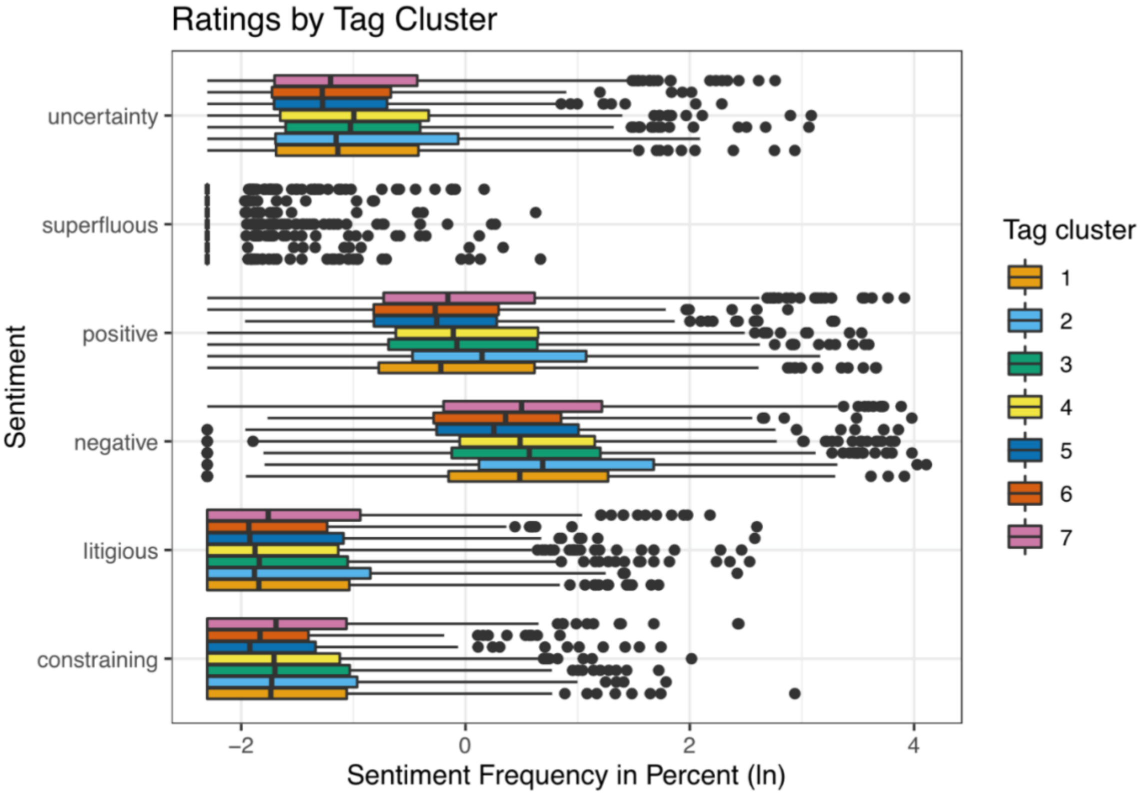


Figure 11. Sentiments by cluster.

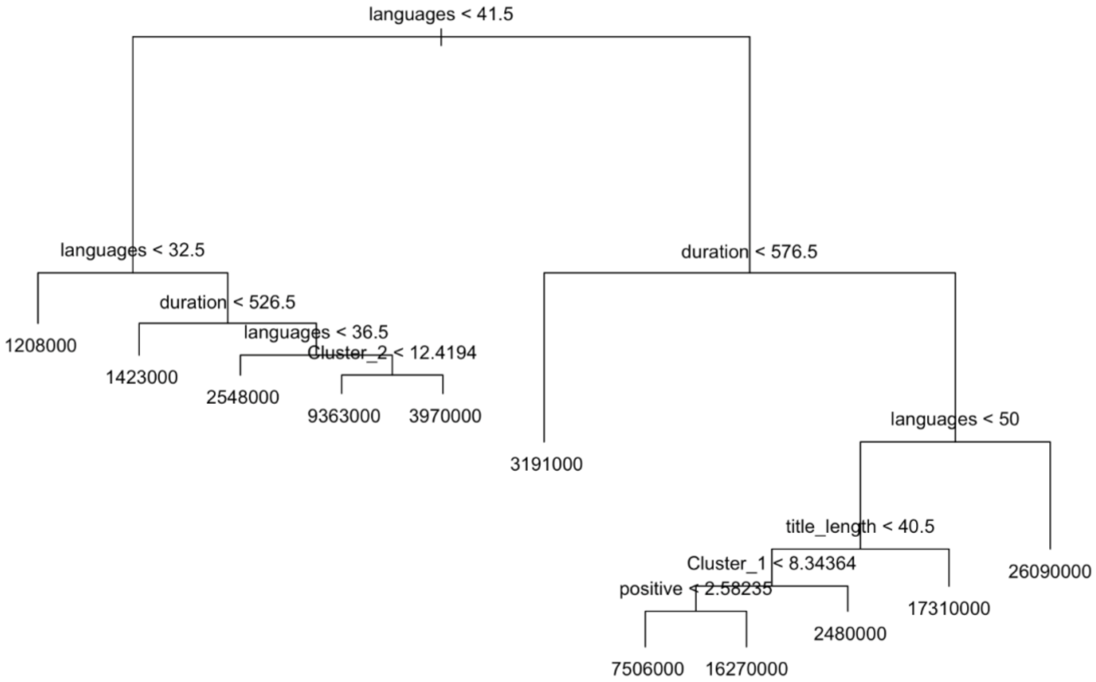
The second part of the textual analysis investigates the sentiment of the transcripts. This can help identify if views vary by amount or type of sentiment. Using the loughran dictionary for sentiment in R, talks are explored for sentiment frequency. As displayed in and Table 2, negative and positive sentiment are more frequent than other sentiments. However, differences between tag clusters do not emerge.

Table 2. Mean percentages of words with specific sentiments by cluster.

Cluster	Constraining	Litigious	Negative	Positive	Superfluous	Uncertainty
1	0.28	0.30	3.23	2.06	0.02	0.62
2	0.36	0.42	4.74	2.95	0.03	0.87
3	0.26	0.36	3.76	2.10	0.01	0.71
4	0.21	0.28	3.18	1.79	0.02	0.63
5	0.23	0.29	3.08	1.59	0.02	0.49
6	0.17	0.25	2.48	1.24	0.01	0.43
7	0.30	0.36	3.68	2.40	0.02	0.72

Predicting Views

As no linear relationship between views and other variables is apparent – with the exception of languages – , views are predicted using a random forest regression, rather than a linear model (James et al., 2017). Due to comments per view being modelled off views, this variable is excluded from this part of the analysis. A single regression tree is run on the entire dataset to visualise some of the splits made in the data (see Figure 12). Predictor variables include year published, languages, the various characteristic groups, tag clusters, negative and positive sentiment, duration, and title and description lengths. In the decision tree visualised below only languages, characteristic groups 1 and 2, duration, title length, as well as positive sentiment are selected by the algorithm to predict



views. The tree only has 9 terminal nodes, indicating that only 11 different values are predicted for number of views.

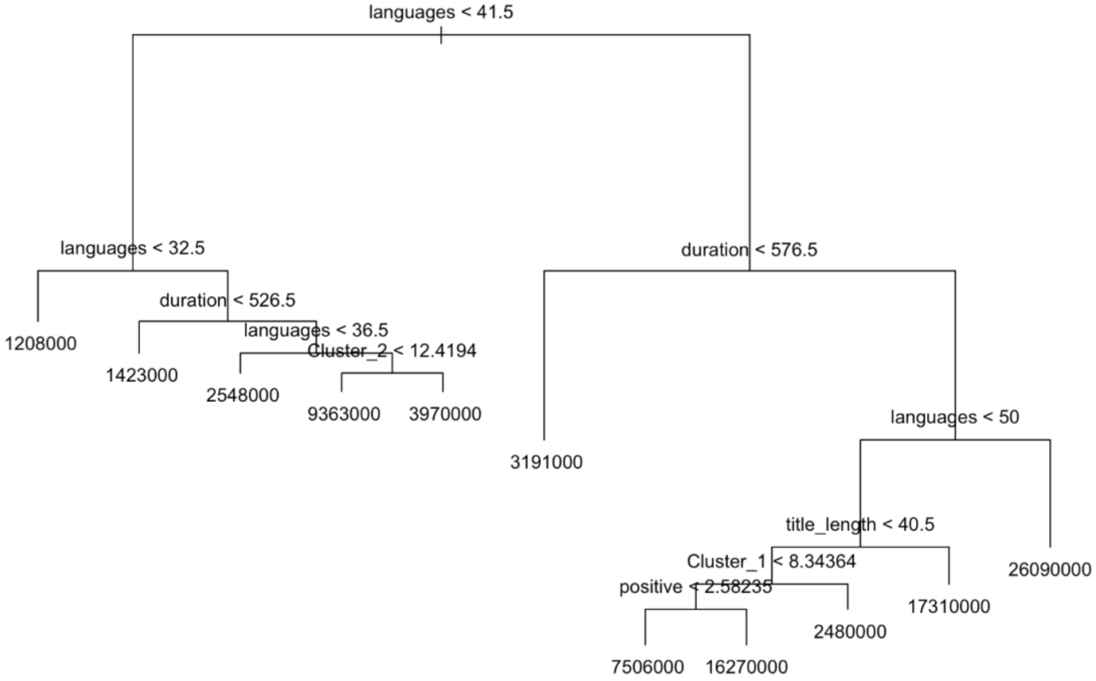


Figure 12. Single regression tree to predict views.
*** Note: Cluster refers to characteristic groups (see Figure 4).*

As trees typically have low predictive accuracy (James et al., 2017), a random forest regression is run using the same variables to improve robustness. This is run a randomly sampled training set (80% of dataset) and tested on the subset not used to train the data. The most important variables to predict views in the random forest regression are languages, duration, characteristic group 1 (percentage of negative ratings), and description length. However, test predictions are found to have a mean absolute percentage error of 38% compared to actual views, suggesting a poor model fit.

Discussion and Conclusions

The current analysis is unable to predict views using the variables investigated. While TED talks with higher percentages of negative ratings appear to have fewer views, topics, number of languages in which a talk is available, sentiment, talk duration, title and description lengths, and year published seem to have little impact on views. The current analysis showcases that a random forest regression is likely the wrong approach to predicting TED talk views. This may be due to the variables tested having little impact on views, or the model being unable to capture the relationship between the variables. Future research could build on this by comparing various models to assess which one best captures trends in the data. Another approach may be to split the data by views and investigate differences in views in a more top-down manner.

Despite being unable to predict views, some patterns emerge in the current analysis. For instance, the current analysis finds that tag keywords can be useful in clustering talks by themes, as they strongly overlap with talk descriptions. This makes clustering topics easier, as it minimises pre-processing of text data. In addition, ratings are found to vary with topics, such that talks about the arts receive more ratings in the ‘beautiful’, ‘courageous’, ‘inspiring’ group than other talks. Moreover, these talks are more frequently shorter than talks about other topics. Finally, talks are more frequently rated positively than negatively.

Although most topics have no steep changes in number of talks filmed and published over time, post-2015 sees a steep increase in talks about social issues and change published, highlighting how trends change over time. Consequently, a longitudinal analysis of views by topic may provide valuable insight into cultural values, issues, and trends over time. Views are likely also impacted by such trends; hence, predictions of views may be improved by a longitudinal analysis. Data for this may be available via YouTube (see Cheng et al., 2008).

Analysing the data available on YouTube, as opposed to from the website may also solve limitations surrounding variable measurement. It is unclear how views in the current dataset are calculated. While comments refer strictly to first level comments made directly on the TED website, it is not clear whether the number of views refers to the number of times a talk was watched on the TED website, or if this variable includes views on other platforms, such as YouTube. This limits the current analysis, as other variables, such as number of languages in which a talk is available, may vary by platform. Using only YouTube views and other statistics from YouTube, rather than the TED website, may solve this issue as it ensures that the same information was available to all viewers.

Furthermore, the inclusion of some of the variables excluded in the current analysis may improve view prediction. First, the finding that there is some variance based on theme proposes that splitting the data into less broad topics may highlight further differences between talks. The ‘related talks’ variable, could allow for a more disaggregated clustering through a social network analysis and thereby build on the current analysis. Secondly, previous research suggests that talks by academic, particularly those about science and technology, are rated more favourably than those presented by non-academics (Sugimoto and Thelwall, 2013). Thus, including the speakers’ occupation in the analysis may improve the current analysis.

Lastly, the current research is limited by the unavailability of geographical data. Brodersen et al. (2012) find that 50% of videos analysed receive over 70% of their views from a single geographical region. While the occurrence of this may be reduced by TED being an international organisation and many videos being available in multiple languages, regional trends cannot be controlled for and may skew the current data.

In conclusion, therefore, although the exploratory analysis highlights certain trends in the data, including lower views for more negatively rated talks, the current analysis is unable to build a model that can accurately predict views. Having content available in various languages appears to increase the number of views and may, therefore, be an approach to make online content more widely accessed. Inferences about what type of content viewers seek cannot be made, although the data indicates that trends may change over time.

References

Banik, R. 2017. TED Talks: Data about TED talks on the TED.com website until September 21st, 2017. *kaggle*.

Brodersen, A., Scellato, S. and Wattenhofer, M. 2012. YouTube around the world: Geographic popularity of videos. *World Wide Web Conference*., p.241.

Cheng, X., Dale, C. and Liu, J. 2008. Statistics and social network of YouTube videos. *IEEE International Workshop on Quality of Service, IWQoS*., pp.229–238.

James, G., Witten, D., Hastie, T. and Tibshirani, R. 2017. *An Introduction to Statistical Learning*. London: Springer.

Pinto, H., Almeida, J.M. and Gonçalves, M.A. 2013. Using Early View Patterns to Predict the Popularity of YouTube Videos. , pp.365–374.

Silge, J. and Robinson, D. 2019. Text Mining with R: A Tidy Approach. *O'Reilly*. [Online]. [Accessed 20 July 2019]. Available from: https://www.tidytextmining.com/tfidf.html.

Sugimoto, C.R. and Thelwall, M. 2013. Scholars on Soap Boxes: Science Communication and Dissemination in TED Videos. *Journal of the American Society for Information Science and Technology*. **64**(4), pp.663–674.

Szabo, G. and Huberman, B.A. 2010. Predicting the popularity of online content. *Communications of the ACM*. **53**(8), p.80.

Tsou, A., Thelwall, M., Mongeon, P. and Sugimoto, C.R. 2014. A community of curious souls: An analysis of commenting behavior on TED Talks videos. *PLoS ONE*. **9**(4).

Appendix A. R code

```
library(tidyverse)
library(lubridate)
library(rjson)
library(ggridges)
library(tm)
library(proxy)
library(fpc)
library(cluster)
library(tidytext)
library(GGally)
library(reshape2)
library(viridis)
library(ggfortify)
library(gridExtra)
library(ggrepel)
library(pander)
library(grid)
library(tree)
library(randomForest)
```

Section 1.1

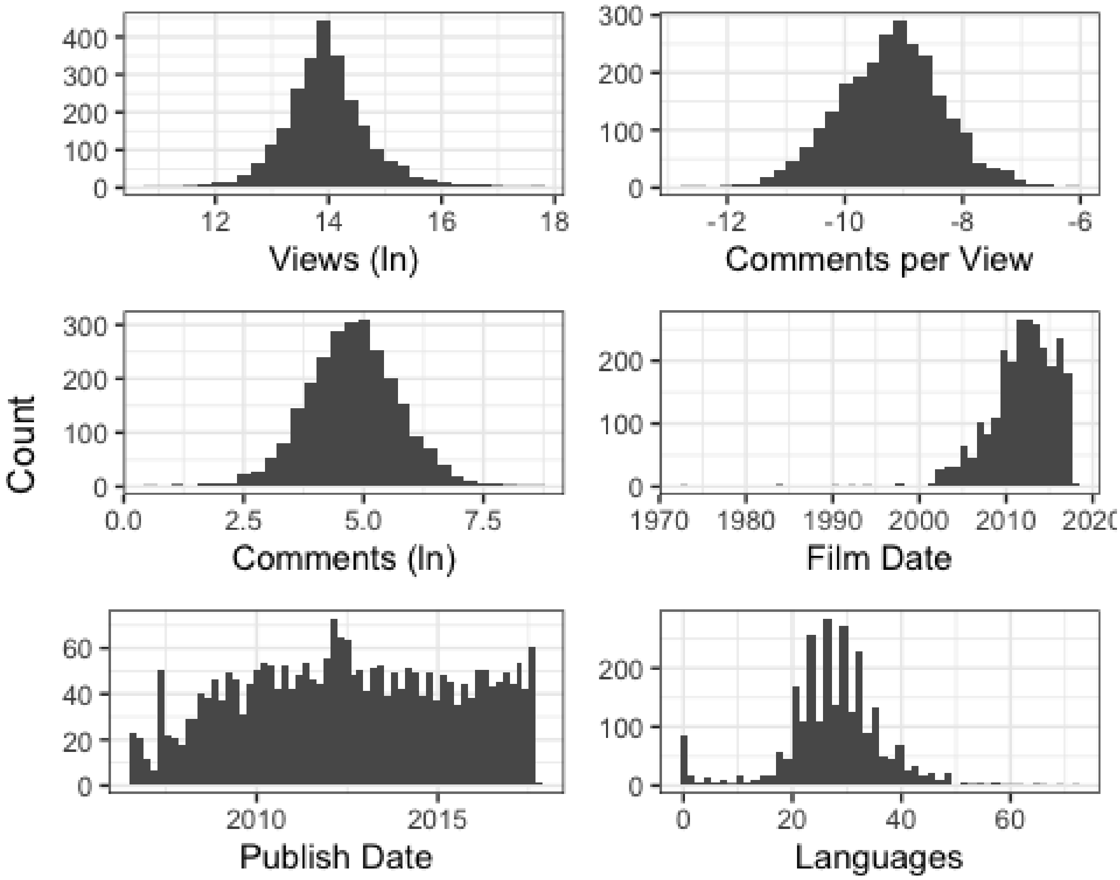
load and clean data

```
# load ted data
ted_data <- read.csv('Data/ted_main.csv', sep=',', header=TRUE, stringsAsFactors = FALSE)

# explore ted_data variables
names(ted_data)

##  [1] "comments"           "description"        "duration"
##  [4] "event"              "film_date"          "languages"
##  [7] "main_speaker"       "name"               "num_speaker"
## [10] "published_date"     "ratings"            "related_talks"
## [13] "speaker_occupation" "tags"               "title"
## [16] "url"                "views"

# clean ted_data
ted_data <- ted_data %>%
  mutate(tags = tolower(tags), # lower case string variables for later analysis
         title = tolower(title),
         film_date = as_datetime(film_date),
         published_date = as_datetime(published_date), # convert dates from UNIX to datetim
e format
         published_year = as.character(substring(published_date, 1, 4)),
         filmed_year = as.character(substring(film_date, 1, 4)),
         talk_id = c(1:as.integer(count(ted_data))), # add ID to each talk
         title_length = nchar(title), # add string length of title
         description_length = nchar(description), # add string length of talk description
         comments_per_view = comments/views) # add comments per view

# load and clean transcript data
ted_transcripts <- read.csv('Data/transcripts.csv', sep=',', header=TRUE, stringsAsFactors
= FALSE) %>%
  mutate(transcript = tolower(transcript)) %>% # change all to lower case
  left_join(select(ted_data, talk_id, url), by='url') # add same ID to each talk as in ted_
data
```

