

Data Mining Project - Movie review analysis

*Adrian Skoczkowski
Norbert Nieżorawski*

1. Introduction

1.1 Abstract

The film industry is one where the opinion of customers is very closely tied to the success of a film. From film critics publishing reviews before the wide release of a movie, to word-of-mouth judgment passed around by moviegoers after it's in theaters, opinions of others contribute greatly to our own decision whether to see a given film. Of course, it is the number of tickets sold that matters to the studios, but indirectly it is the customer "consensus" that really matters. For any movie studio, it would be beneficial and profitable to keep up with customer sentiments concerning their previous films, as well as what qualities of the films the customer base paid most attention to. This can be achieved through online polls, centralized in so called "rating sites", where users can give a movie points or "stars", rating it according to their enjoyment. Moreover, most of these sites enable the user to publish written reviews, and thus to state their praises or complaints clearly. These written reviews will be the primary concern of this study. We have acquired a dataset of almost 5000 written reviews published on the most popular internet rating site, IMDB.com. IMDB was established in 1990 and as of 2021 reports to have over 250 million unique monthly visitors. For many moviegoers, the IMDB score presents an "objective" critical consensus and can oftentimes determine whether a person will watch a film or not. The amount of plain "star" reviews on IMDB is vastly greater than the number of written reviews (they make up only roughly 0.1% of total reviews), hence we were also interested whether sentiment analysis of the written reviews in our database will show similar trends to those outlined by plain "star" reviews. As the focal point of our study, we have chosen to examine and test the children's animated movies of the 2000s - 15 films from 3 different studios: Disney, Fox and DreamWorks. We selected such a dataset based on a few criteria. Studios from this "era" of animation still had very different styles and themes, there was room for experimentation and great variety, despite targeting the same audience. Some films in the dataset are sequels of others, some are only vaguely similar (e.g. showcasing talking animals), and we saw an opportunity for testing if our data analysis algorithms could also find commonalities that are obvious to anyone who watched the films, but would be quite obscure if the input is not even the synopsis, but just reviews concerning the synopsis indirectly. In the study, we're concerned with finding the customer engagement levels, general attitude towards the films, what topics are mentioned most often in the reviews and which films are the most similar. The results of our study may show advantages and disadvantages of this approach to gauging customer interest. With the insight thus gathered, we hope to provide conclusions that will prove useful to the studios we analyze, and others in the industry.

1.2 Research Methodology

In this study we have examined:

- Frequency of words in film using the wordcloud package
- Associations with the most frequent words in given film using findAssoc() function
- Topic number using Arun2010, CaoJuan2009 and Griffis methods
- Topic content in films using LATENT DIRICHLET ALLOCATION algorithm
- Sentiment analysis on topics, films, industries determining number of positive and negative words in a given comment
- Degree of Engagement on topics, films, industries
- Clustering on films and topics using hierarchical dendrogram

1.3 Definitions Used in the Project

Engagement Index (EI) and Popularity-Adjusted Engagement Index (PAEI)

PAEI is a heuristic to measure customer engagement that combines the analysis of review number (movie popularity) and review length (level of audience engagement). The best outcome for a movie is to not only be seen by many people, but also generate lasting impressions in that audience, thus facilitating further consumption of company products. Many low-quality (short) reviews may point to customer sentiment, but not necessarily engagement. Engagement index values of lower than 1 would indicate an increasingly “niche classic” outlook on the movie, while values higher than one an increasingly “forgettable blockbuster” view. Values close to 1 would indicate that the level of viewership corresponds to the level of engagement. This is valuable insight for further analysis, since for example movies being a part of the “cult classic” phenomenon may rank high in sentiment analysis, but not be fit for wider distribution or promotion. For the highest PAEI score, the movie has to have many reviews and an accordingly high number of words per review.

1.4 Research Questions

- Can customer sentiment be accurately assessed based on written reviews?
- Can clustering point to similarities between movies, even if they belong to different studios?
- What are the topics that the reviewers pay most attention to?
- Can customer engagement be assessed based on factors tied to written reviews?

1.5 Research Plan

1. [Choosing the right source of database with opinions](#)
2. [Comments scraping from IMDB](#)
3. [Preprocessing of data](#)
4. [General analysis and calculating engagement](#)
5. [Ranking movies by customer engagement](#)
6. [Producing word clouds on frequency of the words in given film](#)
7. [Producing associations of the most common 3 words in film](#)
8. [Producing both hierarchical and cosine similarity clustering](#)
9. [Latent semantic analysis with 10 most frequent words in given topic](#)
10. [Assessing topic-based customer engagement](#)
11. [Topic-based clustering](#)
12. [Sentiment analysis for topics](#)
13. [Sentiment analysis for films](#)
14. [Sentiment analysis for studios](#)

2. Experiments preparation & results

2.1 Choosing the dataset

We have chosen IMDB.com as the source of our written reviews.

IMDb (an abbreviation of Internet Movie Database) is an online database of information related to films, television series, home videos, video games, and streaming content online – including cast, production crew and personal biographies, plot summaries, trivia, ratings, and fan and critical reviews. IMDb began as a fan-operated movie database on the Usenet group "rec.arts.movies" in 1990, and moved to the web in 1993. It is now owned and operated by IMDb.com, Inc., a subsidiary of Amazon. As of June 2021, the database contained some 8 million titles (including television episodes) and 10.4 million person records. Additionally, the site had 83 million registered users.¹

We deemed IMDB to be the most suitable for our project since it is the largest english-language database of movie reviews in the world, and the reviews themselves come from both casual moviegoers and film critics.

¹ <https://en.wikipedia.org/wiki/IMDb>

We have chosen to examine and test the children's animated movies of the 2000s - 15 films from 3 different studios: Disney, Fox and DreamWorks. We selected such a dataset based on a few criteria. Studios from this "era" of animation still had very different styles and themes, there was room for experimentation and great variety, despite targeting the same audience. Some films in the dataset are sequels of others, some are only vaguely similar (e.g. showcasing talking animals), and we saw an opportunity for testing if our data analysis algorithms could also find commonalities that are obvious to anyone who watched the films, but would be quite obscure if the input is not even the synopsis, but just reviews concerning the synopsis indirectly.

The films are:

- Lilo and Stitch
- Treasure planet
- Brother bear
- Chicken little
- Bolt
- Ice age
- Robots
- Ice age 2
- Ice age 3
- Rio
- Shrek
- Sindbad
- Shrek 2
- Madagascar
- Over the hedge

2.2 Comments scraping

The scraping of IMDB comments posed a unique challenge. Upon entering the IMDB site of a given movie and scrolling down, the user will be presented with only one written review. Upon clicking on the heading “User reviews”, a new page opens, this one containing 25 reviews. The user then has to scroll all the way to the bottom of the page to click “Load More”, thus loading another 25 reviews. It’s easy to see that manual downloading of 5000 reviews in this manner would be a lengthy task. However, scraping the comments via a bot was also problematic. The python package IMDbPY stores movie scores, production data etc. but only 25 reviews for each title. Eventually, we achieved the goal of scraping all the reviews automatically via the BeautifulSoup package. The exact method can be seen in [Appendix 1, “Review scraping code”](#). The reviews were saved in plain text format for preprocessing and conversion to csv.