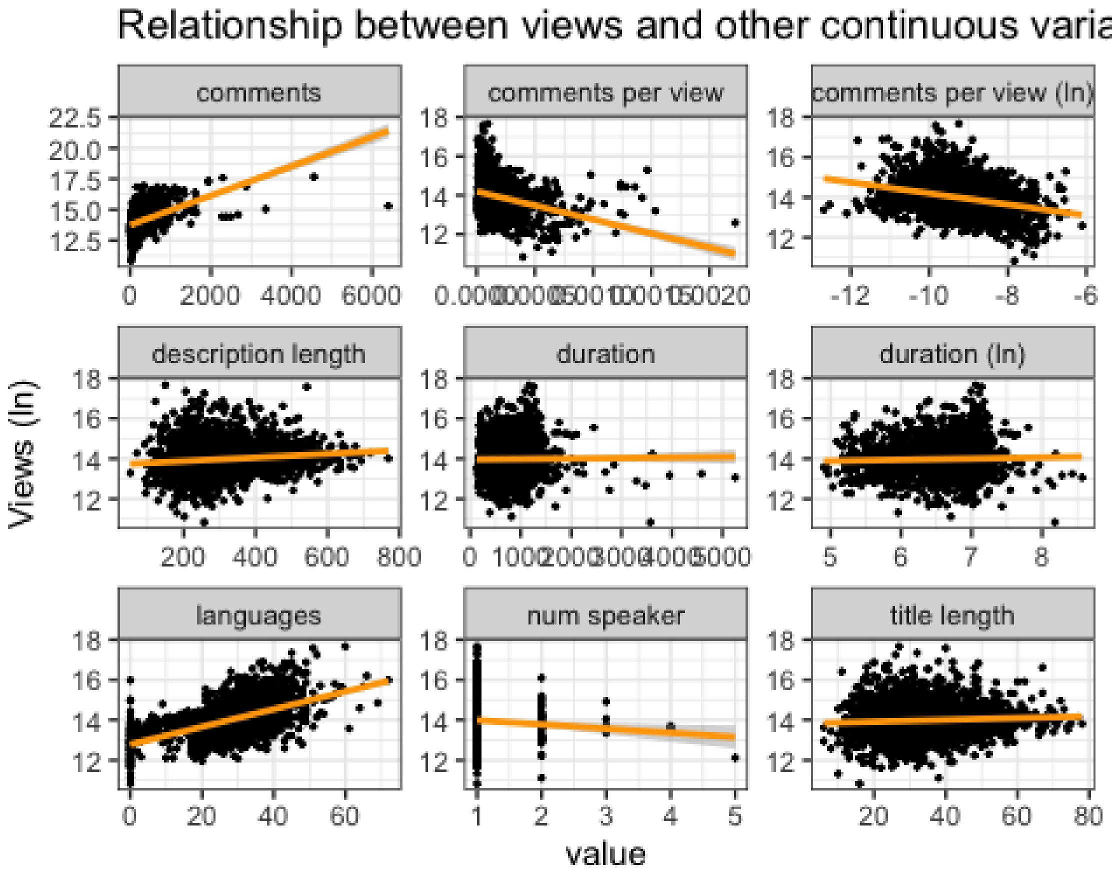
Section 1.2

explore variables visually

```
p1 <- ggplot(ted_data, aes(x=log(views))) +
  geom_histogram() +
  theme_bw() +
  labs(x='Views (ln)', y=NULL)
p2 <- ggplot(ted_data, aes(x=log(comments_per_view))) +
  geom_histogram() +
  theme_bw() +
  labs(x='Comments per View', y=NULL)
p3 <- ggplot(ted_data, aes(x=log(comments))) +
  geom_histogram() +
  theme_bw() +
  labs(x='Comments (ln)', y=NULL)
p4 <- ggplot(ted_data, aes(x=film_date)) + # by film date
  geom_histogram(bins=50) +
  theme_bw() +
  labs(x='Film Date', y=NULL)
p5 <- ggplot(ted_data, aes(x=published_date)) + # by film date
  geom_histogram(bins=60) +
  theme_bw() +
  labs(x='Publish Date', y=NULL)
p6 <- ggplot(ted_data, aes(x=languages)) + # by film date
  geom_histogram(bins=50) +
  theme_bw() +
  labs(x='Languages', y=NULL)

# look at distributions
grid.arrange(p1, p2, p3, p4, p5, p6, nrow=3, ncol=2, left='Count')
```



```
# explore relationships between numerical variables and views
nums <- unlist(lapply(ted_data, is.numeric)) #extract all numerical variables
data_temp <- ted_data[ , nums] %>%
  select(-talk_id) %>% # remove talk_id
  mutate(comments_per_view.log = log(comments_per_view), # add log of comments per view and
duration
         duration.log = log(duration)) %>%
  gather('variable', 'value', c(1, 2, 3, 4, 6, 7, 8, 9, 10)) %>%
  mutate(variable = str_replace_all(variable, '_', ' '),
         variable = str_replace(variable, '.log', ' (ln)'))

ggplot(data_temp, aes(x=value, y=log(views))) +
  geom_jitter(size=0.5) +
  geom_smooth(method='lm', colour='orange') +
  facet_wrap(~variable, scales='free')  +
  labs(title = 'Relationship between views and other continuous variables', y='Views (ln)',
'Value') +
  theme_bw()
```



Section 2.1

Calculte and explore ratings

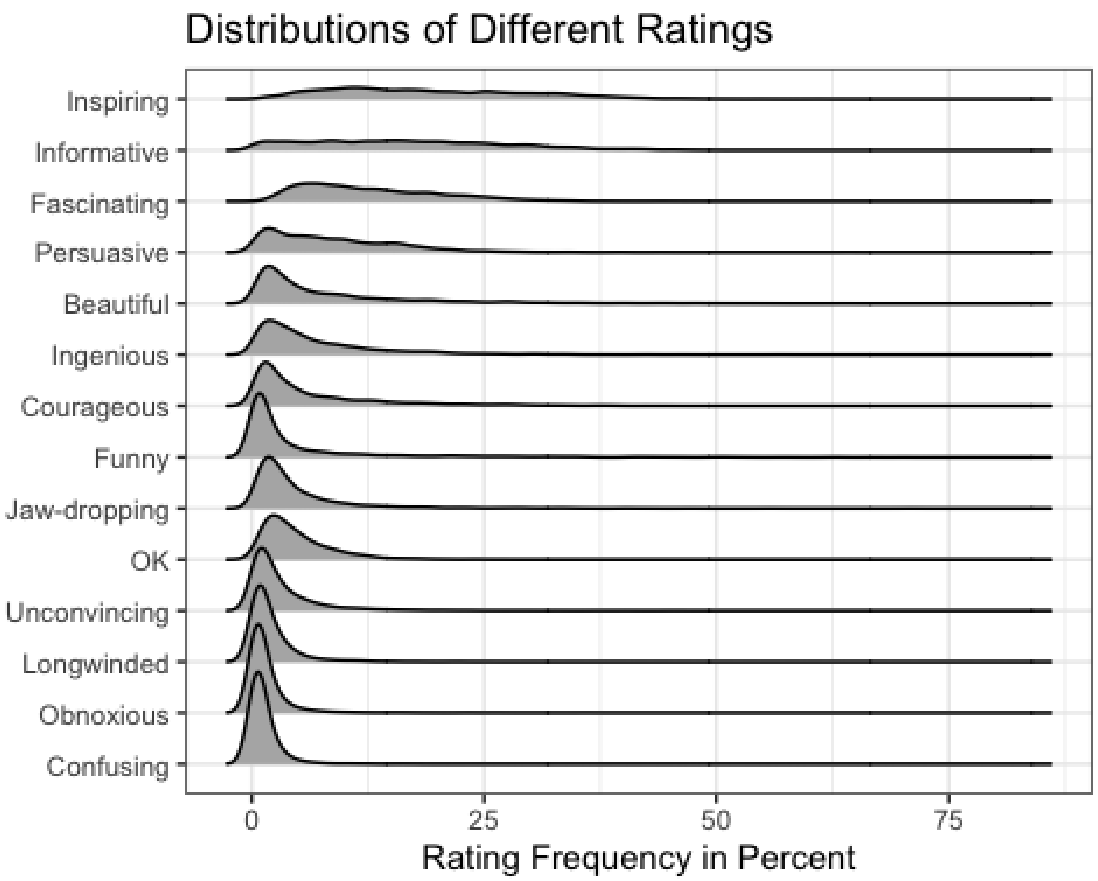
Ratings are currently in a dictionary type format and need to be extracted first.

```
# analyse ratings
# function to clean up ratings
clean_ratings <- function(n){

  ted_ratings <- data.frame(ratings = ted_data$ratings[n]) %>%
    mutate(ratings2 = toString(ratings),
           ratings2 = str_remove_all(ratings2, '\\]|\\{|\\,|\\ '),
           ratings2 = strsplit(ratings2, split='}', fixed=TRUE))

  data <- data.frame(var1 = ted_ratings$ratings2[1]) %>%
    separate(1, into = c('empty', 'id', 'name', 'count'), sep=':', remove=TRUE) %>%
    select(-empty) %>%
    mutate(id = str_remove(id, ", 'name'"),
           name = str_remove(name, "''count'"),
           name = str_remove(name, "'"),
           count = as.integer(count)) %>%
    group_by(id) %>%
    mutate(total_count = sum(count)) %>%
    ungroup() %>%
    select(name, total_count) %>%
    distinct()
  row.names(data) <- data$name
  data <- data %>%
    select(total_count) %>%
    t()
  data <- as.data.frame(data) %>%
    mutate(talk_id = ted_data$talk_id[n])
  return(data)
}

# clean ratings and make ratings df
ratings_data <- clean_ratings(1)

for (i in 2:as.integer(count(ted_data))) {
  temp_data <- clean_ratings(i)
  ratings_data <- rbind(ratings_data, temp_data)}

ratings_data <- ratings_data %>%
  mutate(total = Funny + Beautiful + Ingenious + Courageous + Longwinded + Confusing + Info
rmative + Fascinating +
           Unconvincing + Persuasive + `Jaw-dropping` + OK + Obnoxious + Inspiring)


# convert to percentage
temp <- select(ratings_data, -talk_id, -total)
ratings_percent <- (temp / rowSums(temp) * 100) %>%
  mutate(talk_id = ted_data$talk_id)
```

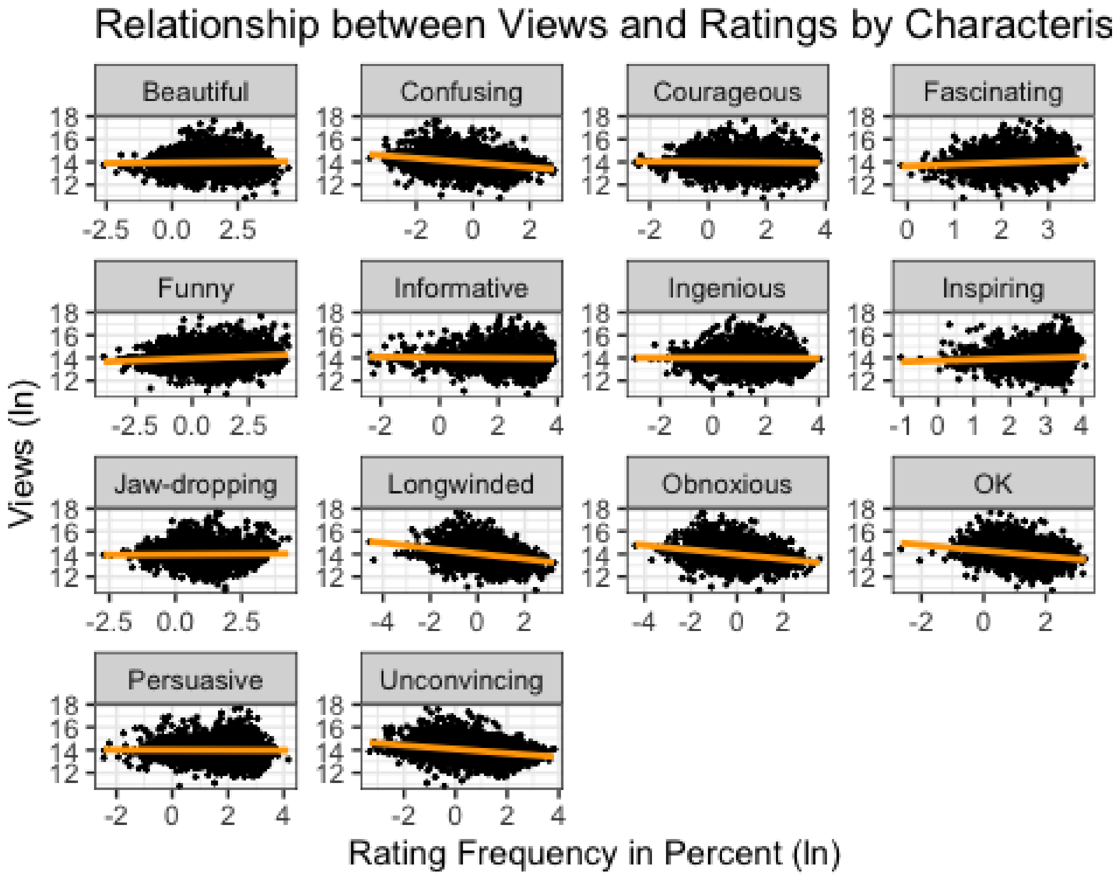
Section 2.2

Visualise ratings: Distributions and relationships with other variable.

```
# look at percentage because more views are likely linked to more comments
ratings_temp <- gather(ratings_percent, 'characteristic', 'rating', c('Funny':'Inspiring'))

# distributions of percentage ratings
ratings_temp %>% right_join(select(ted_data, talk_id, views), by='talk_id') %>%
  ggplot(aes(x=rating, y=reorder(characteristic, rating))) +
  geom_density_ridges_gradient() +
  labs(title = 'Distributions of Different Ratings',
       x = 'Rating Frequency in Percent',
       y = NULL) +
  theme_bw()
```



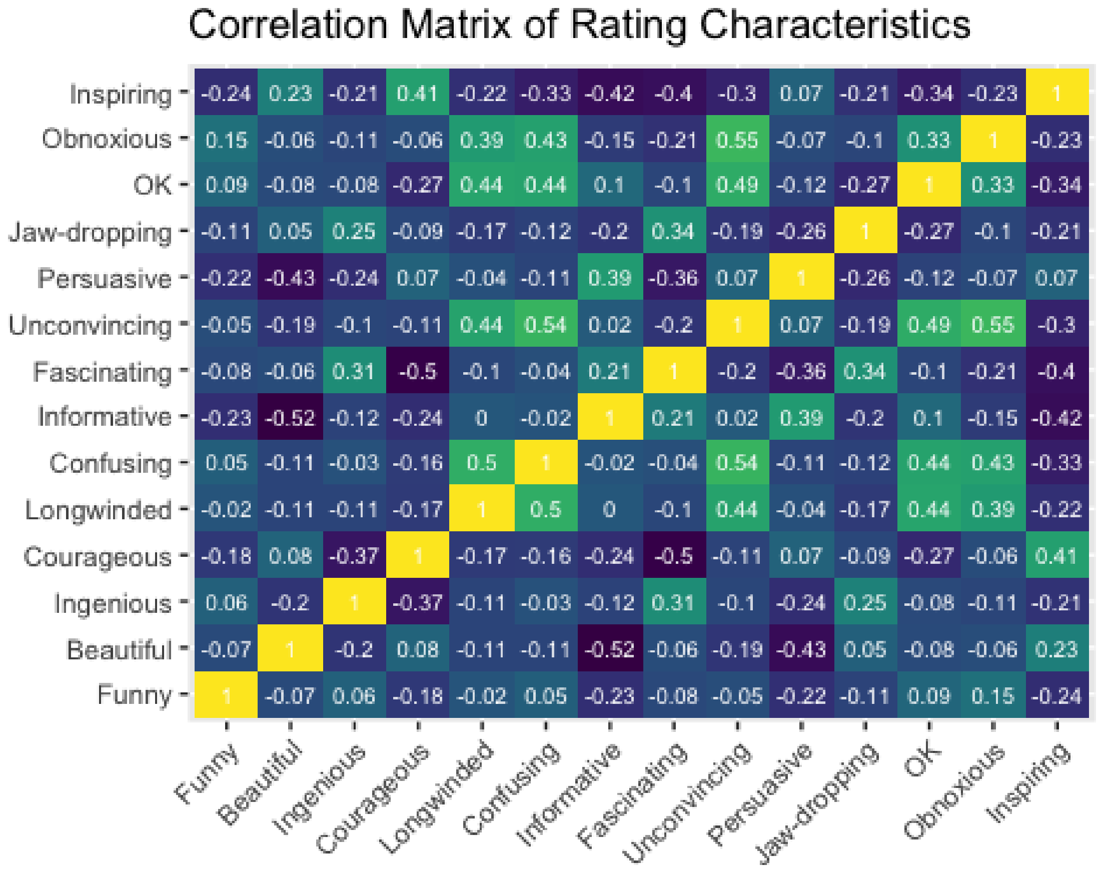
```r
ratings_temp %>% right_join(select(ted_data, talk_id, views), by='talk_id') %>%
  filter(rating != 0) %>%
  ggplot(aes(x=log(rating), y=log(views))) +
  geom_point(size=0.3) +
  geom_smooth(method='lm', colour='orange') +
  facet_wrap(~characteristic, scales = 'free') +
  labs(title = 'Relationship between Views and Ratings by Characteristic',
       x = 'Rating Frequency in Percent (ln)',
       y = 'Views (ln)') +
  theme_bw()
```



Section 2.3

Use ratings to cluster

```
# PCA assumes that variables are uncorrelated
cor_matrix.ratings <- cor(select(ratings_percent, -talk_id), method = c("pearson"))

ggplot(data = melt(cor_matrix.ratings), aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  scale_fill_viridis(guide=FALSE) +
  geom_text(aes(label=round(value, 2)), colour='white', size=2.5) +
  labs(x=NULL, y=NULL,
       title = 'Correlation Matrix of Rating Characteristics')
```

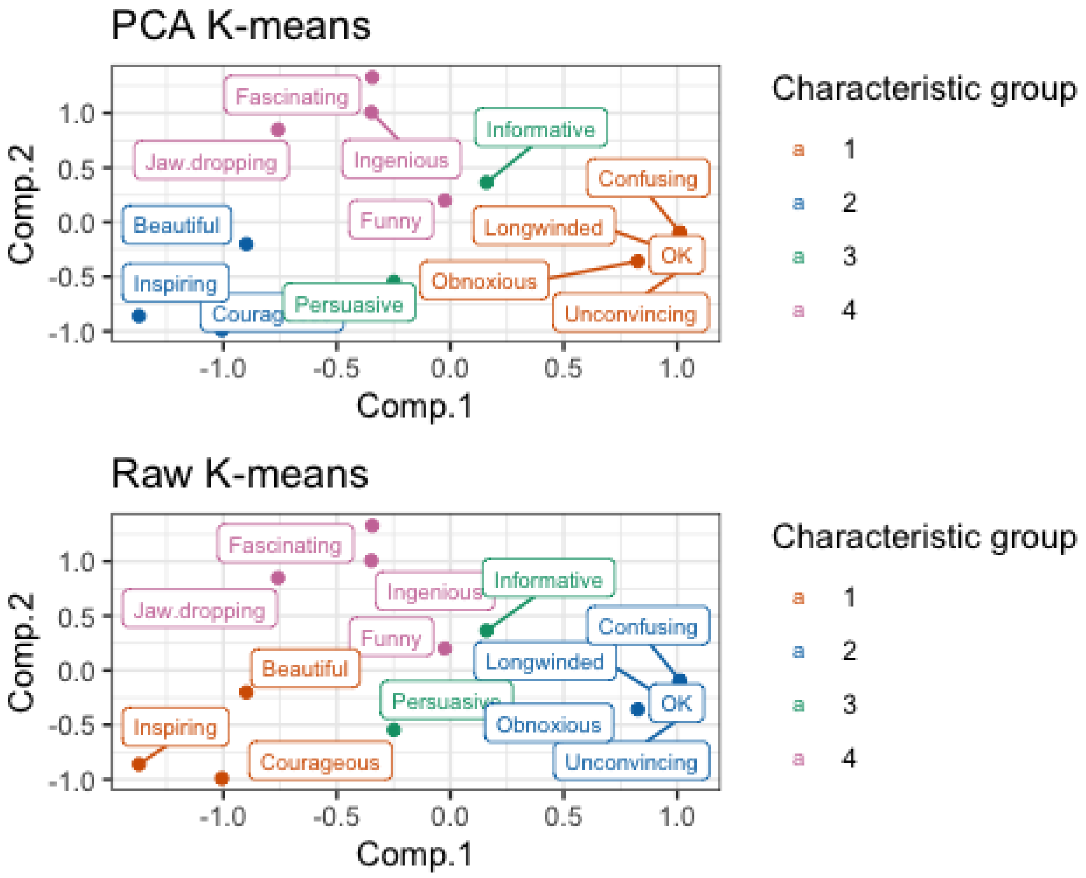


Correlation Matrix of Rating Characteristics

	Funny	Beautiful	Ingenious	Courageous	Longwinded	Confusing	Informative	Fascinating	Unconvincing	Persuasive	Jaw-dropping	OK	Obnoxious	Inspiring
Inspiring	-0.24	0.23	-0.21	0.41	-0.22	-0.33	-0.42	-0.4	-0.3	0.07	-0.21	-0.34	-0.23	1
Obnoxious	0.15	-0.06	-0.11	-0.06	0.39	0.43	-0.15	-0.21	0.55	-0.07	-0.1	0.33	1	-0.23
OK	0.09	-0.08	-0.08	-0.27	0.44	0.44	0.1	-0.1	0.49	-0.12	-0.27	1	0.33	-0.34
Jaw-dropping	-0.11	0.05	0.25	-0.09	-0.17	-0.12	-0.2	0.34	-0.19	-0.26	1	-0.27	-0.1	-0.21
Persuasive	-0.22	-0.43	-0.24	0.07	-0.04	-0.11	0.39	-0.36	0.07	1	-0.26	-0.12	-0.07	0.07
Unconvincing	-0.05	-0.19	-0.1	-0.11	0.44	0.54	0.02	-0.2	1	0.07	-0.19	0.49	0.55	-0.3
Fascinating	-0.08	-0.06	0.31	-0.5	-0.1	-0.04	0.21	1	-0.2	-0.36	0.34	-0.1	-0.21	-0.4
Informative	-0.23	-0.52	-0.12	-0.24	0	-0.02	1	0.21	0.02	0.39	-0.2	0.1	-0.15	-0.42
Confusing	0.05	-0.11	-0.03	-0.16	0.5	1	-0.02	-0.04	0.54	-0.11	-0.12	0.44	0.43	-0.33
Longwinded	-0.02	-0.11	-0.11	-0.17	1	0.5	0	-0.1	0.44	-0.04	-0.17	0.44	0.39	-0.22
Courageous	-0.18	0.08	-0.37	1	-0.17	-0.16	-0.24	-0.5	-0.11	0.07	-0.09	-0.27	-0.06	0.41
Ingenious	0.06	-0.2	1	-0.37	-0.11	-0.03	-0.12	0.31	-0.1	-0.24	0.25	-0.08	-0.11	-0.21
Beautiful	-0.07	1	-0.2	0.08	-0.11	-0.11	-0.52	-0.06	-0.19	-0.43	0.05	-0.08	-0.06	0.23
Funny	1	-0.07	0.06	-0.18	-0.02	0.05	-0.23	-0.08	-0.05	-0.22	-0.11	0.09	0.15	-0.24

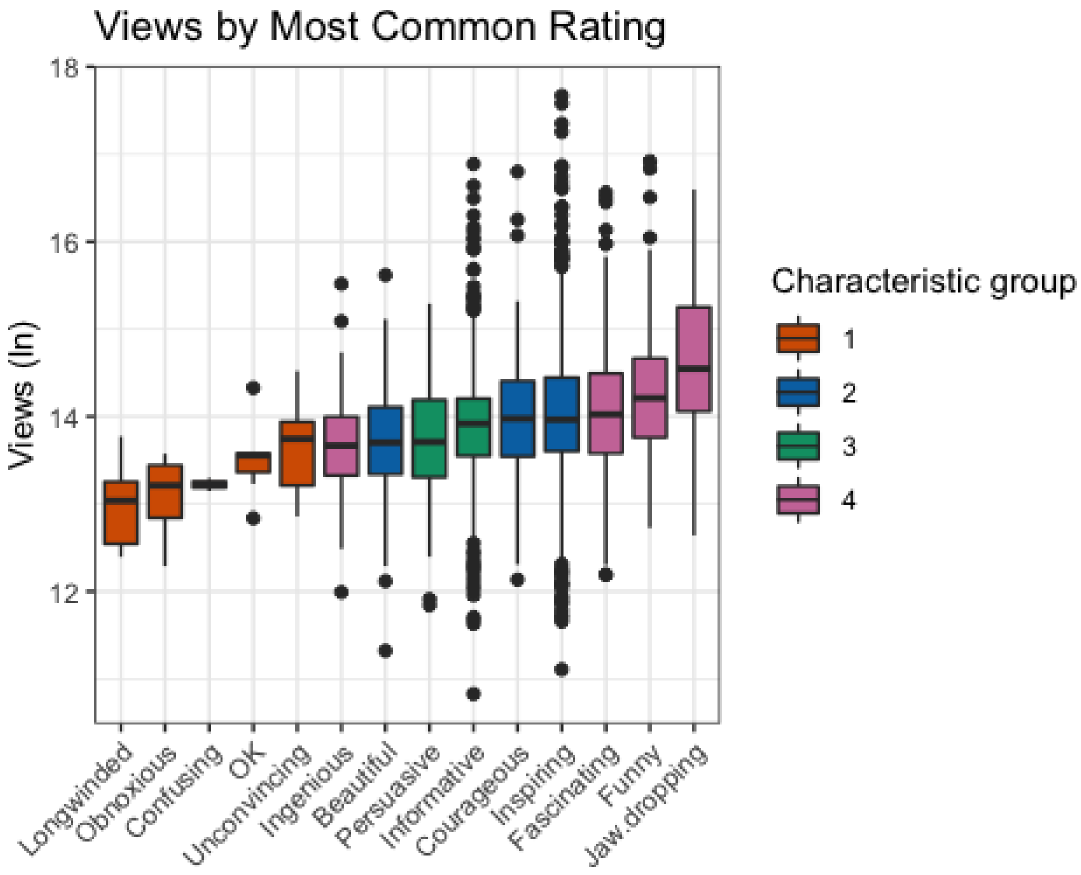
```
# --> mostly not correlated, but some at +/-0.5 (Beautiful & Informative, and Fascinating &
Courageous)

pca1 <- princomp(data.frame(cor_matrix.ratings))

summary(pca1) # shows variance explained by components

## Importance of components:
##                           Comp.1    Comp.2    Comp.3     Comp.4     Comp.5
## Standard deviation     0.8164320 0.6577584 0.5441495 0.30585829 0.26642943
## Proportion of Variance 0.3865779 0.2509169 0.1717250 0.05425473 0.04116816
## Cumulative Proportion  0.3865779 0.6374948 0.8092198 0.86347451 0.90464267
##                            Comp.6     Comp.7     Comp.8     Comp.9
## Standard deviation     0.22326903 0.16725177 0.15885538 0.14324012
## Proportion of Variance 0.02891039 0.01622327 0.01463528 0.01189944
## Cumulative Proportion  0.93355306 0.94977634 0.96441162 0.97631105
##                             Comp.10     Comp.11     Comp.12     Comp.13
## Standard deviation     0.129908279 0.105936098 0.092616658 0.064571365
## Proportion of Variance 0.009787476 0.006508561 0.004974796 0.002418115
## Cumulative Proportion  0.986098528 0.992607089 0.997581885 1.000000000
##                             Comp.14
## Standard deviation     1.850013e-09
## Proportion of Variance 1.984937e-18
## Cumulative Proportion  1.000000e+00

pca_data <- data.frame(pca1$scores)[,c(1,2,3)] %>%
  # here using the first 3 components to explain 80% of variance in data
  # if using 2 only 64%
  mutate(characteristic = row.names(pca1$loadings))

set.seed(1) # set random seed to ensure replicability of results
kmns_pca <- kmeans(select(pca_data, -characteristic), 4) # divide data (first 2 pca comps)
```

```
into 3 clusters

# plot pca vs non-pca cluster results
kmns <- kmeans(data.frame(cor_matrix.ratings), 4)

cluster_data <- pca_data %>%
  mutate(cluster_pca = paste('Cluster', kmns_pca$cluster, sep='_'),
         cluster = paste('Cluster_', kmns$cluster),
         characteristic = pca_data$characteristic)


p1 <- ggplot(cluster_data, aes(x=Comp.1, y=Comp.2, colour=cluster_pca, label=characteristic
)) +
  geom_point()+
  geom_label_repel(size=2.5) +
  labs(title = 'PCA K-means',
       colour = 'Characteristic group') +
  scale_colour_manual(labels=c('1', '2', '3', '4'),
                      values=c("#D55E00", "#0072B2", "#009E73", "#CC79A7")) +
  theme_bw()

p2 <- ggplot(cluster_data, aes(x=Comp.1, y=Comp.2, colour=cluster, label=characteristic)) +
  geom_point() +
  geom_label_repel(size=2.5) +
  labs(title = 'Raw K-means',
       colour = 'Characteristic group') +
  scale_colour_manual(labels=c('1', '2', '3', '4'),
                      values=c("#D55E00", "#0072B2", "#009E73", "#CC79A7")) +
  theme_bw()

grid.arrange(p1, p2, nrow=2)
```

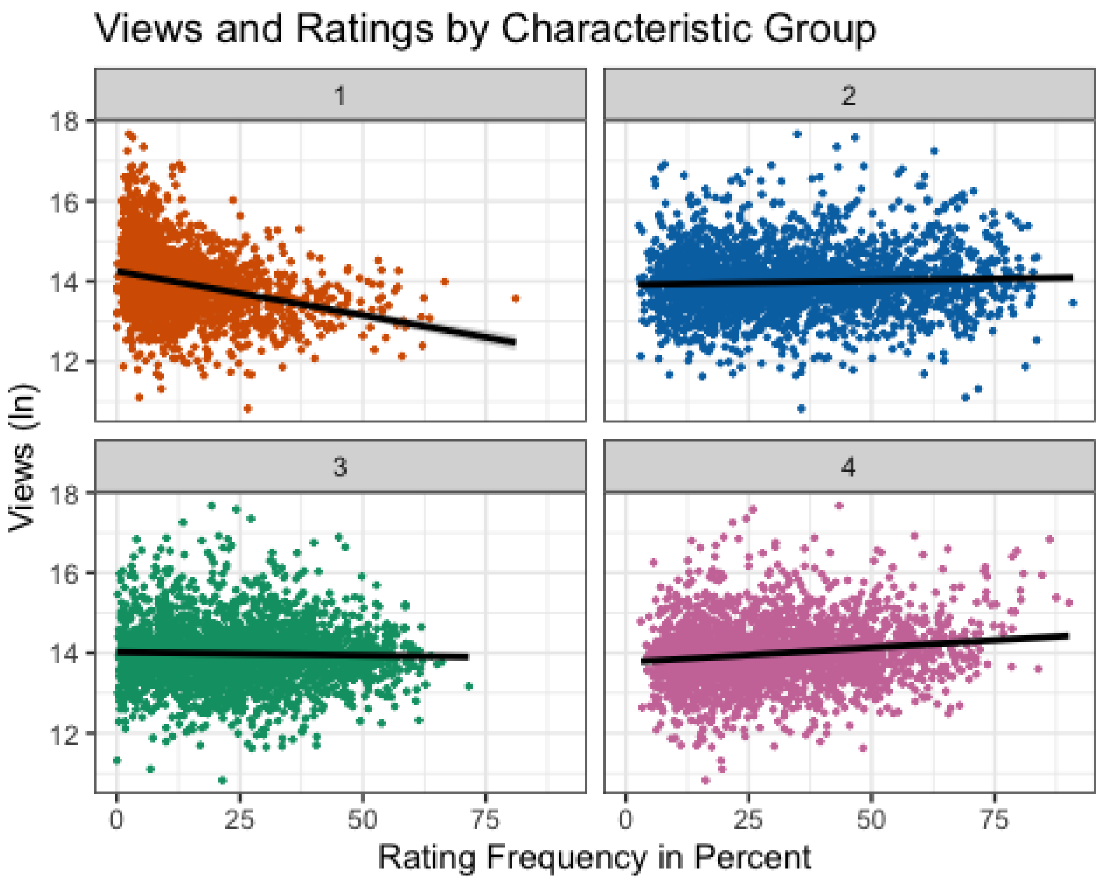


Section 2.4

```
rating_cluster <- ratings_temp %>%
  #gsub(characteristic, "-", ".") %>%
  mutate(characteristic = ifelse(characteristic == 'Jaw-dropping', 'Jaw.dropping', characte
ristic)) %>%
  left_join(select(cluster_data, characteristic, cluster_pca), by='characteristic')

cluster_lookup <- rating_cluster %>%
  select(cluster_pca, characteristic) %>%
  distinct()

rating_cluster %>%
  group_by(talk_id) %>%
  top_n(1, rating) %>%
  left_join(ted_data, by='talk_id') %>%
  ggplot(aes(x=reorder(characteristic, log(views)), y=log(views), fill=cluster_pca)) +
  geom_boxplot() +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  scale_fill_manual(labels=c('1', '2', '3', '4'),
                    values=c("#D55E00", "#0072B2", "#009E73", "#CC79A7")) +
  theme_bw() +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  labs(title='Views by Most Common Rating',
       x=NULL,
       y='Views (ln)',
       fill='Characteristic group')
```



```
# ratings and views
cluster_ratings <- select(ratings_data, -total) %>%
  gather(characteristic, rating, c(1:14)) %>%
  mutate(characteristic = ifelse(characteristic == 'Jaw-dropping', 'Jaw.dropping', characteristic)) %>%
  left_join(distinct(select(rating_cluster, cluster_pca, characteristic)), by='characteristic') %>%
  group_by(talk_id, cluster_pca) %>%
  mutate(total = sum(rating)) %>%
  select(-characteristic, -rating) %>%
  distinct() %>%
  group_by(talk_id) %>%
  mutate(percent = total/sum(total) *100) %>%
  ungroup() %>%
  select(talk_id, cluster_pca, percent)

cluster_ratings %>%
  mutate(cluster_pca = str_replace(cluster_pca, "Cluster_", "")) %>%
  left_join(ted_data, by='talk_id') %>%
  ggplot(aes(x=percent, y=log(views))) +
  geom_point(size=0.5, aes(colour=cluster_pca)) +
  scale_colour_manual(labels=c('1', '2', '3', '4'),
                      values=c("#D55E00", "#0072B2", "#009E73", "#CC79A7"),
                      guide=FALSE) +
  geom_smooth(method='lm', colour='black') +
  facet_wrap(~cluster_pca) +
  labs(title = 'Views and Ratings by Characteristic Group',
       y = 'Views (ln)',
       x = 'Rating Frequency in Percent') +
  theme_bw()
```