2.3 Preprocessing of data

The preprocessing on data were made by removing first importing the text document and applying appropriate methods like:

- removing punctuation
- removing numbers
- removing special characters
- lowering text
- removing English stopwords
- Stripping whitespace
- Stemming

After that the files were saved in csv so that we could work separately on them. The exact code is in [Appendix 2: Preprocessing](#) section.

2.4 General overview

As mentioned in section 2.1, we have chosen 15 films from 3 different studios: Disney, Fox and DreamWorks, all from years 2001-2011. The numbers of reviews for each film varied, from just 92 (*Sindbad*) to 706 (*Shrek*). The number of total reviews across the three studios also varied, with Fox having the least (1334 reviews), Disney having only slightly more (1443 reviews), and with DreamWorks winning by a large margin (2078 reviews). Firstly, we analyzed the average word count of the reviews for each film. They ranged from ~84 words on average (*Shrek*) to ~129 words on average (*Ice age 3*). When it comes to studios, Disney had the shortest reviews on average with ~96 words, DreamWorks had ~100 words per review and Fox had ~112 words. The next step was to somehow derive customer engagement levels from the two aforementioned values: review count and length. The Engagement Index value was derived by taking two values: the percentage of total reviews belonging to the given movie and the percentage of total words belonging to the given movie, and dividing them. The use and meaning of the Engagement Index may be found in [section 1.3](#). The best (i.e. most closely approaching 1) Engagement Index was found for *Over the hedge* (EI = 1.002382), and the worst for *Ice age* (EI = 1.571707). Disney had the best EI out of all studios, and DreamWorks had the worst. All the information above is presented in Table 1.

Tab 1.

Title	Studio	Year of production	# of reviews	Average length of review [words]	Engagement index (EI) (% tot. reviews / % tot. words)
Lilo and Stitch	Disney	2002	421	92.28741	1.536162
Treasure planet	Disney	2002	309	86.47573	1.203266
Brother bear	Disney	2003	206	93.61165	0.7410283
Chicken little	Disney	2005	293	105.058	0.9391521
Bolt	Disney	2008	214	103.8692	0.6937843
TOTAL DISNEY			1443/ 288.6 (avg)	481.3019 / 96.260 (avg)	1.02267854 (avg)
Ice age	Fox	2002	423	90.62884	1.571707
Robots	Fox	2005	297	110.6801	0.9036172
Ice age 2	Fox	2006	259	113.4247	0.7689349
Ice age 3	Fox	2009	136	<u>129.4559</u>	0.3537648
Rio	Fox	2011	219	116.4612	0.6332284
TOTAL FOX			1334/ 266.8 (avg)	650.6507 / 112.130 (avg)	0.84625046 (avg)
Shrek	DreamWorks	2001	<u>706</u>	84.84561	2.80203
Sindbad	DreamWorks	2003	92	119.8043	0.2585907
Shrek 2	DreamWorks	2004	651	90.11521	2.432655
Madagascar	DreamWorks	2005	331	107.9547	1.032485
Over the hedge	DreamWorks	2006	298	100.1107	<u>1.002382</u>
TOTAL DW			2078/ 415.6 (avg)	502.8305 / 100.566 (avg)	1.50562854 (avg)
TOTAL ALL			4855/ 323.6 (avg)	1634.883 / 102.985 (avg)	1.124852513 (avg)

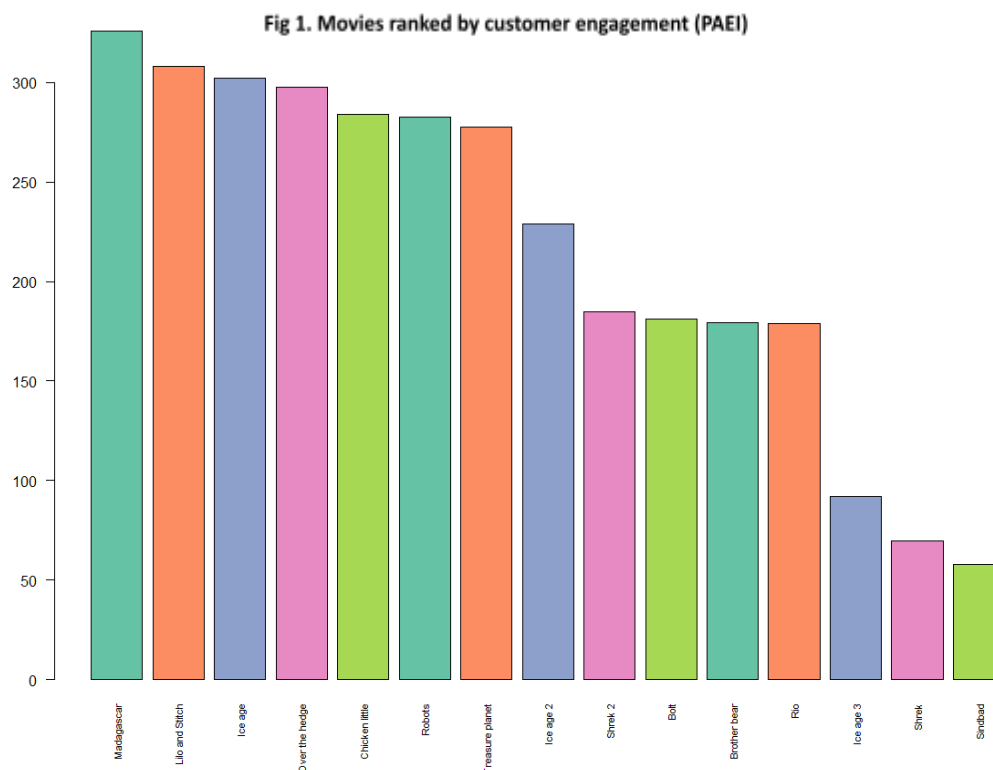
2.5 Ranking movies by customer engagement

Nextly, we calculated the Popularity-adjusted Engagement Index with the following formula: $PAEI = reviews - (0.5 * reviews * |1 - EI|)$; The 0.5 multiplier was arbitrarily chosen to present the best results - some multiplier values produced negative results, others were too punishing for movies with a smaller popularity. The 0.5 value was a good middle-ground, but further testing would be necessary to determine the objectively best formula for assessing customer engagement. The results of ranking the PAEI varied somewhat from the results of plain EI analysis: the highest score was achieved by *Madagascar* (PAEI = 325.6237), and the lowest by *Sindbad* (57.89517). Notably, the movie *Shrek*, which had gathered the most reviews out of any movie, was the second to last in the PAEI ranking - thus showcasing that higher review numbers don't necessarily represent higher customer engagement levels. The information above is presented in Table 2.