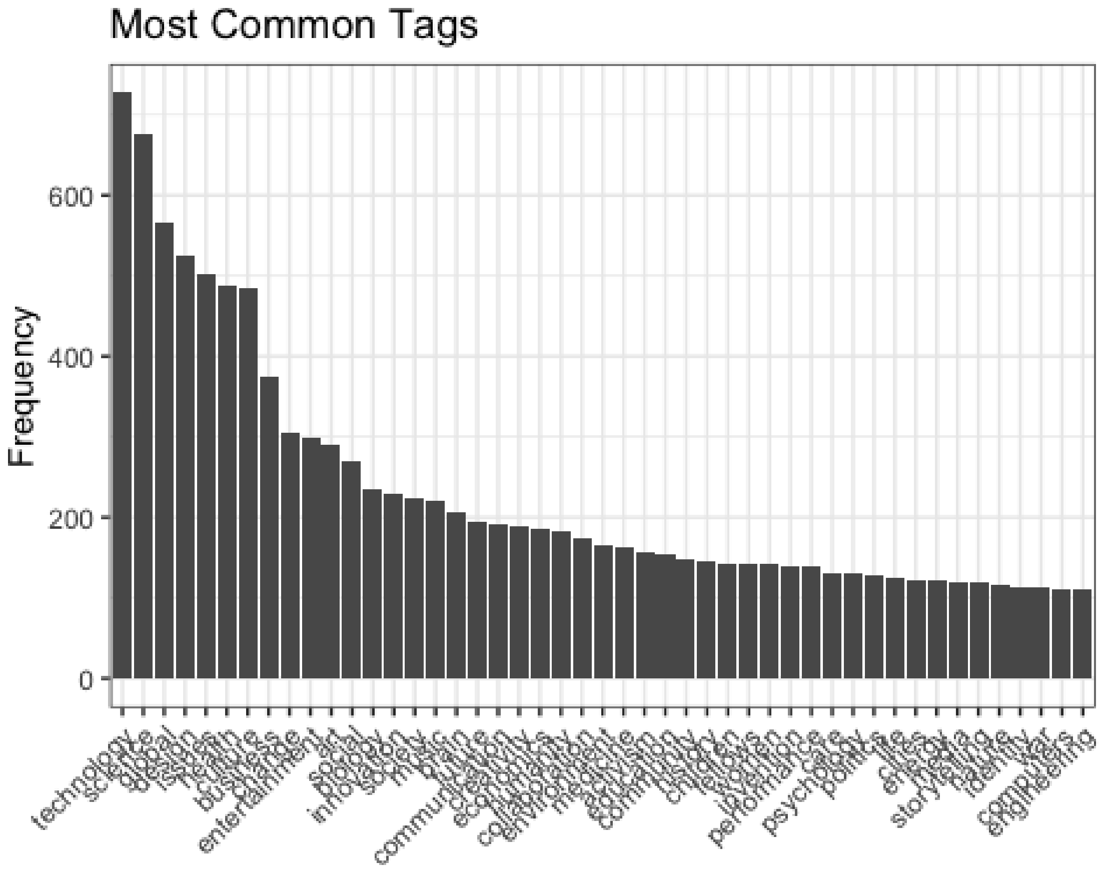


Section 3.1

Text Analysis

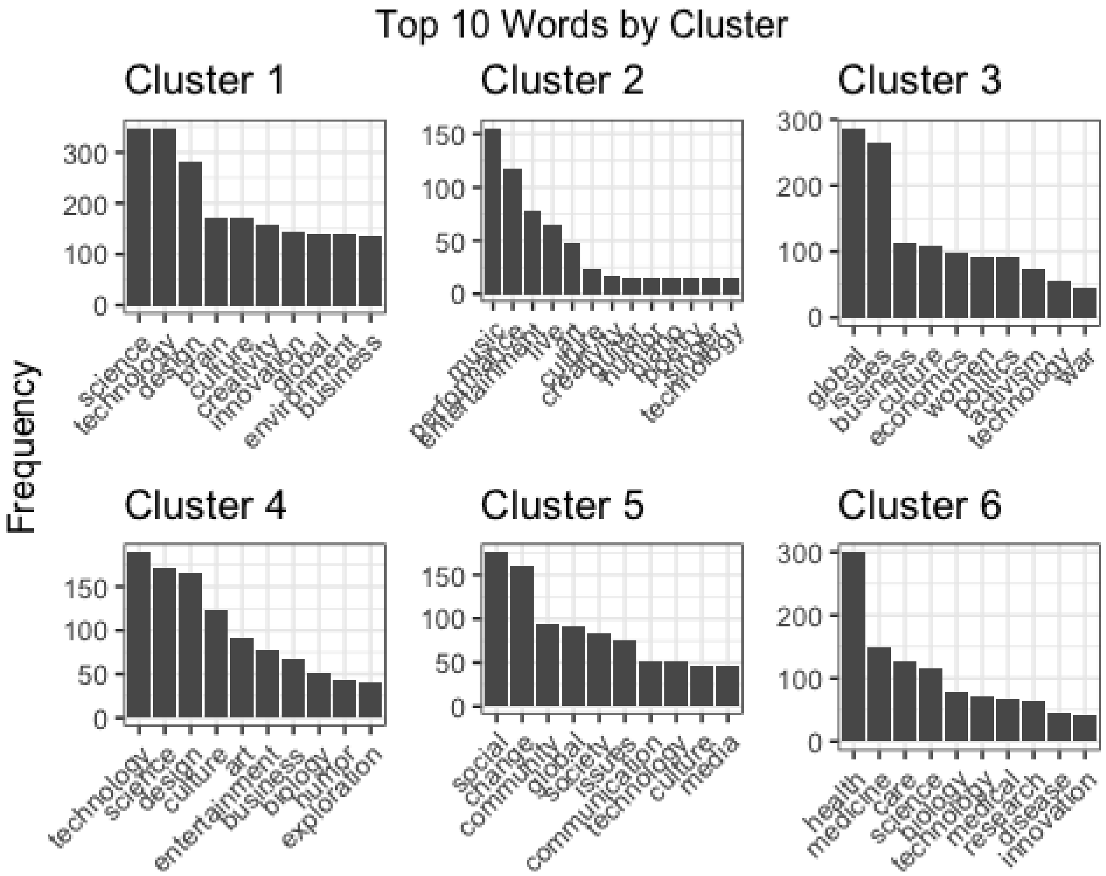
Cluster talks by keyword tags.

```r
# function for plot
wordfreq_by_cluster <- function(n_cluster, top_words) {
  data_temp <- filter(clusters, cuts == n_cluster)
  temp_corpus <- Corpus(VectorSource(data_temp$tags)) %>% # convert to corpus
    tm_map(removeWords, stopwords("english")) %>% # clean up
    tm_map(removePunctuation) %>%
    tm_map(stripWhitespace) %>%
    tm_map(removeWords, c('tedx', 'fellow', 'ted'))

  temp_dtm <- DocumentTermMatrix(temp_corpus)
  temp_freq <- colSums(as.matrix(temp_dtm))

  temp_freq_df <- data.frame(word=names(temp_freq), freq=temp_freq) %>%
    arrange(freq) %>%
    top_n(top_words)

  ggplot(subset(temp_freq_df), aes(x = reorder(word, -freq), y = freq)) +
    geom_bar(stat = "identity") +
    labs(x = NULL,
         y = NULL,
         title = paste('Cluster', n_cluster, sep=' '),
         size=5) +
    theme_bw() +
    theme(axis.text.x=element_text(angle=45, hjust=1))
}

# prepare data
tag_corpus <- Corpus(VectorSource(ted_data$tags)) %>% # convert to corpus
  tm_map(removeWords, stopwords("english")) %>% # clean up
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace) %>%
  tm_map(removeWords, c('tedx', 'fellow', 'ted'))

tag_dtm <- DocumentTermMatrix(tag_corpus)

# look at most frequent terms
tag_freq <- colSums(as.matrix(tag_dtm))

tag_freq_df <- data.frame(word=names(tag_freq), freq=tag_freq)

ggplot(subset(tag_freq_df, freq>110), aes(x = reorder(word, -freq), y = freq)) +
  geom_bar(stat = "identity") +
  theme_bw() +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  labs(title = 'Most Common Tags',
       y = 'Frequency',
       x = NULL)
```



```
# cluster by keywords
# hierarchical clustering on tags
sparse_tag_dtm <- removeSparseTerms(tag_dtm, sparse=0.95)
mat <- as.matrix(sparse_tag_dtm)
docsdissim <- dist(scale(mat))

h <- hclust(docsdissim, method = "ward.D")

cuts <- cutree(h, 6)
clusters <- data.frame(cuts) %>%
  mutate(talk_id = ted_data$talk_id) %>%
  left_join(ted_data, by='talk_id') %>%
  mutate(cluster = paste('group', as.character(cuts)),
         views = as.double(views))

clusters_final <- clusters

# get summary
cluster_summary_h <- clusters %>%
  group_by(cluster) %>%
  mutate(mean = mean(views), std = sd(views), count = n()) %>%
  ungroup() %>%
  select(cluster, mean, std, count) %>%
  distinct()

# plot word counts by cluster
p1 <- wordfreq_by_cluster(1, 10)
p2 <- wordfreq_by_cluster(2, 10)
p3 <- wordfreq_by_cluster(3, 10)
p4 <- wordfreq_by_cluster(4, 10)
p5 <- wordfreq_by_cluster(5, 10)
p6 <- wordfreq_by_cluster(6, 10)
```

```
grid.arrange(p1, p2, p3, p4, p5, p6, ncol=3, nrow=2,
             left="Frequency", top="Top 10 Words by Cluster")
```

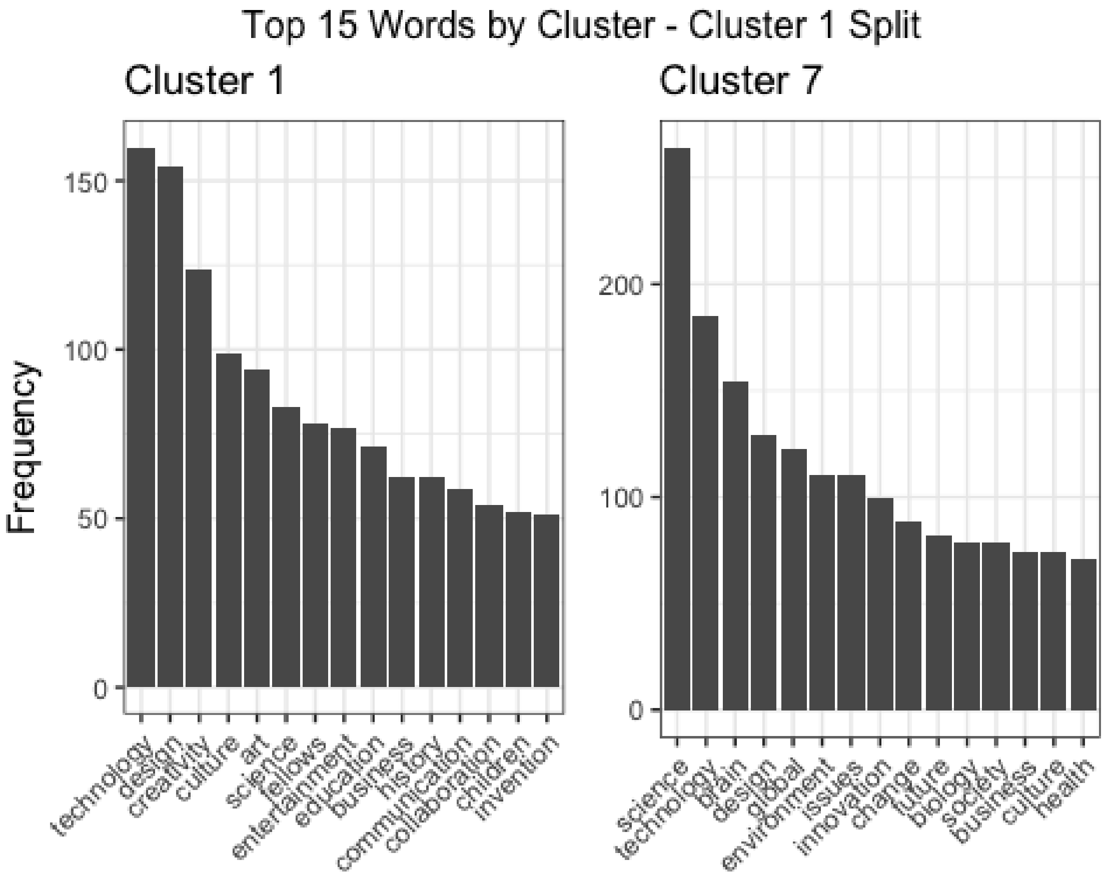


#

Section 3.2 split cluster 1 into 2 new clusters.

```
##### split up cluster one further:
temp_data <- clusters_final %>%
  filter(cuts==1) %>%
  select(-cuts, -cluster)

tag_corpus.cluster1 <- Corpus(VectorSource(temp_data$tags)) %>% # convert to corpus
  tm_map(removeWords, stopwords("english")) %>% # clean up: remove stopwords, punctionation
, extra white space, and specific common, meaningless terms
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace) %>%
  tm_map(removeWords, c('tedx', 'fellow', 'ted'))

tag_dtm.cluster1 <- DocumentTermMatrix(tag_corpus.cluster1)

# cluster by keywords
# hierarchical clustering on tags
sparse_tag_dtm.cluster1 <- removeSparseTerms(tag_dtm.cluster1, sparse=0.95)
mat.cluster1 <- as.matrix(sparse_tag_dtm.cluster1)
docsdissim.cluster1 <- dist(scale(mat.cluster1))

h.cluster1 <- hclust(docsdissim.cluster1, method = "ward.D")

cuts.cluster1 <- cutree(h.cluster1, 2)
group_no_other <- max(clusters_final$cuts)-1

clusters.cluster1 <- data.frame(cuts.cluster1) %>%
```

```r
  mutate(talk_id = temp_data$talk_id) %>%
  left_join(temp_data, by='talk_id') %>%
  mutate(cuts.split1 = ifelse(cuts.cluster1==1, 1, cuts.cluster1+group_no_other),
         views = as.double(views))

clusters_final.cluster1 <- clusters.cluster1 %>%
  select(talk_id, cuts.split1)

clusters_final.all <- clusters_final %>%
  left_join(clusters_final.cluster1, by='talk_id') %>%
  mutate(cuts.split1 = ifelse(is.na(cuts.split1), cuts, cuts.split1),
         cluster.split1 = ifelse(cluster=='group 1', paste('group', as.character(cuts.split
1)), cluster))

# get summary
cluster_summary_h.split1 <- clusters_final.all %>%
  group_by(cluster.split1) %>%
  mutate(mean = mean(views), std = sd(views), count = n()) %>%
  ungroup() %>%
  select(cluster.split1, mean, std, count) %>%
  distinct()

clusters <- mutate(clusters_final.all, cuts = cuts.split1)

# plot word counts by cluster
p1 <- wordfreq_by_cluster(1, 15)
p7 <- wordfreq_by_cluster(7, 15)

grid.arrange(p1, p7, ncol=2, nrow=1,
             left="Frequency", top="Top 15 Words by Cluster - Cluster 1 Split")
```