Tab 2.

Title	Popularity-adjusted engagement index (PAEI) $reviews - (0.5 * reviews * 1 - EI)$	Rank
Madagascar	325.6237	1
Lilo and Stitch	308.1379	2
Ice age	302.084	3
Over the hedge	297.6451	4
Chicken little	284.0858	5
Robots	282.6872	6
Treasure planet	277.5954	7
Ice age 2	229.0771	8
Shrek 2	184.6708	9
Bolt	181.2349	10
Brother bear	179.3259	11
Rio	178.8385	12
Ice age 3	92.05601	13
Shrek	69.88341	14
Sindbad	57.89517	15

This information was also represented visually in the bar chart in Figure 1.



2.6 Word clouds of frequent words in all films

Using the wordcloud package in R we managed to create visual representations of the 30 most frequent words used in the reviews of each film. The size of a given word represents how often it appeared - the more, the larger the word. Accordingly, the color of the word corresponds to the frequency of its use in the database. The results are presented in the Table 3 below.

Tab 3.

[illegible]

Ice age


human film
funni tiger make kid
mammoth watch
diego anim like
sid see sloth voic
good just ice
also littl stori realli
shrek get
babi great
charact age
even
movi



Ice age 2

sequel
elli think good
first make great meltdown like
diego manni get see ice
funni origin film
realli just
age still littl scrat
mammoth stori
charact
anim

Ice age 3

film movi
buck anim
funni first babi new good
diego dinosaur
like still kid elli much
famili scrat dawn
sid ice realli see
stori watch get just
charact manni
age

<p><i>Lilo and Stitch</i></p>	 <p>A word cloud for the movie 'Lilo and Stitch'. The most prominent words are 'stitch' in large yellow letters, 'movi' in large yellow letters, 'disney' in large green letters, 'film' in large pink letters, and 'charact' in large blue letters. Other visible words include 'sister', 'girl', 'stori', 'think', 'make', 'hawaiian', 'alien', 'good', 'see', 'just', 'kid', 'love', 'get', 'great', 'also', 'like', 'funni', 'even', 'much', 'famili', 'nani', and 'littl'.</p>
<p><i>Madagascar</i></p>	 <p>A word cloud for the movie 'Madagascar'. The most prominent words are 'madagascar' in large blue letters, 'charact' in large pink letters, 'funni' in large blue letters, 'penguin' in large blue letters, 'movi' in large yellow letters, and 'film' in large green letters. Other visible words include 'wild', 'even', 'kid', 'see', 'new', 'voic', 'realli', 'make', 'find', 'lion', 'friend', 'zoo', 'zebra', 'good', 'rock', 'just', 'get', 'watch', 'stori', 'alex', 'marti', 'great', and 'zoo'.</p>
<p><i>Over the hedge</i></p>	 <p>A word cloud for the movie 'Over the Hedge'. The most prominent words are 'charact' in large blue letters, 'hedge' in large blue letters, 'anim' in large yellow letters, 'good' in large blue letters, 'realli' in large blue letters, 'famili' in large blue letters, 'kid' in large blue letters, 'funni' in large blue letters, 'stori' in large blue letters, 'voic' in large blue letters, 'love' in large blue letters, 'make' in large blue letters, 'just' in large blue letters, 'raccoon' in large blue letters, 'willi' in large blue letters, 'enjoy' in large blue letters, 'see' in large blue letters, 'human' in large blue letters, 'hammi' in large blue letters, 'bear' in large blue letters, 'food' in large blue letters, 'film' in large green letters, and 'movi' in large yellow letters. Other visible words include 'also', 'great', 'watch', 'dreamwork', 'get', 'find', 'like', and 'zoo'.</p>

<p><i>Sindbad</i></p>	 <p>A word cloud for the movie 'Sindbad'. The most prominent words are 'sinbad' in large yellow letters, 'film' in pink, 'good' in blue, 'movi' in green, and 'anim' in pink. Other visible words include 'enjoy', 'character', 'make', 'voic', 'like', 'action', 'marina', 'stori', 'friend', 'proteus', 'great', 'disney', 'goddess', 'kid', 'love', 'book', 'dreamwork', 'eri', 'realli', 'see', 'sea', 'adventur', and 'book'.</p>
<p><i>Treasure planet</i></p>	 <p>A word cloud for the movie 'Treasure Planet'. The most prominent words are 'treasur' in pink, 'film' in green, 'disney' in green, 'movi' in yellow, 'charact' in pink, and 'planet' in blue. Other visible words include 'watch', 'good', 'look', 'even', 'much', 'still', 'also', 'silver', 'space', 'stori', 'see', 'get', 'love', 'island', 'john', 'realli', 'like', 'just', 'classic', and 'make'.</p>

The results of the word cloud visualizations are promising for further analysis - we see that often-mentioned words can mostly be categorized threefold:

plot-related - names of characters, locations, things appearing in the film

qualities - such as "animation", "story", "action" which indicate which qualities of the movie the viewers reacted to

emotions - such as "love", "enjoy", "good", primarily useful for sentiment analysis

2.7 Top 3 Associations of the 3 most frequent words

Table 4 below shows 3 most frequent words in a given movie and accordingly 3 words with the best correlation to each of these words with most frequency. [Code](#)

Tab 4.

Bolt	<div> <div>bolt penni believ show 0.71 0.59 0.57</div> <div>movi anim first meet 0.40 0.38 0.37</div> <div>film charact disney said 0.45 0.44 0.43</div> </div>
Brother bear	<div> <div>smovi begin escap chang 0.58 0.57 0.52</div> <div>bear brother kill kenai 0.78 0.71 0.67</div> <div>disney brother anim best 0.57 0.53 0.48</div> </div>
Chicken little	<div> <div>movi year make littl 0.34 0.32 0.29</div> <div>littl chicken sky father 0.85 0.60 0.59</div> <div>chicken littl sky fall 0.85 0.69 0.65</div> </div>
Ice age	<div> <div>film anim charact thisth 0.60 0.56 0.50</div> <div>anim film ice age 0.60 0.50 0.48</div> <div>movi afraid anachronist influenc 0.25 0.24 0.23</div> </div>
Ice age 2	<div> <div>movi skip watch sound 0.36 0.32 0.31</div> <div>ice age meltdown possum 0.94 0.65 0.52</div> <div>film filmmak howev charact 0.55 0.51 0.49</div> </div>