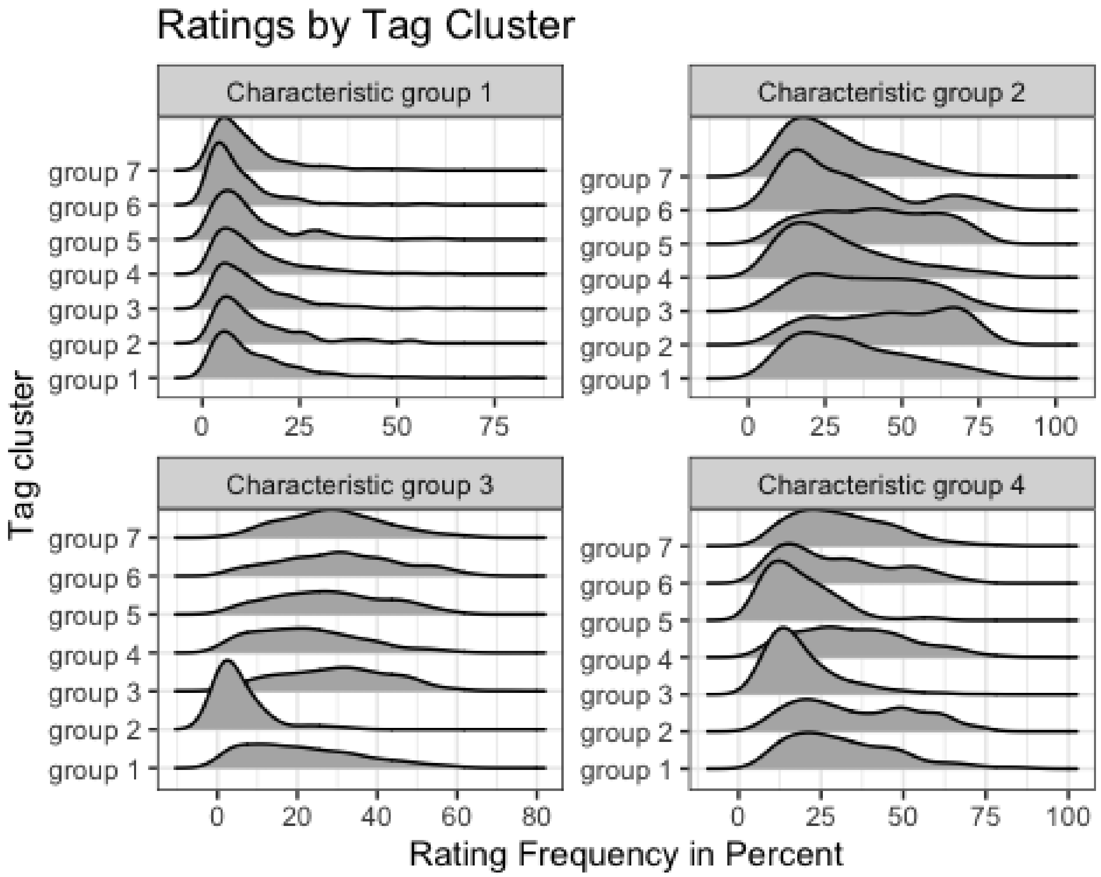


#

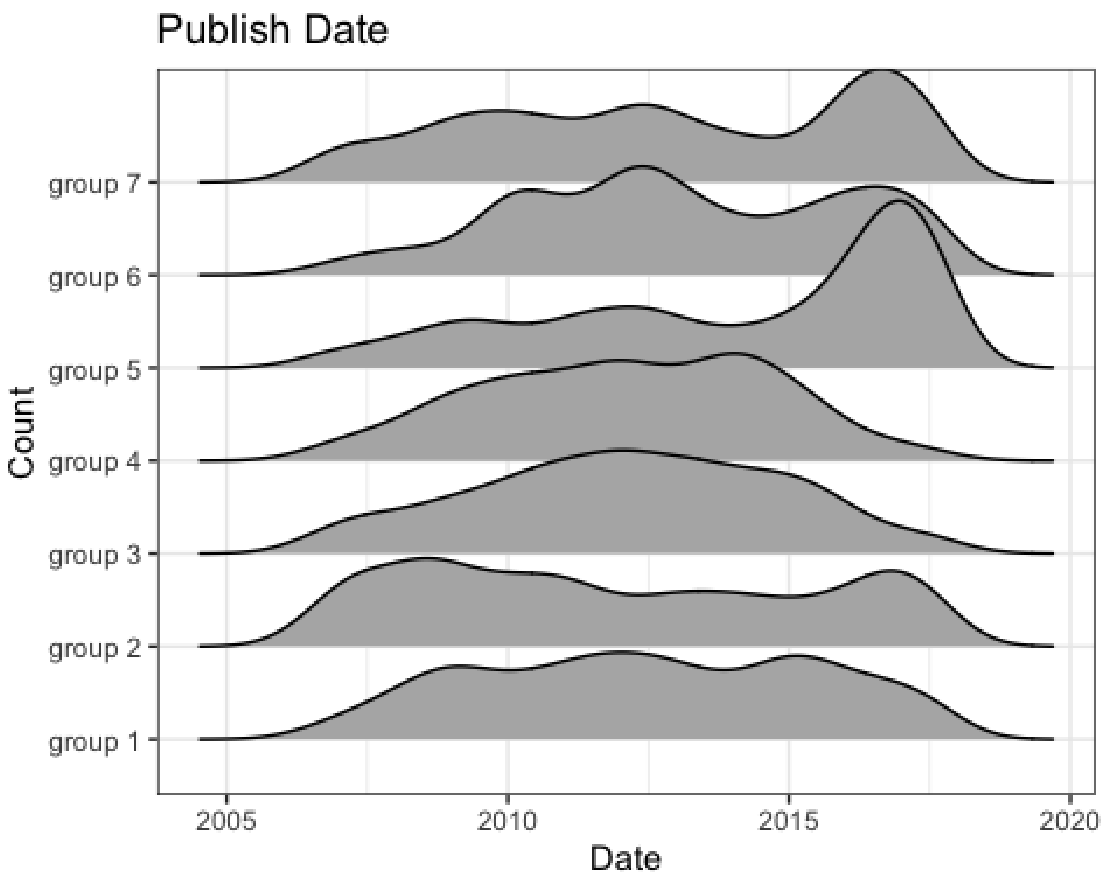
Section 3.3

Explore relationships between tag clusters and other variables.

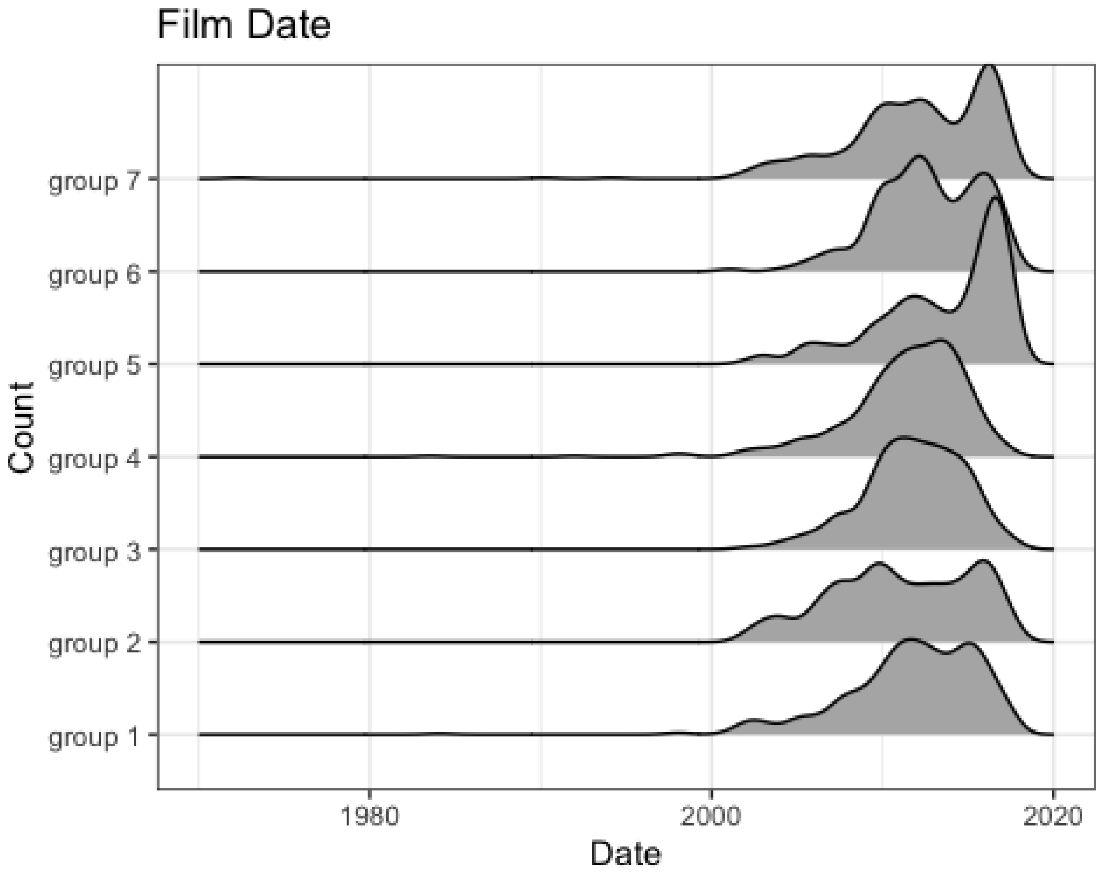
```
clusters_final.all %>%
  left_join(cluster_ratings, by='talk_id') %>%
  mutate(cluster_pca = str_replace(cluster_pca, "Cluster_", "Characteristic group ")) %>%
  ggplot(aes(x=percent, y=cluster.split1)) +
  geom_density_ridges_gradient() +
  scale_fill_continuous() +
  facet_wrap(~cluster_pca, scales='free') +
  theme_bw() +
  labs(title='Ratings by Tag Cluster',
       x='Rating Frequency in Percent',
       y='Tag cluster')
```



```
clusters_final.all %>%
  left_join(cluster_ratings, by='talk_id') %>%
  ggplot(aes(x=published_date, y=cluster.split1)) +
  geom_density_ridges() +
  theme_bw() +
  labs(title='Publish Date', x='Date', y='Count', fill='Tag cluster')
```



```r
clusters_final.all %>%
  left_join(cluster_ratings, by='talk_id') %>%
  ggplot(aes(x=film_date, y=cluster.split1)) +
  geom_density_ridges() +
  theme_bw() +
  labs(title='Film Date', x='Date', y='Count', fill='Tag cluster')
```



```
# explore variables by h clusters

mean <- clusters_final.all %>%
  select(views, comments_per_view, languages, duration, title_length, cluster.split1) %>%
  group_by(cluster.split1) %>%
  summarise(views=mean(views),
            comments_per_view=mean(comments_per_view),
            languages=mean(languages),
            title_length=mean(title_length)) %>%
  ungroup()

summary <- clusters_final.all %>%
  select(views, comments_per_view, languages, duration, title_length, cluster.split1) %>%
  group_by(cluster.split1) %>%
  summarise(views_sd=sd(views),
            comments_per_view_sd=sd(comments_per_view),
            languages_sd=sd(languages),
            title_length_sd=sd(title_length)) %>%
  ungroup() %>%
  full_join(mean, by='cluster.split1')

names(summary)[names(summary)=='cluster.split1'] <- 'Cluster'
pander(summary[ , order(names(summary))])
```

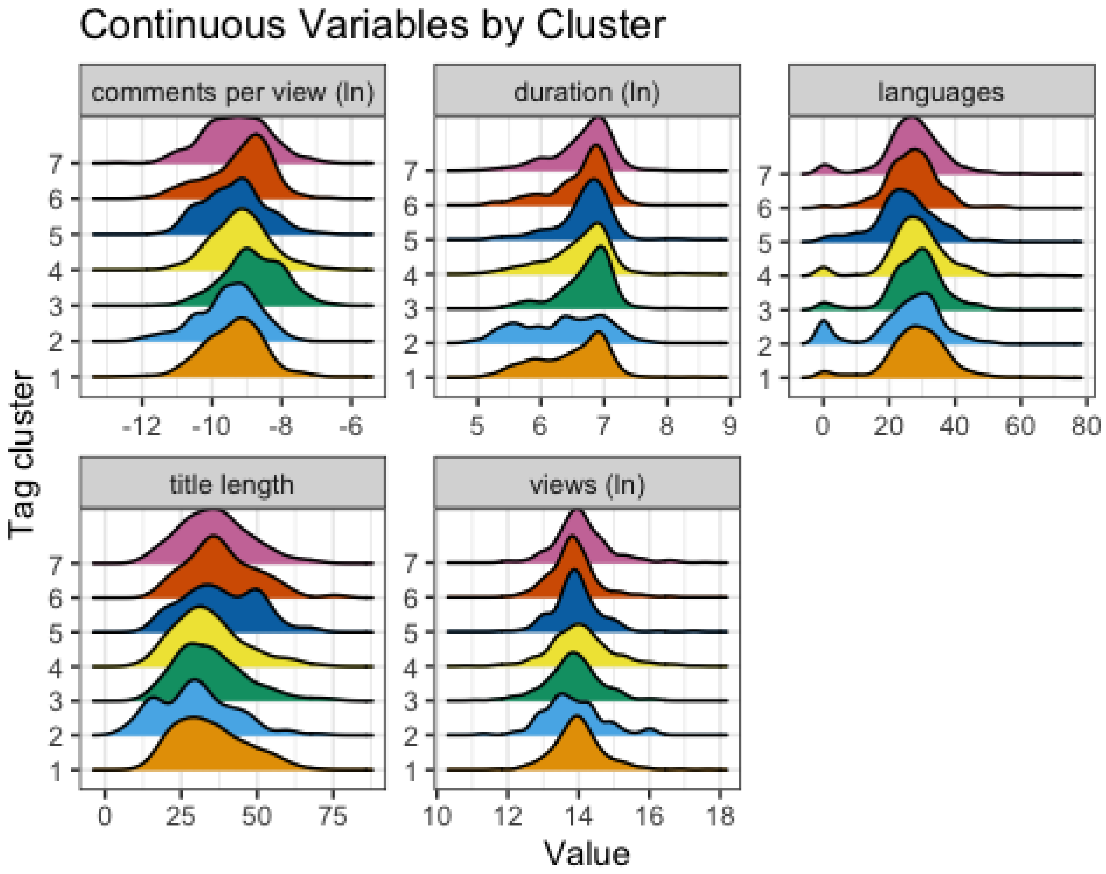
Table continues below

Cluster	comments_per_view	comments_per_view_sd	languages	languages_sd
group 1	0.0001183	0.0001139	28.39	9.756
group 2	0.0001014	8.333e-05	24.95	11.76

group 3	0.0002125	0.0001988	27.96	8.941
group 4	0.0001534	0.000187	28.12	9.713
group 5	0.0001275	0.0001432	24.71	9.161
group 6	0.0001445	0.0001321	26.93	7.786
group 7	0.0001396	0.0001512	26.8	9.529

title_length	title_length_sd	views	views_sd
34.56	11.69	1672153	2726636
29.74	12.66	1539769	1839860
34.73	11.7	1390121	1331611
34.18	11.57	2009432	2884849
38.08	12.15	1474493	2260907
38.36	11.53	1333179	1514802
36.05	11.43	1912565	2994173

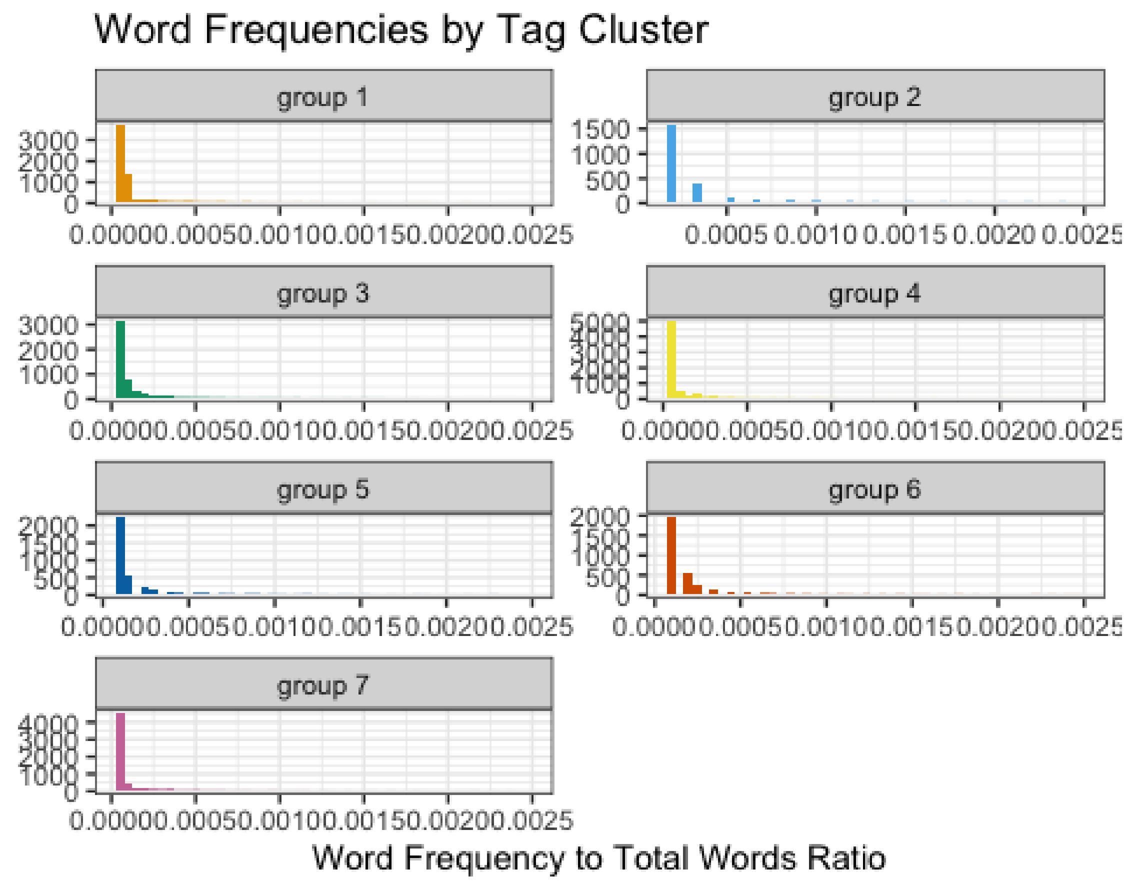
```r
clusters_final.all %>%
  mutate(views_log = log(views),
         comments_per_view_log = log(comments_per_view),
         duration_log = log(duration)) %>%
  select(cluster.split1, views_log, comments_per_view_log, languages, duration_log, title_length) %>%
  gather('var', 'value', c(2:6)) %>%
  mutate(var = str_replace(str_replace_all(paste(var), '_', ' '), 'log', '(ln)'),
         cluster.split1 = str_replace(paste(cluster.split1), 'group ', '')) %>%
  ggplot(aes(x=value, y=cluster.split1, fill=cluster.split1)) +
  geom_density_ridges_gradient(show.legend = FALSE, bins=50) +
  facet_wrap(~var, scale='free') +
  theme_bw() +
  scale_fill_manual(labels=c('1', '2', '3', '4', '5', '6', '7'),
                    values=c("#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2",
                             "#D55E00", "#CC79A7")) +
  labs(title='Continuous Variables by Cluster',
       x='Value',
       y='Tag cluster')
```



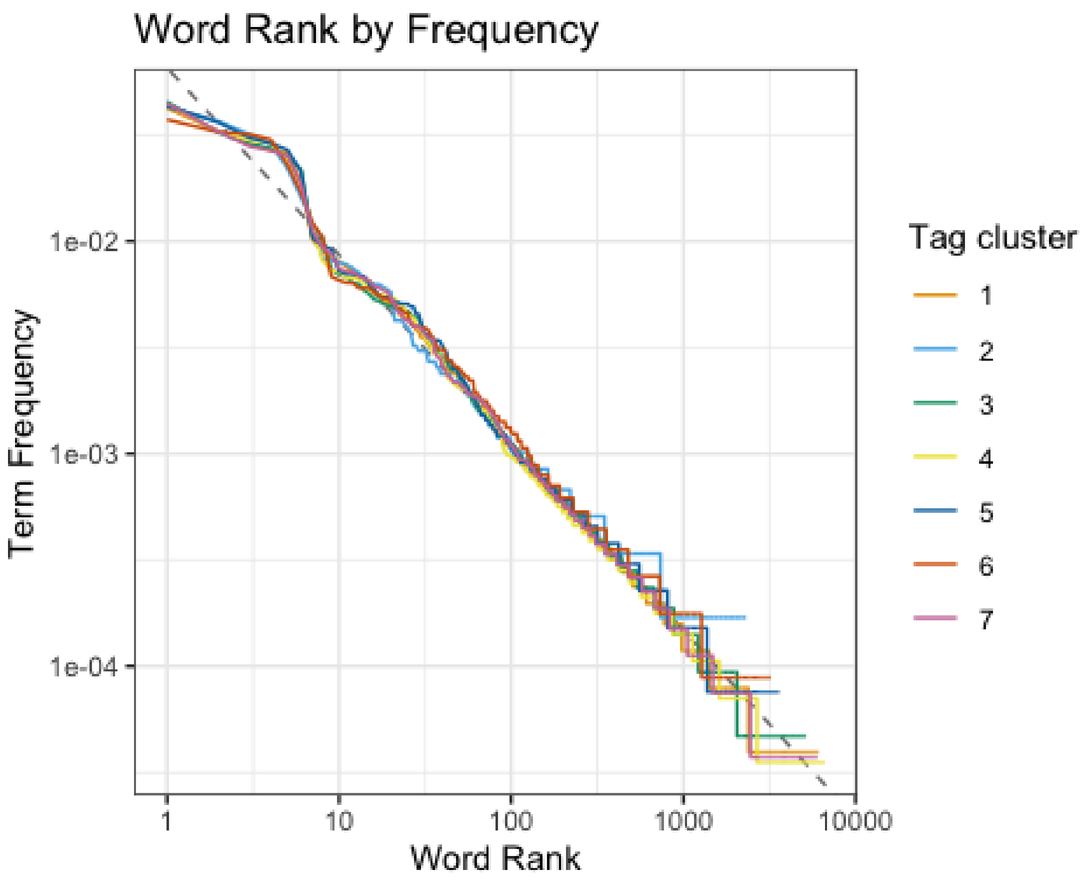
Section 4.1

TF-IDF

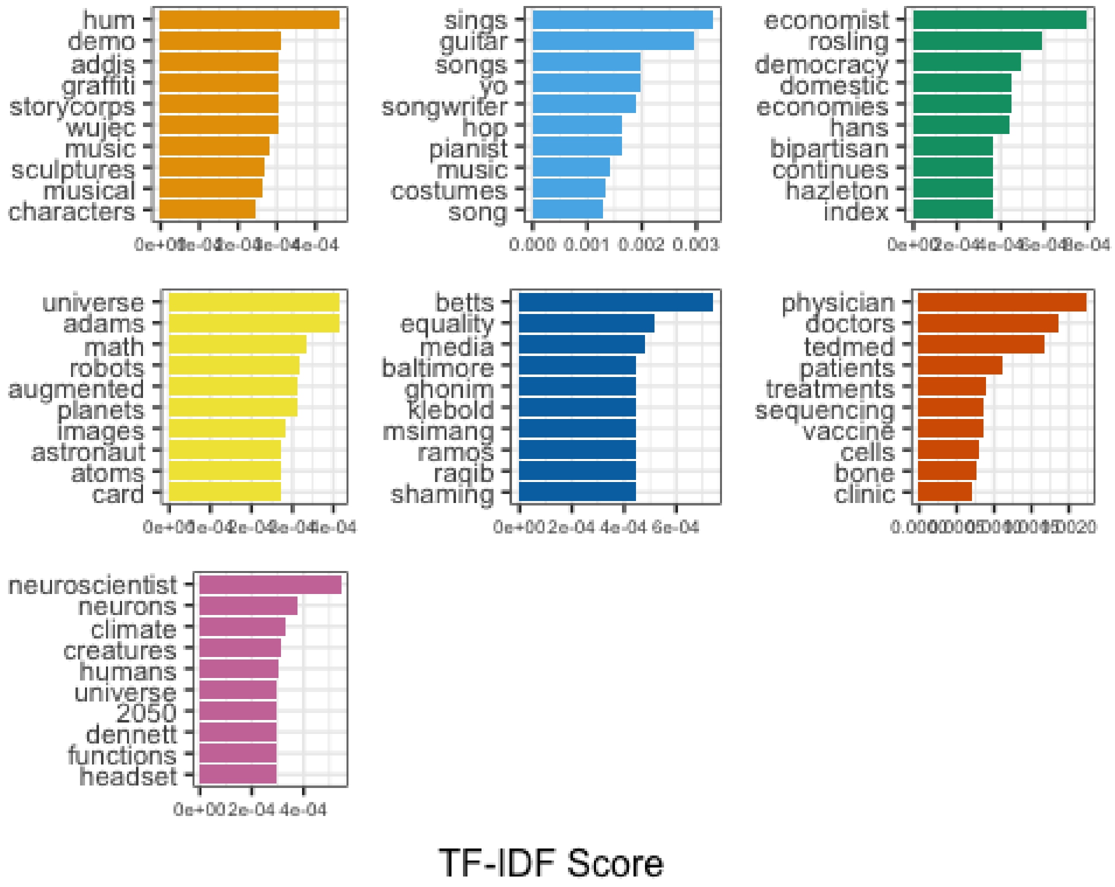
```
# most important words in descriptions
description_words.cluster <- ted_data %>%
  select(description, talk_id) %>%
  left_join(select(clusters_final.all, cluster.split1, talk_id), by='talk_id') %>%
  mutate(cluster=cluster.split1) %>%
  unnest_tokens(word, description) %>%
  count(cluster, word, sort = TRUE) %>%
  group_by(cluster) %>%
  mutate(cluster_total = sum(n)) %>%
  ungroup()

# word appearance by clusters
ggplot(description_words.cluster, aes(n/cluster_total, fill=cluster)) +
  geom_histogram(show.legend = FALSE, bins=50) +
  xlim(NA, 0.0025) +
  facet_wrap(~cluster, ncol=2, scales="free") +
  theme_bw() +
  scale_fill_manual(labels=c('1', '2', '3', '4', '5', '6', '7'),
                    values=c("#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2",
                             "#D55E00", "#CC79A7")) +
  labs(title='Word Frequencies by Tag Cluster',
       x='Word Frequency to Total Words Ratio',
       y=NULL)
```



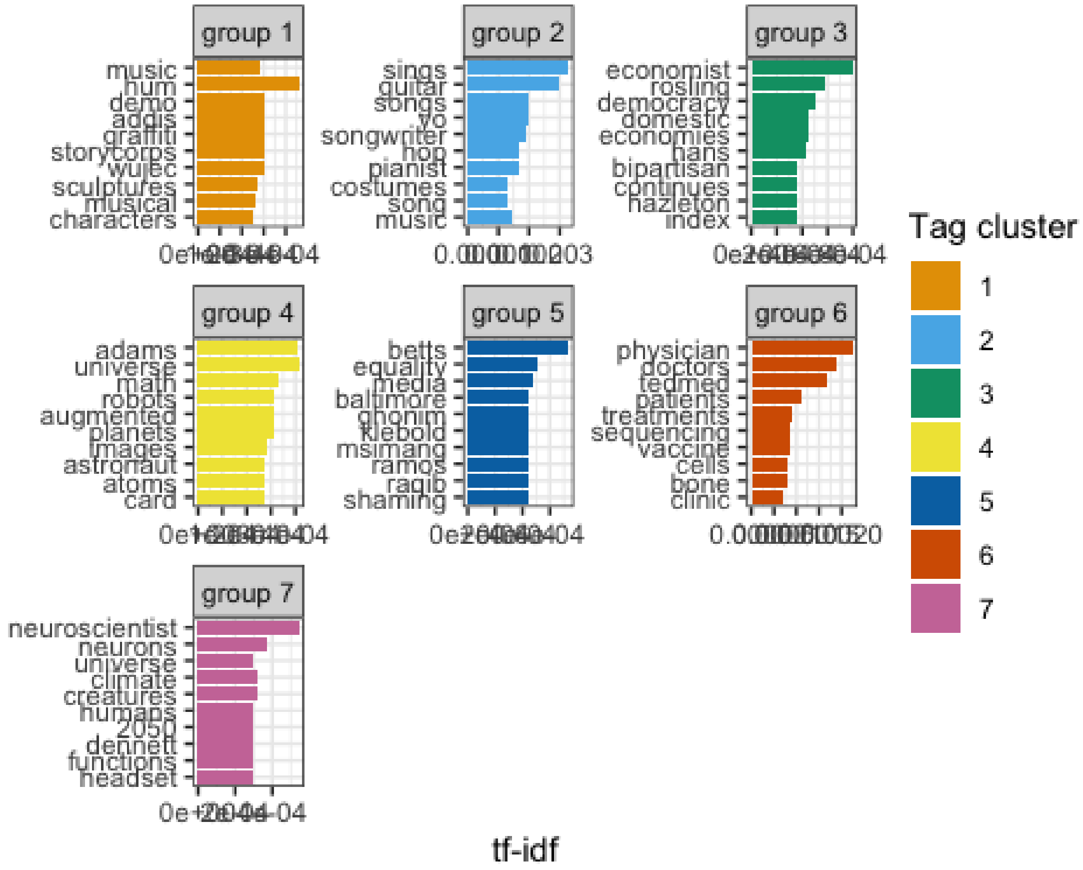
```
# Zipf's law states that the frequency that a word appears is inversely proportional to its
rank
# Check for this relationship by visualising the data
freq_by_rank <- description_words.cluster %>%
  group_by(cluster) %>%
  mutate(rank = row_number(),
         term_frequency = n/cluster_total)

# fit power law line
pl <- lm(log10(term_frequency) ~ log10(rank), data=freq_by_rank)

# plot frequency, rank and power law line to check if data is typical
freq_by_rank %>%
  ggplot(aes(rank, term_frequency, colour = cluster)) +
  geom_abline(intercept=pl$coefficients[1], slope=pl$coefficients[2], color = "gray50", lin
etype = 2) +
  geom_line(size=0.5)+
  scale_x_log10() +
  scale_y_log10() +
  theme_bw() +
  scale_colour_manual(labels=c('1', '2', '3', '4', '5', '6', '7'),
                      values=c("#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2",
                               "#D55E00", "#CC79A7")) +
  labs(title='Word Rank by Frequency',
       x='Word Rank',
       y='Term Frequency',
       colour='Tag cluster')
```



```
# the relationship is not linear at the extreme ends, but overall it seems to fit
# the data appears mostly typical, so the analysis can continue


# The higher the td_idf the more important the word in a certain document;
# tf-idf takes into account words that appear frequently in one but not across talks
description_words.cluster.bound <- description_words.cluster %>%
  bind_tf_idf(word, cluster, n) %>%
  select(-cluster_total) %>%
  arrange(cluster, -tf_idf) %>%
  mutate(cluster2 = cluster[6])

# visualise most important words
wordfreq_by_cluster <- function(n_cluster, top_words) {

  colour_list = c("#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A
7")
  cluster_no = as.numeric(str_sub(n_cluster, -1))
  my_colour =  colour_list[cluster_no]

  temp_plot <- description_words.cluster.bound %>%
    filter(cluster == n_cluster) %>%
    mutate(word = factor(word, levels = rev(unique(word)))) %>%
    group_by(cluster) %>%
    top_n(top_words, tf_idf) %>%
    top_n(top_words, word) %>% # added as many tf_idf scores had the same value, to limit t
he graph to 10 words
    arrange(desc(tf_idf)) %>%
    ungroup() %>%
    ggplot(aes(x=reorder(word, tf_idf), y=tf_idf, fill=cluster)) +
    geom_col(show.legend = FALSE, fill=my_colour) +
    labs(x = NULL, y = "tf-idf") +
    coord_flip() +
    theme_bw() +
    labs(x=NULL, y=NULL) +
    theme(axis.text.x = element_text(size = 6))
  return(temp_plot)
}

p1 <- wordfreq_by_cluster('group 1', 10)
p2 <- wordfreq_by_cluster('group 2', 10)
p3 <- wordfreq_by_cluster('group 3', 10)
p4 <- wordfreq_by_cluster('group 4', 10)
p5 <- wordfreq_by_cluster('group 5', 10)
p6 <- wordfreq_by_cluster('group 6', 10)
p7 <- wordfreq_by_cluster('group 7', 10)

grid.arrange(p1, p2, p3, p4, p5, p6, p7, nrow=3, ncol=3,
             bottom='TF-IDF Score')
```



```
# The following code does not reorder properly, so other version was used
description_words.cluster.bound %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  group_by(cluster) %>%
  top_n(10, tf_idf) %>%
  top_n(10, word) %>%
  ungroup() %>%
  ggplot(aes(x=reorder(word, tf_idf), y=tf_idf, fill=cluster)) +
  geom_col() +
  labs(x = NULL, y = "tf-idf") +
  coord_flip() +
  theme_bw()  +
  scale_fill_manual(labels=c('1', '2', '3', '4', '5', '6', '7'),
                    values=c("#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2",
                             "#D55E00", "#CC79A7")) +
  labs(fill='Tag cluster') +
  facet_wrap(~cluster, scales='free')
```



Section 5.1

Sentiment Analysis

```
## prep transcript data

ted_corpus <- VCorpus(VectorSource(ted_transcripts$transcript)) %>% # convert to corpus
  tm_map(removeWords, stopwords("english")) %>% # clean up
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace)

ted_txt <- tidytext::tidy(ted_corpus) %>%
  mutate(talk_id = ted_transcripts$talk_id) %>%
  mutate(total_words = str_count(text, " ") + 1)
```