Ice age 3	<pre> \$movie agre purpos enough 0.65 0.58 0.56 ice age dawn dinosaur 0.97 0.69 0.59 age ice dawn dinosaur 0.97 0.67 0.56 </pre>
Lilo and Stitch	<pre> \$movie everyth see need 0.31 0.31 0.29 stitch lilo jumba nani 0.82 0.52 0.48 lilo stitch nani sister 0.82 0.59 0.52 </pre>
Madagascar	<pre> \$movie humor laugh even 0.35 0.33 0.32 anim madagascar natur zoo 0.52 0.45 0.45 film moment dreamwork madagascar 0.46 0.45 0.44 </pre>
Over the hedge	<pre> \$movie hit went prepar 0.31 0.26 0.25 anim hedg food film 0.61 0.54 0.47 \$film improv anim cast 0.49 0.47 0.45 </pre>
Rio	<pre> \$movie devilish select comedi 0.42 0.42 0.39 rio janeiro citi saldanha 0.60 0.59 0.53 \$anim rio charact film 0.49 0.47 0.44 </pre>
Robots	<pre> \$movie just girl kid 0.27 0.26 0.26 robot citi rodney ratchet 0.69 0.57 0.55 \$film robot oldfashion want 0.35 0.34 0.34 </pre>

Shrek	<div>shrek</div> <div>donkey 0.65 fiona 0.60 farquaad 0.59</div> <div>smovi</div> <div>like 0.34 lot 0.33 say 0.30</div> <div>film</div> <div>anim 0.44 prime 0.44 adapt 0.40</div>
Shrek 2	<div>shrek</div> <div>fiona 0.69 donkey 0.57 fairi 0.57</div> <div>\$movi</div> <div>watch funni first 0.31 0.27 0.26</div> <div>film</div> <div>headach 0.58 bonanza 0.52 ceochairman 0.52</div>
Sindbad	<div>sindbad</div> <div>artist 0.62 grab 0.62 imdb 0.62</div> <div>smovi</div> <div>play 0.63 say 0.62 tension 0.62</div> <div>film</div> <div>noth 0.65 make 0.64 wave 0.63</div>
Treasure planet	<div>movi</div> <div>creator 0.45 action 0.44 sake 0.44</div> <div>disney</div> <div>silver 0.47 planet 0.45 treasur 0.45</div> <div>film</div> <div>treasur 0.54 planet 0.49 jim 0.46</div>

We can conclude that the 3 most frequent words in film are usually main title or main character and the terms like film or movie. Quite often when the title does not portray the main character it shows only connotations different from the plot of the film. When the word describes the main character or something with a film title it associates mainly with the plot of the film. On the other hand when there is a word like film, movi, anim it shows emotions connected to the film.

2.8 Inter-corporal similarity clustering

Hierarchical clustering

In Figure 2 we see that the dendrogram algorithm correctly identified “Shrek” cluster, but the “Ice age” cluster is missing “Ice age 3”. “Madagascar” and “Over the hedge” placed close together, possibly because of the many similarities in the movies’ style and humor, coming from the same studio in similar years etc.

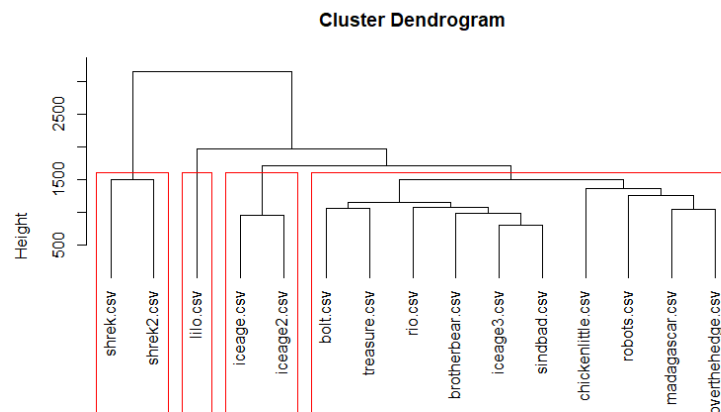


Fig 2.

Cosine similarity clustering

The graph (Fig 3.) below provides us with interesting insight into the types of reviews in our datasets, as well as similarities between reviews of different films. The clustering correctly identified the Shrek / Shrek 2 cluster and as we can see, it is an outlier compared to the rest of the dataset. We speculate that this may be due to numerous “joke” reviews of Shrek we encountered, and because other films have mostly serious reviews, Shrek will become more distant. Another outlier is “Lilo & Stitch”, but a closer inspection of the reviews would be needed to explain this occurrence. Further, we see that the “Ice age” movies have quite similar reviews, and the same can be said about “Treasure planet” and “Sindbad”, two movies that despite coming from different studios, have a similar animation style and tone.

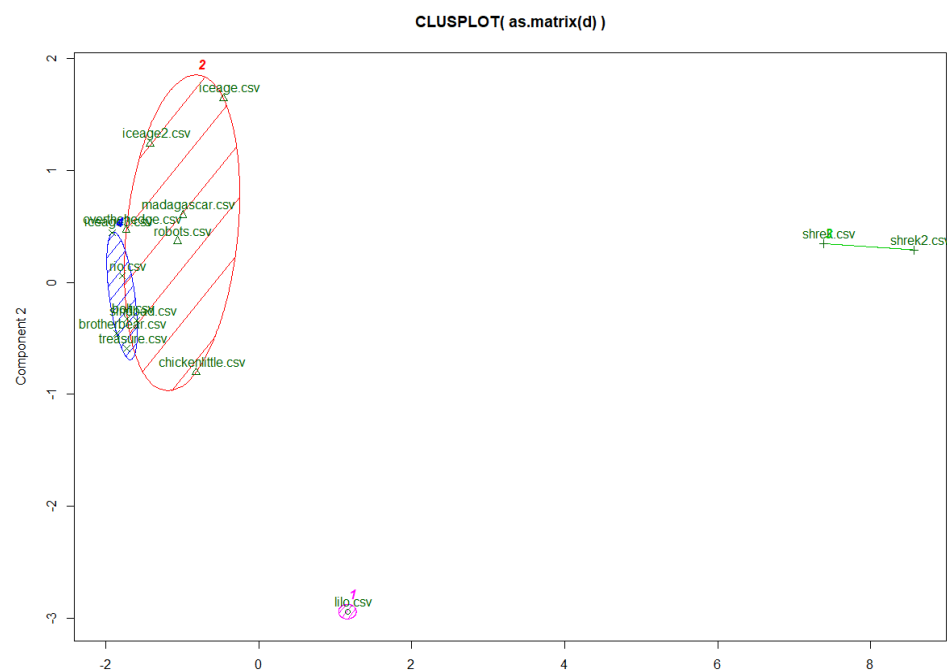


Fig 3.