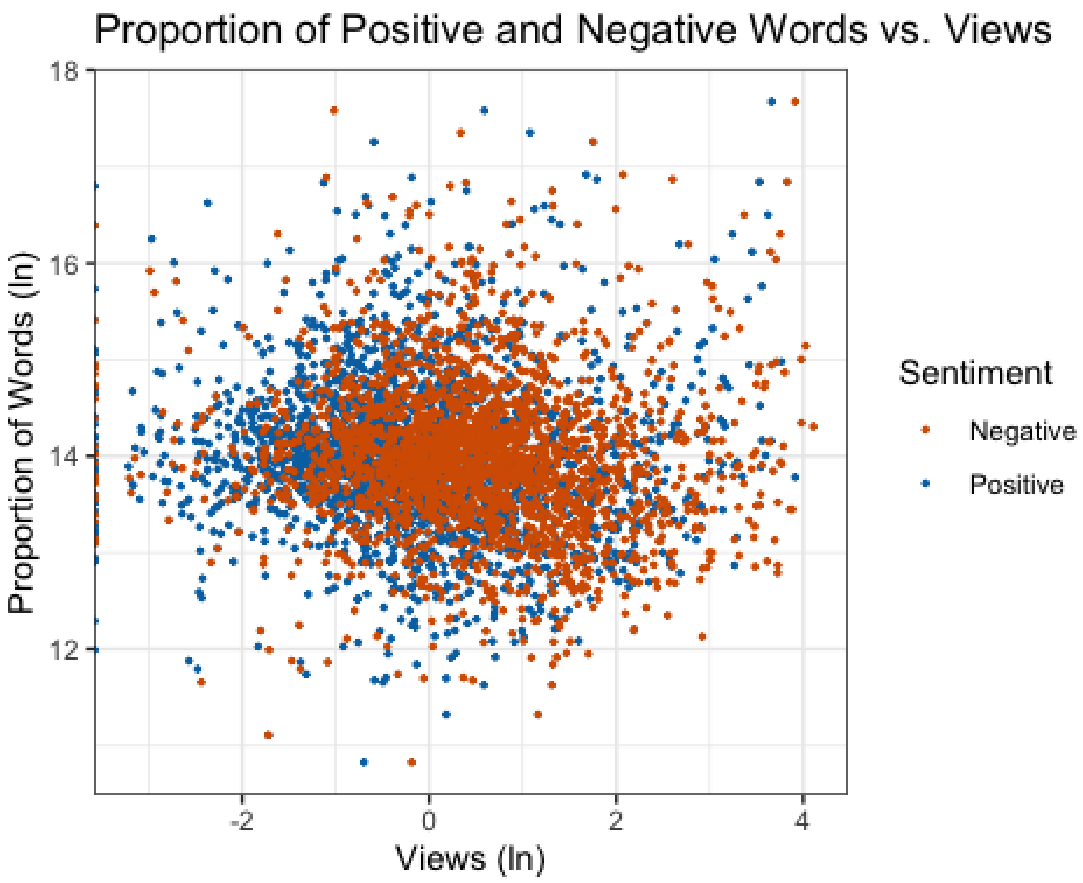
LOUGHRAN: calculate occurence various sentiments

```
##### LOUGHRAN

get_sentiment_loughran <- function(row_n){
  temp_txt <- ted_txt[row_n,]
  tidy_temp_text <- temp_txt %>%
    select(text, id) %>%
    group_by(id) %>%
    unnest_tokens(word, text) %>%
    ungroup() %>% # you can experiment with not including this when you compute count()
    anti_join(stop_words, by='word')
  temp_result <- tidy_temp_text %>%
    inner_join(get_sentiments("loughran"), by='word') %>%
    count(id, sentiment, word) %>%
    ungroup() %>%
    group_by(sentiment) %>%
    summarize(words = sum(n)) %>%
    mutate(id = row_n)
  return(temp_result)
}

# run on first talk to create df
ted_sentiment <- get_sentiment_loughran(1)
# run across all other talks and append to df
for (i in c(2:as.integer(count(ted_txt)))){
  data_temp <- get_sentiment_loughran(i)
  ted_sentiment <- rbind(ted_sentiment, data_temp)
}

ted_sentiment <- ted_sentiment %>%
  spread(sentiment, words)

ted_sentiment[is.na(ted_sentiment)] <- 0

ted_sentiment.loughran <- (ted_sentiment / rowSums(ted_sentiment) * 100) %>%
  mutate(talk_id = ted_sentiment$id) %>%
  select(-id)
```

Section 5.2

Explore sentiments

```
# plot sentiment and views

ted_sentiment.loughran %>% gather('sentiment', 'count', c(positive, negative)) %>%
  left_join(ted_data, by='talk_id') %>%
  ggplot(aes(x=log(count), y=log(views), colour=sentiment)) +
  geom_jitter(size=0.5) +
  scale_colour_manual(labels=c('Negative', 'Positive'),
                      values=c("#D55E00", "#0072B2")) +
  theme_bw() +
  labs(title='Proportion of Positive and Negative Words vs. Views',
       x='Views (ln)', y='Proportion of Words (ln)',
       colour='Sentiment')
```



```
# look at sentiment by clusters (tags, hierarchical)
ted_sentiment.loughran %>%
  left_join(select(clusters_final.all, cluster.split1, talk_id), by='talk_id') %>%
  gather(sentiment, sent_score, c(1:6)) %>%
  group_by(cluster.split1, sentiment) %>%
  mutate(sent_mean = mean(sent_score)) %>%
  ungroup() %>%
  select(-sent_score, -talk_id) %>%
  distinct() %>%
  spread(sentiment, sent_mean) %>%
  pander()
```

Table continues below

cluster.split1	constraining	litigious	negative	positive	superfluous
group 1	0.276	0.3032	3.235	2.055	0.01866
group 2	0.357	0.4157	4.735	2.951	0.02717
group 3	0.2638	0.3634	3.756	2.103	0.01342
group 4	0.206	0.2825	3.18	1.792	0.01526
group 5	0.23	0.2856	3.082	1.594	0.02059
group 6	0.1658	0.251	2.479	1.239	0.008341
group 7	0.3018	0.3604	3.682	2.402	0.02043

uncertainty
0.6215
0.8746

0.7072

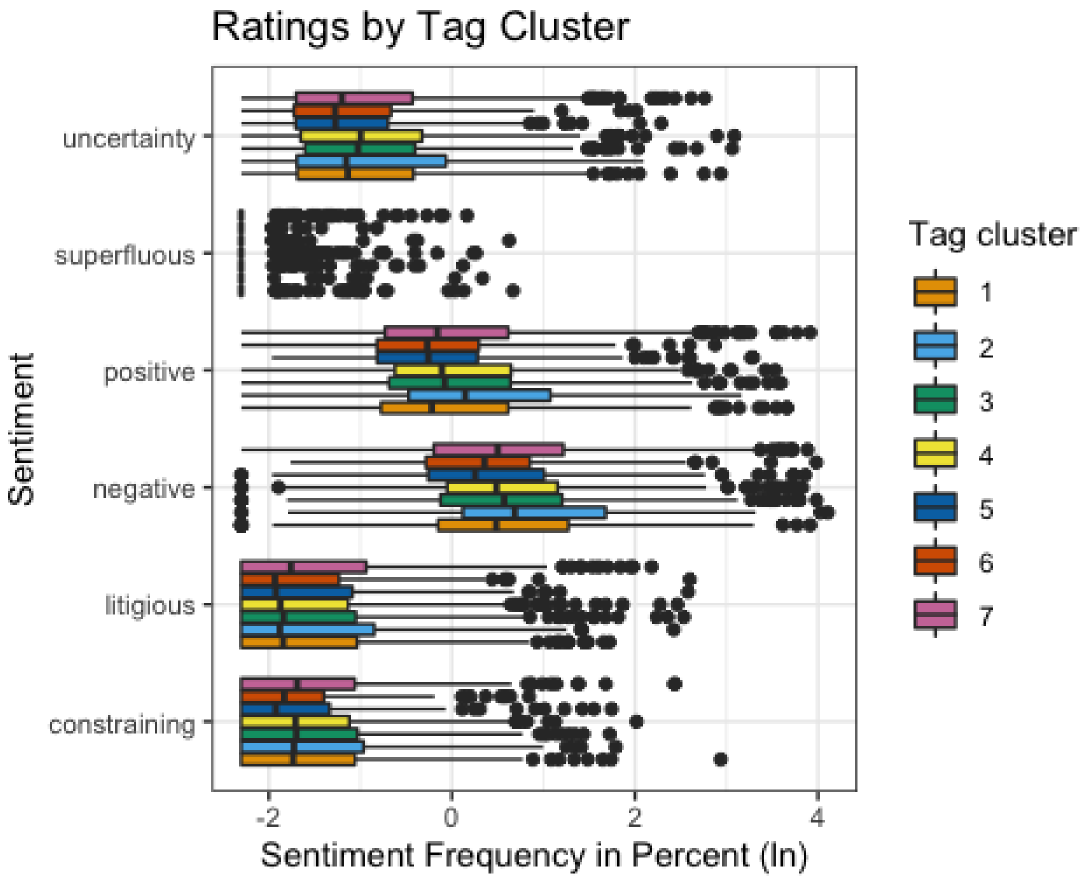
0.6344

0.4912

0.4286

0.7153

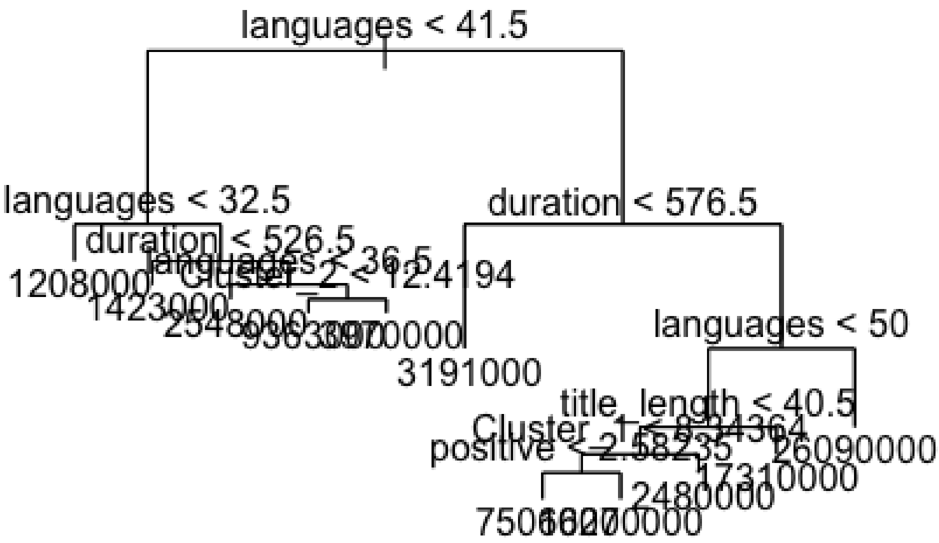
```
ted_sentiment.loughran %>%
  gather('sentiment', 'percent', c(1:6)) %>%
  left_join(select(clusters_final.all, cluster.split1, talk_id), by='talk_id') %>%
  ggplot(aes(x=sentiment, y=log(percent+0.1), fill=cluster.split1)) +
  geom_boxplot() +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  coord_flip() +
  theme_bw() +
  scale_fill_manual(labels=c('1', '2', '3', '4', '5', '6', '7'),
                    values=c("#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2",
                             "#D55E00", "#CC79A7")) +
  labs(title = 'Ratings by Tag Cluster',
       y='Sentiment Frequency in Percent (ln)',
       x='Sentiment',
       fill='Tag cluster')
```



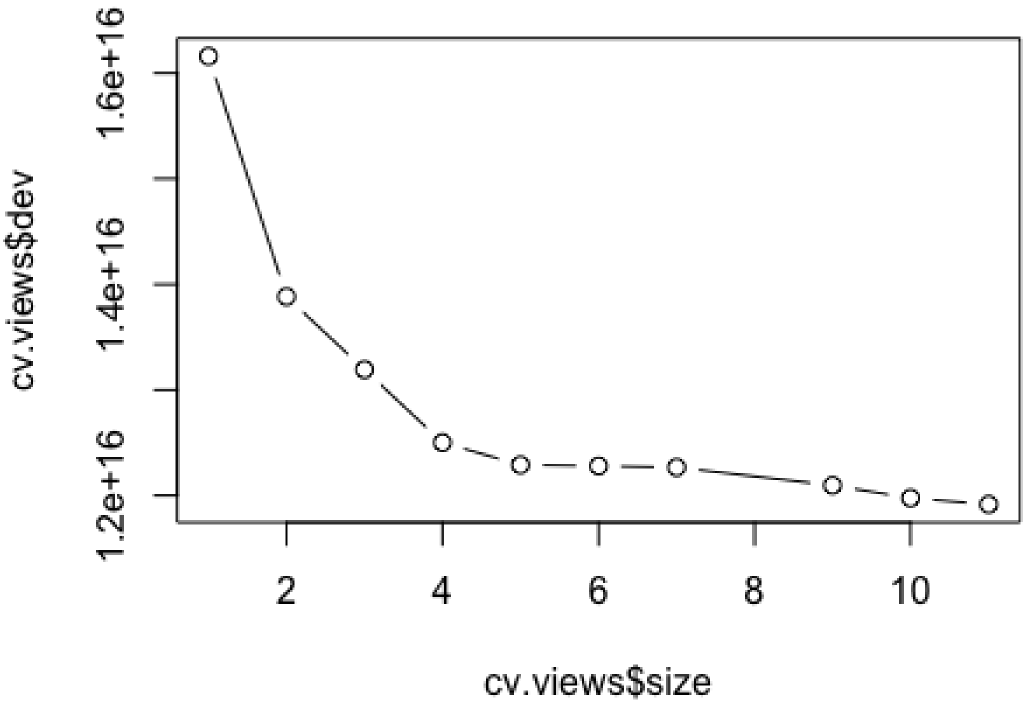
Section 6.1

Decision Trees and Random Forests

```
clusters <- cluster_lookup

rf_data <- clusters_final.all %>%
  left_join(ratings_percent, by='talk_id') %>%
  left_join(ted_sentiment.loughran, by='talk_id')

data <- rf_data %>%
  gather(characteristic, value, c(28:41)) %>%
  left_join(clusters, by='characteristic') %>%
  select(-characteristic) %>%
  filter(!is.na(cluster_pca)) %>%
  group_by(talk_id, cluster_pca) %>%
  mutate(value=sum(value)) %>%
  distinct() %>%
  spread(cluster_pca, value) %>%
  ungroup()

my_vars.views <- c('views', 'cuts', 'published_year', 'languages', 'duration',
             'Cluster_1', 'Cluster_2', 'Cluster_3', 'Cluster_4', 'description_length',
             'title_length', 'negative', 'positive')

my_data <- select(data, my_vars.views) %>%
  mutate(cuts = paste('Group', cuts, sep='_'),
         dummy = 1) %>%
  spread(cuts, dummy)
my_data[is.na(my_data)] <- 0

set.seed(2)
tree.views <- tree::tree(views~., my_data)

summary(tree.views)

##
## Regression tree:
## tree::tree(formula = views ~ ., data = my_data)
## Variables actually used in tree construction:
## [1] "languages"    "duration"     "Cluster_2"    "title_length"
## [5] "Cluster_1"    "positive"
## Number of terminal nodes:  11
## Residual mean deviance:  2.975e+12 = 7.553e+15 / 2539
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -24460000   -600700   -215600         0    245000  21140000

tree.views

## node), split, n, deviance, yval
##       * denotes terminal node
##
##   1) root 2550 1.591e+16  1698000
##     2) languages < 41.5 2424 5.456e+15  1464000
##       4) languages < 32.5 1902 1.764e+15  1208000 *
##       5) languages > 32.5 522 3.116e+15  2394000
##        10) duration < 526.5 222 2.173e+14  1423000 *
##        11) duration > 526.5 300 2.534e+15  3113000
##          22) languages < 36.5 211 1.047e+15  2548000 *
##          23) languages > 36.5 89 1.259e+15  4455000
##            46) Cluster_2 < 12.4194 8 4.120e+14  9363000 *
##            47) Cluster_2 > 12.4194 81 6.350e+14  3970000 *
##     3) languages > 41.5 126 7.755e+15  6212000
##       6) duration < 576.5 79 5.671e+14  3191000 *
##       7) duration > 576.5 47 5.254e+15 11290000
##        14) languages < 50 42 2.313e+15  9529000
##          28) title_length < 40.5 36 1.788e+15  8232000
##            56) Cluster_1 < 8.34364 29 1.488e+15  9620000
##             112) positive < 2.58235 22 6.000e+14  7506000 *
##             113) positive > 2.58235 7 4.810e+14 16270000 *
##            57) Cluster_1 > 8.34364 7 1.227e+13  2480000 *
##          29) title_length > 40.5 6 1.004e+14 17310000 *
##        15) languages > 50 5 1.717e+15 26090000 *

plot(tree.views)
text(tree.views, pretty=0)
```



```
# cv consideres the number of end nodes!
cv.views <- cv.tree(tree.views)
plot(cv.views$size, cv.views$dev, type='b')
```