2.9 Latent semantic analysis

The table below shows Topic Modeling of all the films based on the LATENT DIRICHLET ALLOCATION algorithm. The number of topics was created based on the v. The topics were named after 10 most frequent words in that topic. [Code](#)

Movie name	Topic 1 (10)	Topic 2 (10)	Topic 3 (10)	Topic 4 (10)
Bolt (214)	Film recommendation : movi, like, just, realii, see, love, watch, kid, action, also numb: 84	Film spectrum: film, anim, disney, charact, good, stori, voic, great, travolta, john numb: 63	Film plot: bolt, dog, show, penni, make, get, real, cat, think, believ numb: 67	
Brother bear (206)	Film spectrum: stori, great, make, music, love, song, scene, peopl, feel, charact numb: 44	Film plot: bear, brother, kenai, koda, kill, two, take, love, voic, moos numb: 50	Film actors & producers: film, disney, anim, good, best, much, look, collin, say, phil numb: 52	Film recommendation : movi, like, see, just, charact, kid, watch, think, get, realli numb: 60
Chicken little (293)	Film plot: littl, chicken, alien, sky, fall, father, voic, end, also, town numb: 74	Film spectrum: like, just, stori, charact, good, realli, make, even, think, thing numb: 70	Film recommendation: fmovi, kid, see, watch, children, year, enjoy, great, love, laugh numb: 82	Film producers: film, disney, anim, pixar, first, made, say, lot, charact, mani numb: 67
Ice age (423)	Film recommendation : movi, good, like, funni, great, just, kid, realli, watch, see numb: 192	Film plot: sid, age, babi, human, ice, diego, voic, mammoth, sloth, tiger numb: 116	Film spectrum: film, anim, charact, stori, age, ice, shrek, even, look, make numb: 115	
Ice age 2 (259)	Film description: ice, age, first, sequel, scrat, charact, origi, meltdown, still, new numb: 55	Film spectrum: film, anim, charact, scene, plot, even, seem, get, stori, joke numb: 52	Film recommendation: movi, good, just, like, funni, see, realli, great, think, kid numb: 88	Film plot: manni, mammoth, sid, diego, elli, possum, end, find, voic, water numb: 64
	Film plot:	Film description:	Film	Film spectrum:

Ice age 3 (136)	dinosaur, sid, buck, manni, new, diego, elli, babi, get, love numb: 38	age, ice, first, still, dinosaur, funni, two, part, also, better numb: 29	recommendation: movi, like, good, just, watch, kid, see, realli, enjoy, much numb: 43	film, charact, anim, stori, scene, second, make, voic, littl, pegg numb: 26
Lilo and Stitch (421)	Film plot: stitch, lilo, famili, littl, alien, sister, get, also, nani, girl numb: 167	Film background: movi, disney, film, anim, charact, see, love, good, just, great numb: 254		
Madagascar (331)	Film recommendation : movi, like, funni, see, good, kid, watch, realli, get, great numb: 127	Film spectrum: anim, charact, stori, even, make, also, plot, littl, scene, end numb: 53	Film producers: film, just, madagascar, dreamwork, much, shrek, joke, seem, look, know numb: 65	Film plot: penguin, zoo, lion, voic, madagascar, alex, marti, zebra, wild, friend numb: 86
Over the hedge (298)	Film plot: hedg, food, anim, voic, bear, human, raccoon, willi, find, bruce numb: 81	Film recommendation : movi, kid, see, watch, love, even, famili, much, plot, scene numb: 120	Film spectrum: film, anim ,charact, like, good, make, stori, just, also, realli numb: 97	
Rio (219)	Film spectrum: movi, charact, like,snim, good,watch, great, realli, just, music numb: 144	Film plot: rio, film, bird, flu, anim, stori, macaw, voic, blue, jewel numb: 75		
Robots (296)	Film recommendation : movi, anim, good, see, kid, great, watch, adult, enjoy, think numb: 112	Film spectrum: film, like, charact, just, stori, realli, funni, joke, even, much numb: 105	Film plot: robot, rodney, voic, william, robin, citi, big, world, make, part numb: 79	
Shrek (706)	Film plot: shrek, donkey, tale, fairi, princess, murphi, voic, eddi, ogr, fiona numb: 211	Film spectrum: film, anim, charact, stori, disney, even, make, mani, best, children numb: 216	Film recommendation: movi, like, good, just, funni, see, great, realli, watch, love numb: 279	

Shrek 2 (651)	Film recommendation : movi, firs, just, great, funni, see, like, wach, get, realli numb:286	Film spectrum: film, charact, anim, good, origin, sequelk, stori, even, first, make numb: 177	Film plot: shrek ,fiona, fairi, far, puss, donkey, boot, godmoth, charm, king numb: 188	
Sindbad (92)	Film spectrum: film, anim, good, like, just, great, stori, voic, enjoy, see numb: 60	Film plot: sindbad, movi, charact, eri, book, love, realli, also, adventur, proteus numb: 32		
Treasure planet (309)	Film plot: charact, jim, silver, space, john, also, voic, sihp, hawkin numb: 92	Film spectrum: disney, treasur, film, anim, planet, stori, make, much, island, classic numb: 96	Film recommendation: movi, like, good, great, see, realli, watch, love, just, still numb: 121	

The most frequent number of topics was 3. The number of topics does not correlate with the number of words in a given film. The structure of topics was quite similar. Always we have a plot topic associated mostly with film characters. Often there are topics like spectrum which contains words describing other qualities of film then its plot like words: “stori”, “charact”, “film”, “anim”. Many times there is a topic with recommendations, which contains mostly emotional words. When there are 4 topics we have actors & producers or producers topic which contains actors and producers names. The rest of the topics are probably very similar to the ones I have mentioned already.

2.10 Topic-based Customer Engagement

The numbers of reviews corresponding to each topic that were discovered in the last section were used to create rankings of movies that have reviews most often mentioning a given topic. This analysis was done for three topics - "Film plot", "Film spectrum" and "Film recommendation". More topics were found in earlier analysis (such as "Film producers"), but they were found only in a limited number of films.

Fig 4. Number of reviews mentioning topic "Film plot"

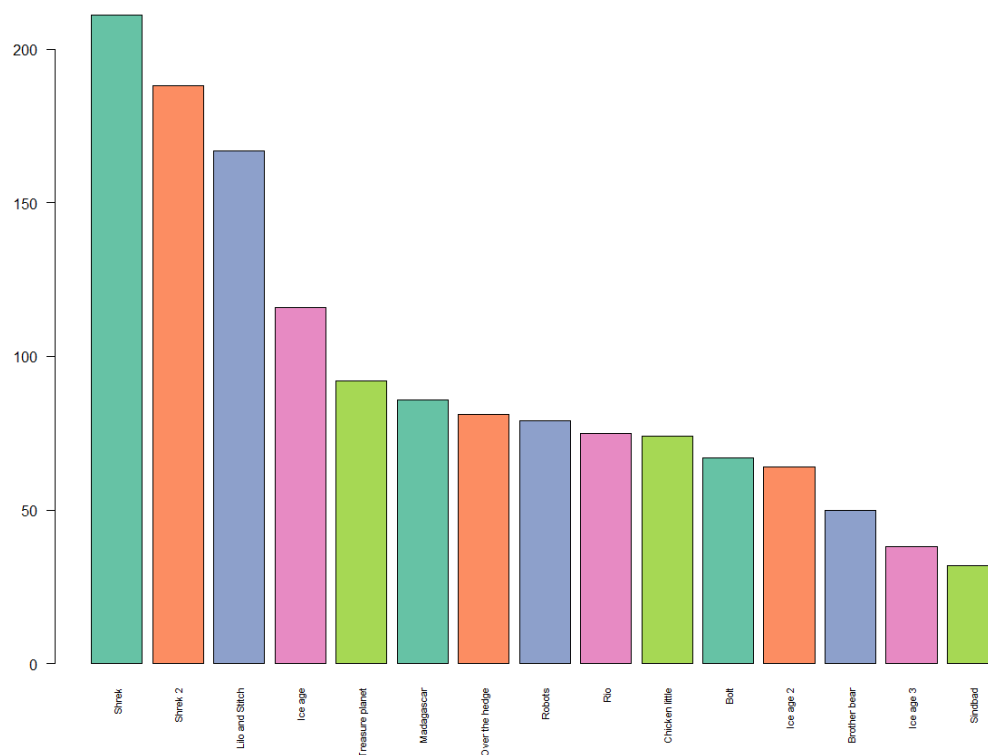


Fig 5. Number of reviews mentioning topic “Film spectrum”

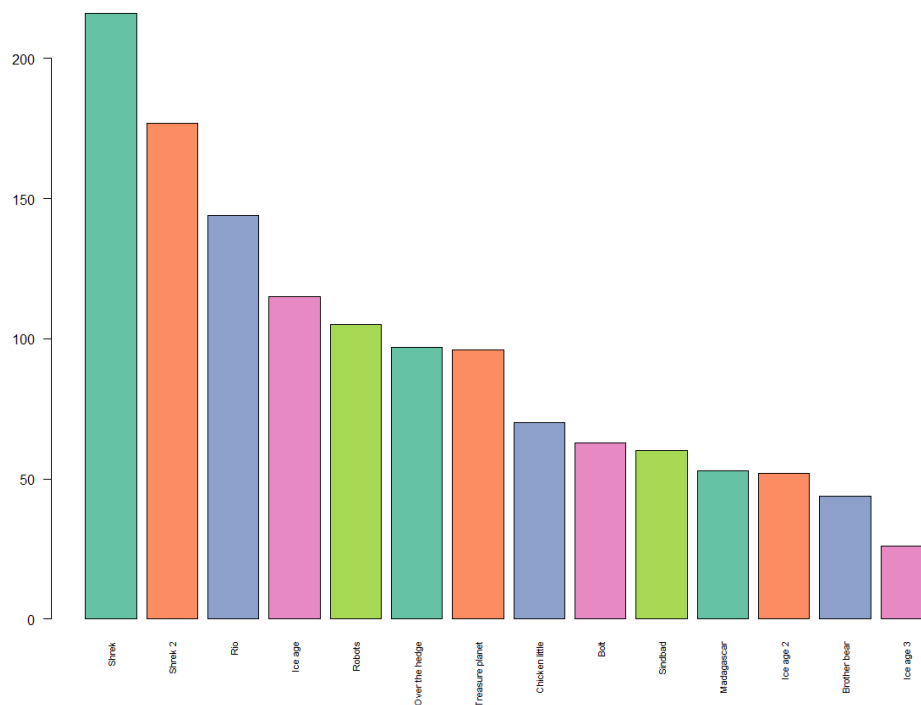
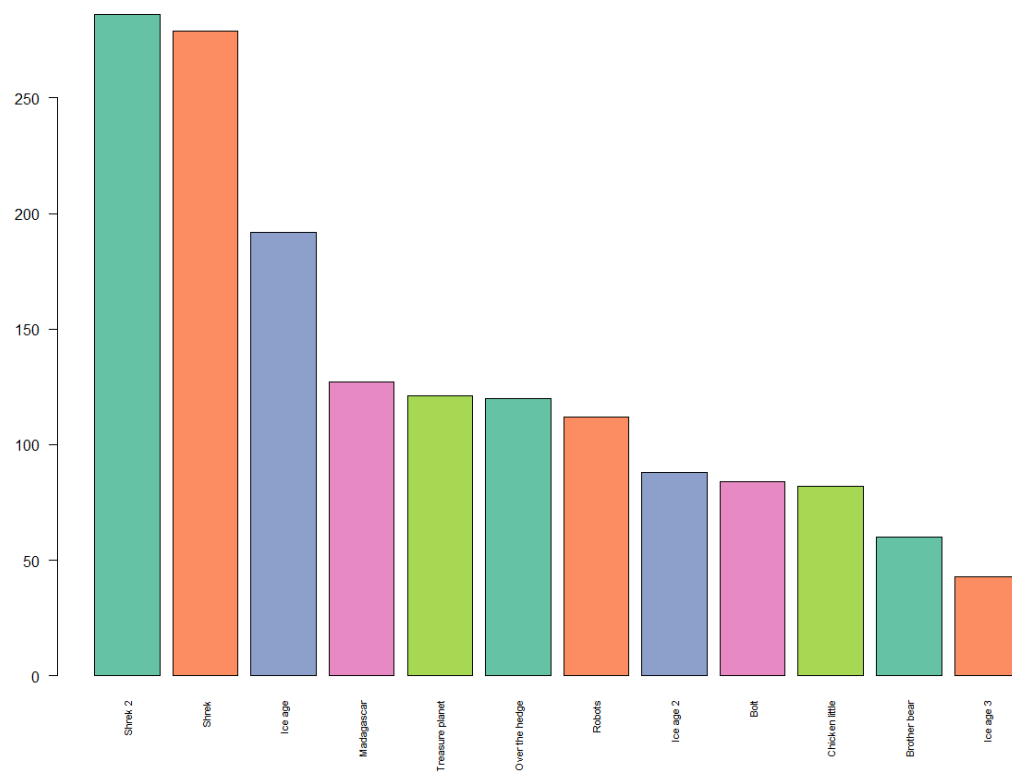


Fig 6. Number of reviews mentioning topic “Film recommendation”



2.11 Clusterization of Topics

The table below shows clusterization of topics in a given film using hierarchical dendrograms. [Code](#)

Movie name	Topics
Bolt (214)	<pre>graph TD; A[Film_plot.txt] --- B[Film_recommendation.txt]; A --- C[Film_spectrum.txt]; B --- C;</pre>
Brother bear (206)	<pre>graph TD; A[Film_plot.txt] --- B[Film_actors&producers.txt]; A --- C[Film_recommendation.txt]; A --- D[Film_spectrum.txt]; B --- C; B --- D; C --- D;</pre>
Chicken little (293)	<pre>graph TD; A[Film_plot.txt] --- B[Film_recommendation.txt]; A --- C[Film_producers.txt]; A --- D[Film_spectrum.txt]; B --- C; B --- D; C --- D;</pre>