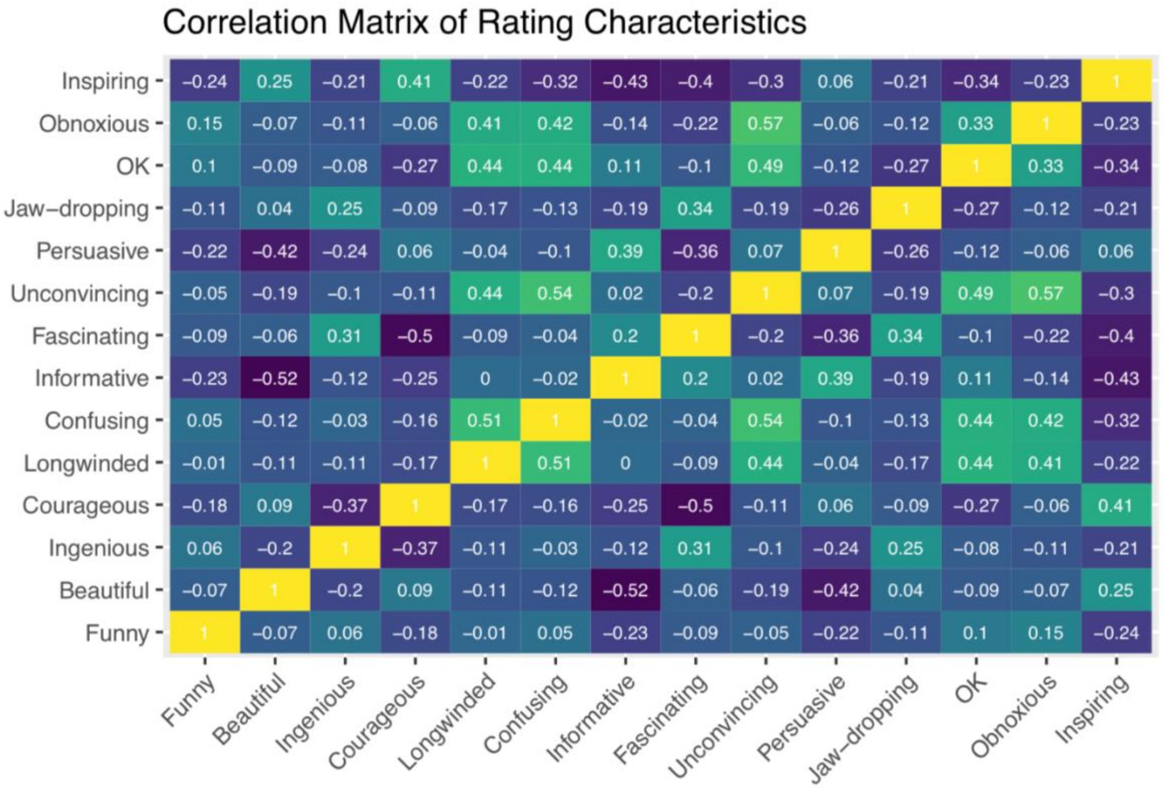


```
train <- sample(1:nrow(my_data), nrow(my_data)/1.25)
tree.views <- tree(views~., data=my_data, subset=train)
summary(tree.views)

##
## Regression tree:
## tree(formula = views ~ ., data = my_data, subset = train)
## Variables actually used in tree construction:
## [1] "languages"          "duration"           "Cluster_2"
## [4] "published_year"     "title_length"       "Cluster_1"
## [7] "description_length"
## Number of terminal nodes:  14
## Residual mean deviance:  2.464e+12 = 4.992e+15 / 2026
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -12420000   -598100   -214900         0    254300  21420000

views.test <- my_data[-train, 'views']

# random forest
rf.views <- randomForest(views~., data=my_data, subset=train,  mtry=8, importance=TRUE)
yhat.rf <- predict(rf.views, newdata=my_data[-train,])
mean((yhat.rf-views.test$views)^2)

## [1] 3.010201e+12

mean((abs(yhat.rf-views.test$views)/yhat.rf)*100)

## [1] 38.35393

importance(rf.views)
```

```
##                       %IncMSE IncNodePurity
## published_year     10.4758917  3.828372e+14
## languages          37.2688913  3.869477e+15
## duration           18.3666586  1.756402e+15
## Cluster_1           8.2421969  1.253642e+15
## Cluster_2           5.0579237  5.577106e+14
## Cluster_3           4.0487132  4.200747e+14
## Cluster_4           5.2843550  7.406531e+14
## description_length  8.4763835  9.154057e+14
## title_length        3.3525559  4.718764e+14
## negative            3.2516475  5.995647e+14
## positive            3.6000232  6.291466e+14
## Group_1             0.2136327  4.149603e+13
## Group_2             0.5744950  2.060741e+13
## Group_3             2.0542918  3.869202e+13
## Group_4             1.5334005  1.633980e+14
## Group_5             1.3027663  5.835607e+13
## Group_6             1.6784638  3.889076e+13
```

Appendix B. Supplementary Rating Visualisations.

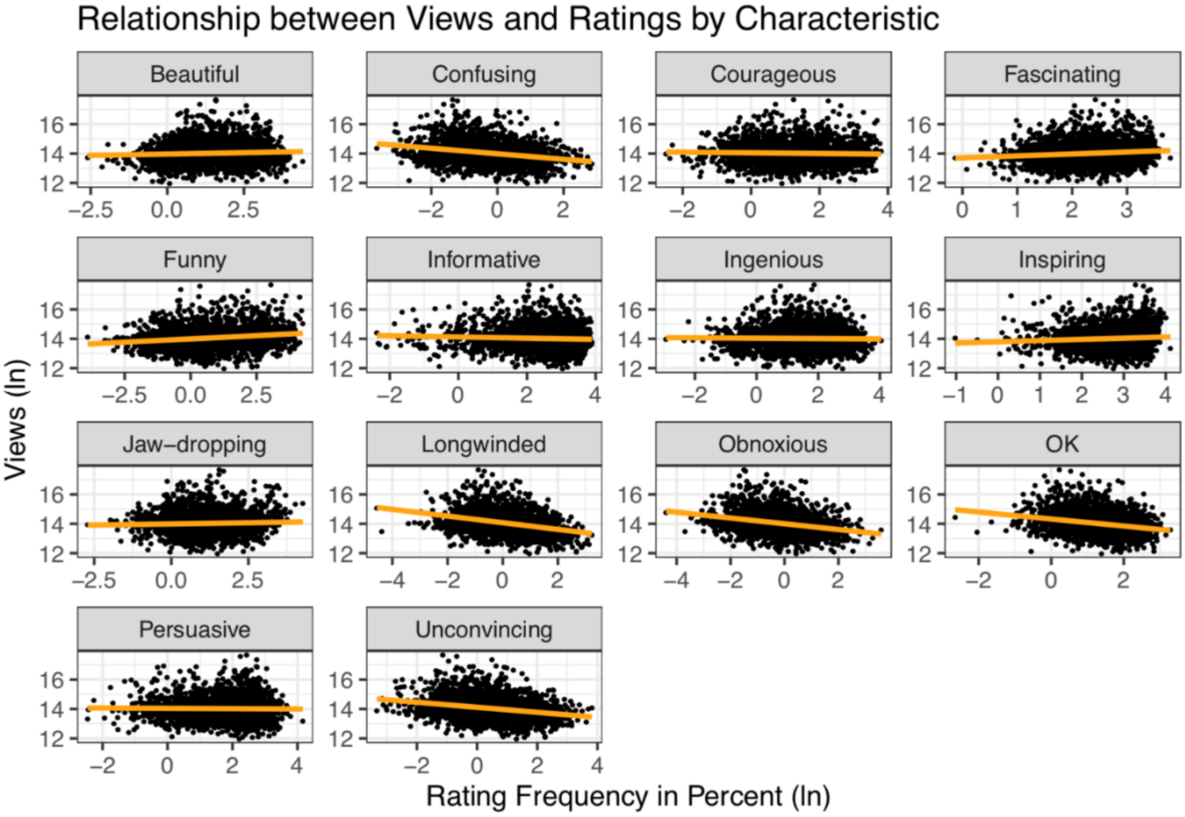
a)



Correlation Matrix of Rating Characteristics

	Funny	Beautiful	Ingenious	Courageous	Longwinded	Confusing	Informative	Fascinating	Unconvincing	Persuasive	Jaw−dropping	OK	Obnoxious	Inspiring
Inspiring	−0.24	0.25	−0.21	0.41	−0.22	−0.32	−0.43	−0.4	−0.3	0.06	−0.21	−0.34	−0.23	1
Obnoxious	0.15	−0.07	−0.11	−0.06	0.41	0.42	−0.14	−0.22	0.57	−0.06	−0.12	0.33	1	−0.23
OK	0.1	−0.09	−0.08	−0.27	0.44	0.44	0.11	−0.1	0.49	−0.12	−0.27	1	0.33	−0.34
Jaw−dropping	−0.11	0.04	0.25	−0.09	−0.17	−0.13	−0.19	0.34	−0.19	−0.26	1	−0.27	−0.12	−0.21
Persuasive	−0.22	−0.42	−0.24	0.06	−0.04	−0.1	0.39	−0.36	0.07	1	−0.26	−0.12	−0.06	0.06
Unconvincing	−0.05	−0.19	−0.1	−0.11	0.44	0.54	0.02	−0.2	1	0.07	−0.19	0.49	0.57	−0.3
Fascinating	−0.09	−0.06	0.31	−0.5	−0.09	−0.04	0.2	1	−0.2	−0.36	0.34	−0.1	−0.22	−0.4
Informative	−0.23	−0.52	−0.12	−0.25	0	−0.02	1	0.2	0.02	0.39	−0.19	0.11	−0.14	−0.43
Confusing	0.05	−0.12	−0.03	−0.16	0.51	1	−0.02	−0.04	0.54	−0.1	−0.13	0.44	0.42	−0.32
Longwinded	−0.01	−0.11	−0.11	−0.17	1	0.51	0	−0.09	0.44	−0.04	−0.17	0.44	0.41	−0.22
Courageous	−0.18	0.09	−0.37	1	−0.17	−0.16	−0.25	−0.5	−0.11	0.06	−0.09	−0.27	−0.06	0.41
Ingenious	0.06	−0.2	1	−0.37	−0.11	−0.03	−0.12	0.31	−0.1	−0.24	0.25	−0.08	−0.11	−0.21
Beautiful	−0.07	1	−0.2	0.09	−0.11	−0.12	−0.52	−0.06	−0.19	−0.42	0.04	−0.09	−0.07	0.25
Funny	1	−0.07	0.06	−0.18	−0.01	0.05	−0.23	−0.09	−0.05	−0.22	−0.11	0.1	0.15	−0.24

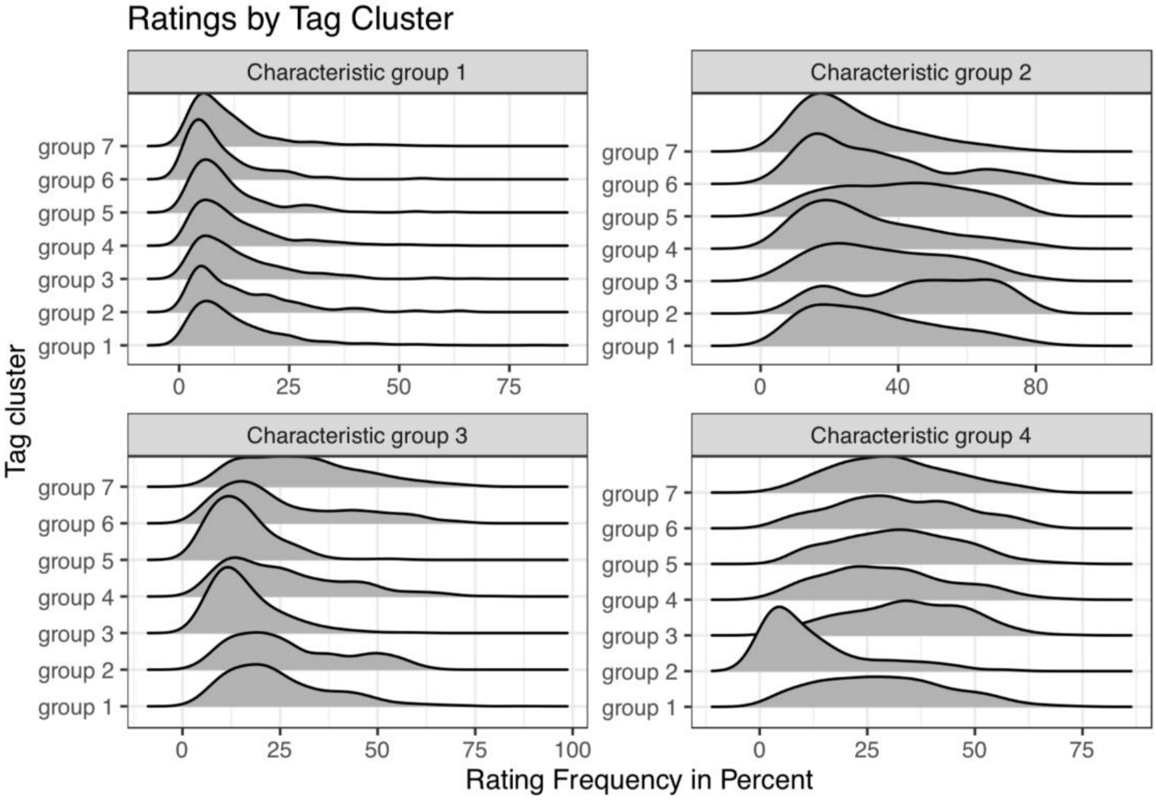
b)



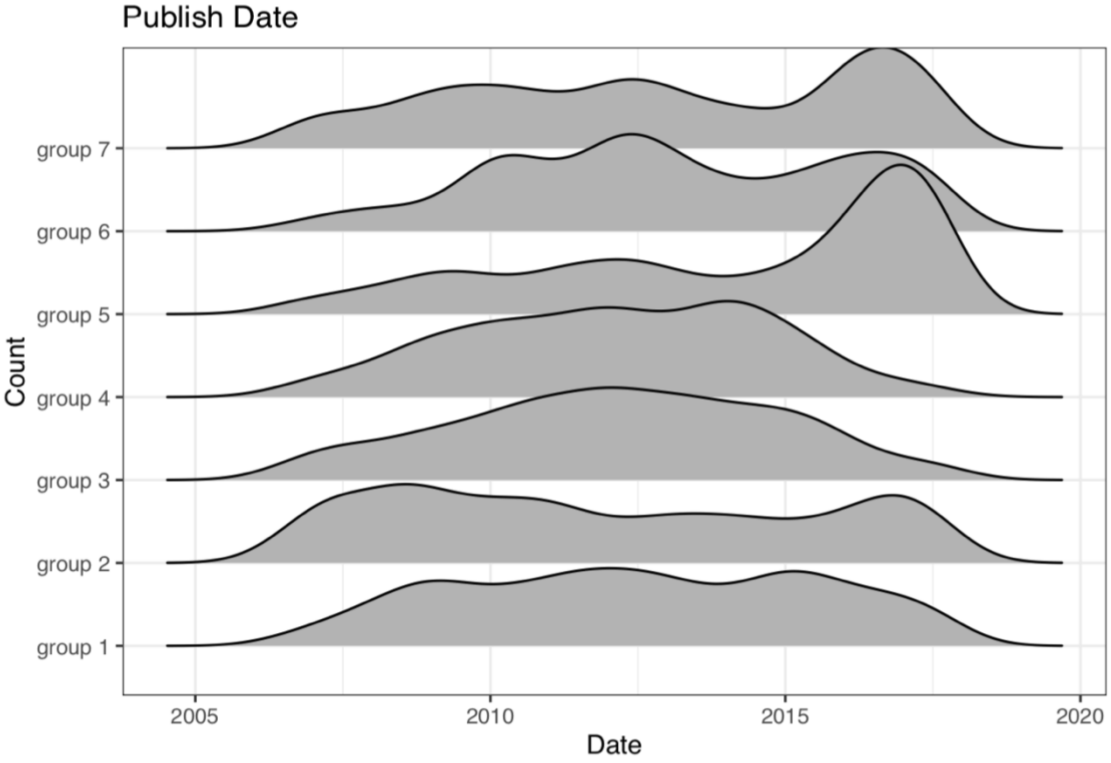
Relationship between Views and Ratings by Characteristic

Appendix C. Supplementary Tag Cluster Visualisations.

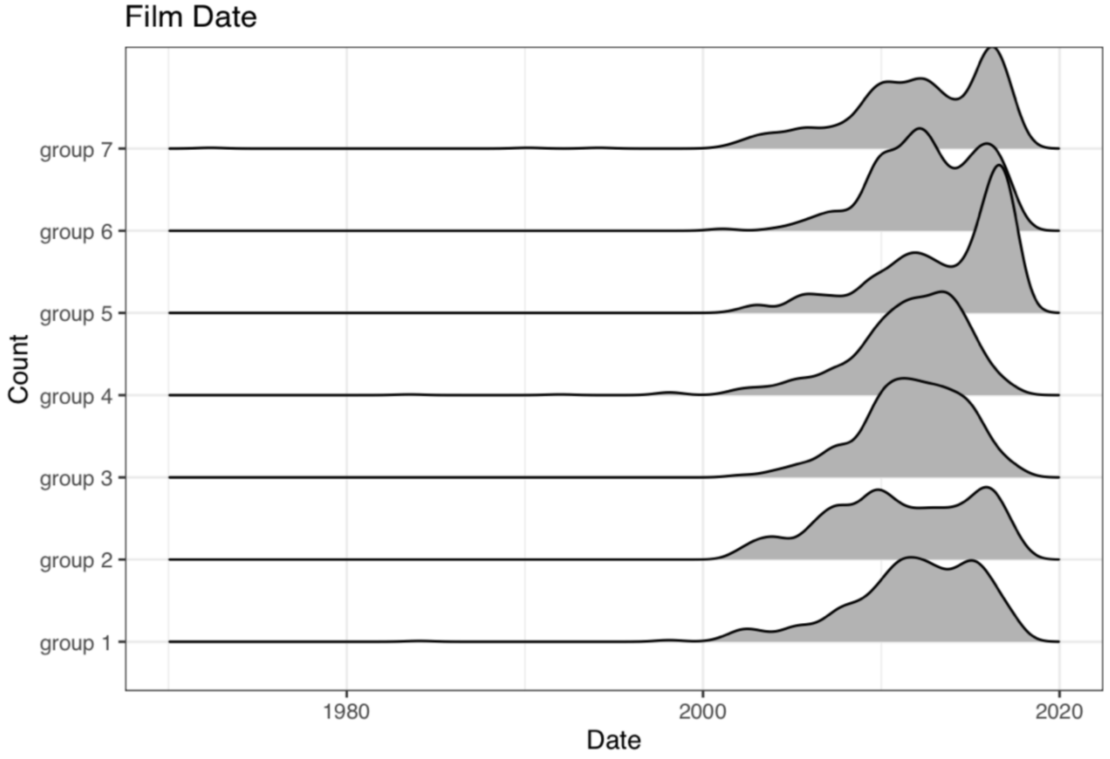
a)



b)



c)


Film Date
Count
group 7
group 6
group 5
group 4
group 3
group 2
group 1
1980
2000
2020
Date