Ice age (423)	<pre>graph TD; A[Film_plot.txt] --> B[Film_recommendation.txt]; A --> C[Film_spectrum.txt]; B --> D[Film_description.txt]; B --> E[Film_spectrum.txt];</pre>
Ice age 2 (259)	<pre>graph TD; A[Film_plot.txt] --> B[Film_recommendation.txt]; A --> C[Film_spectrum.txt]; B --> D[Film_description.txt]; B --> E[Film_spectrum.txt]; D --> F[Film_spectrum.txt];</pre>
Ice age 3 (136)	<pre>graph TD; A[Film_plot.txt] --> B[Film_recommendation.txt]; A --> C[Film_spectrum.txt]; B --> D[Film_description.txt]; B --> E[Film_spectrum.txt]; D --> F[Film_spectrum.txt];</pre>
Lilo and Stitch (421)	only 2, don't need

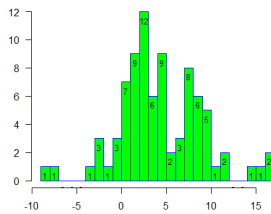
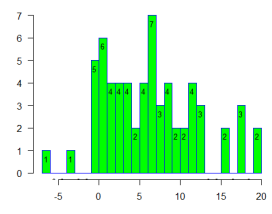
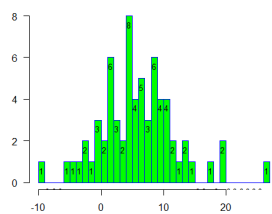
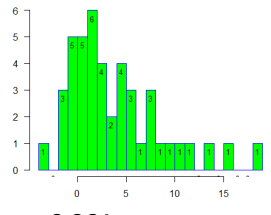
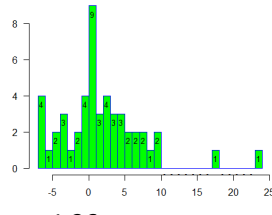
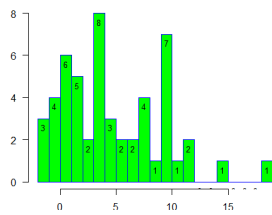
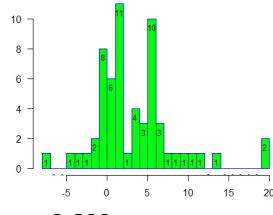
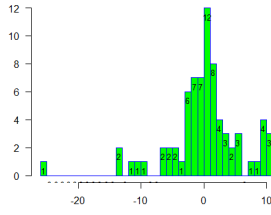
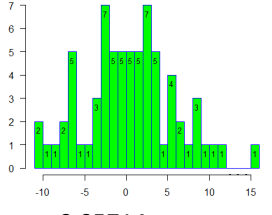
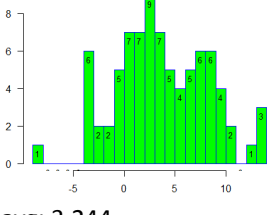
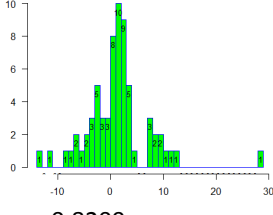
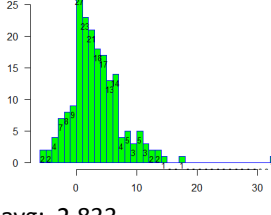
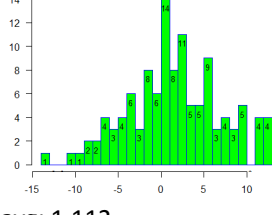
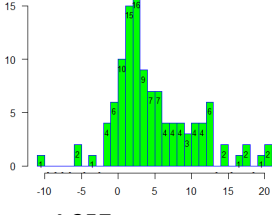
Madagascar (331)	<pre>graph TD; A[Film_plot.txt] --> B[Film_recommendation.txt]; A --> C[Film_producers.txt]; B --> D[Film_producers.txt]; B --> E[Film_spectrum.txt]; C --> E;</pre>
Over the hedge (298)	<pre>graph TD; A[Film_recommendation.txt] --> B[Film_plot.txt]; A --> C[Film_spectrum.txt]; B --> C;</pre>
Rio (219)	only 2, don't need
Robots (296)	<pre>graph TD; A[Film_plot.txt] --> B[Film_recommendation.txt]; A --> C[Film_spectrum.txt]; B --> C;</pre>

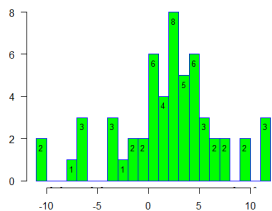
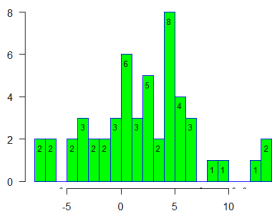
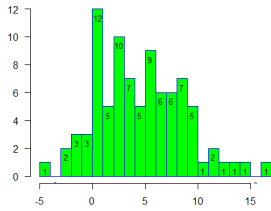
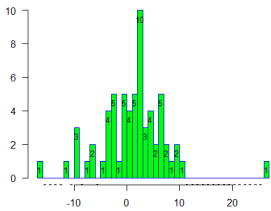
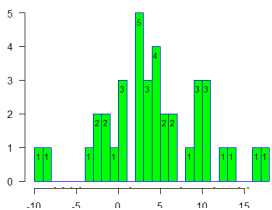
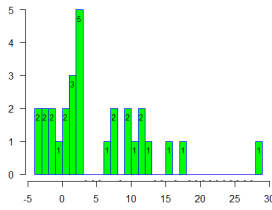
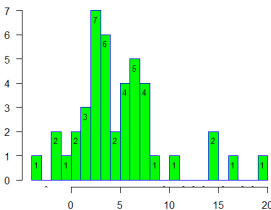
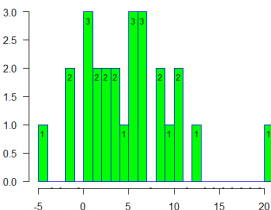
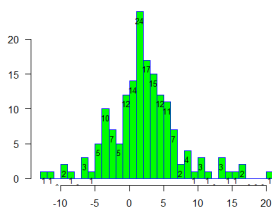
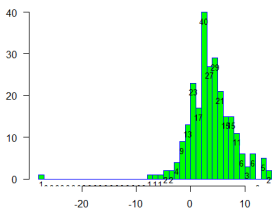
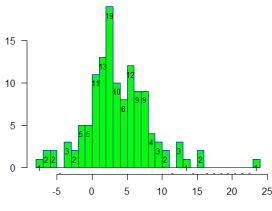
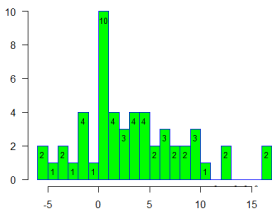
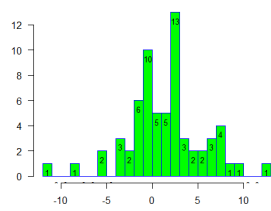
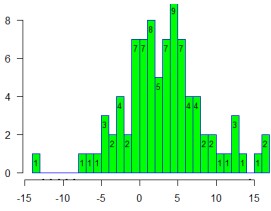
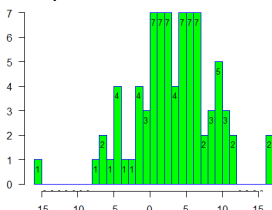
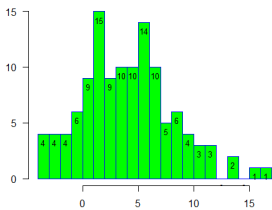
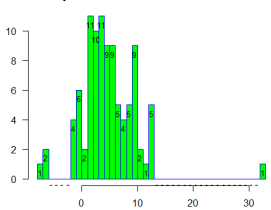
Shrek (706)	<pre> graph TD A[Film_plot.txt] --- B[Film_recommendation.txt] B --- C[Film_spectrum.txt] </pre>
Shrek 2 (651)	<pre> graph TD A[Film_plot.txt] --- B[Film_recommendation.txt] B --- C[Film_spectrum.txt] </pre>
Sindbad (92)	only 2, don't need
Treasure planet (309)	<pre> graph TD A[Film_recommendation.txt] --- B[Film_spectrum.txt] B --- C[Film_plot.txt] </pre>

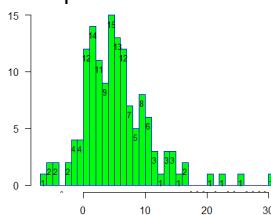
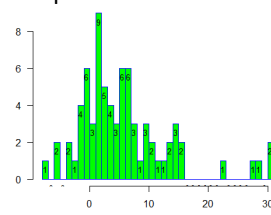
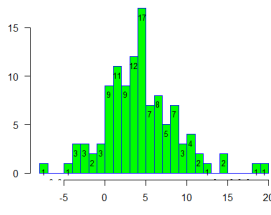
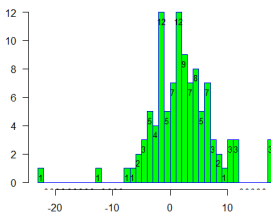
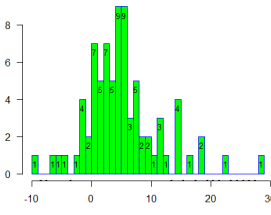
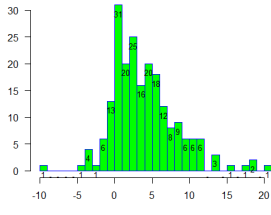
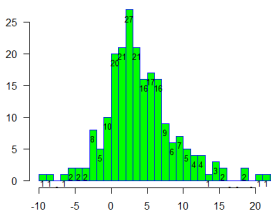
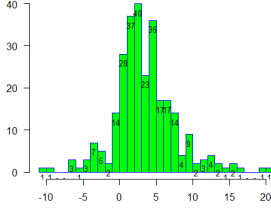
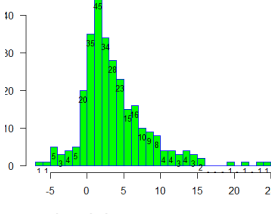
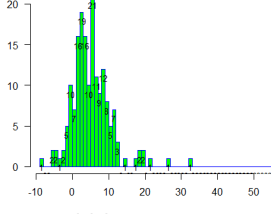
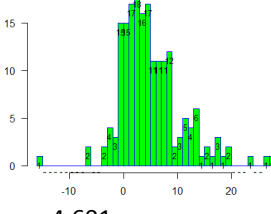
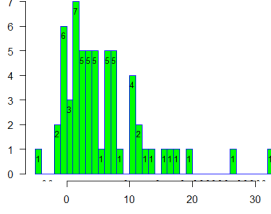
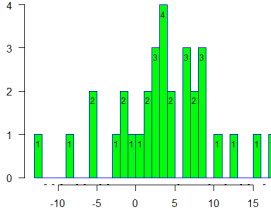
On the above table we can see clustering on topics of given film. The most correlatable topic was spectrum which contains all spectrum words of a given film. Recommendation was correlating the most with spectrum proving that spectrum not only has nouns describing film but also adjectives describing emotions.

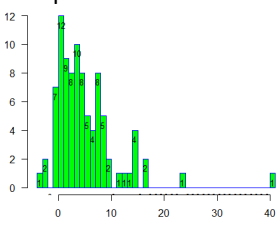
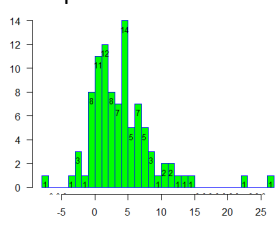
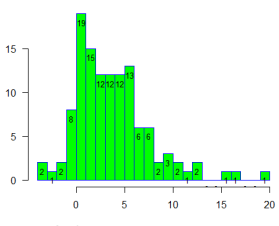
2.12 Sentiment analysis on specific topics

The table below shows a Topic Sentiment analysis chart based on earlier chosen topics. Moreover below every chart there are also statistics about number of words in topic and average sentiment analysis score. [Code](#)

Movie name (# total reviews)	Topic 1 (10)	Topic 2 (10)	Topic 3 (10)	Topic 4 (10)
Bolt (214)	Film recommendation:  avg: 3.845 numb: 84	Film spectrum:  avg: 5.873 numb: 63	Film plot:  avg: 5.448 numb: 67	
Brother bear (206)	Film spectrum:  avg: 3.364 numb: 44	Film plot:  avg: 1.36 numb: 50	Film actors & producers:  avg: 4.25 numb: 52	Film recommendation:  avg: 2.833 numb: 60
Chicken little (293)	Film plot:  avg: -0.4865 numb: 74	Film spectrum:  avg: -0.05714 numb: 70	Film recommendation:  avg: 3.244 numb: 82	Film producers:  avg: 0.8209 numb: 67
Ice age (423)	Film recommendation:  avg: 2.833 numb: 192	Film plot:  avg: 1.112 numb: 116	Film spectrum:  avg: 4.357 numb: 115	

<p><i>Ice age 2</i> (259)</p>	<p>Film description:</p>  <p>avg: 1.455 numb: 55</p>	<p>Film spectrum:</p>  <p>avg: 1.538 numb: 52</p>	<p>Film recommendation:</p>  <p>avg: 4.068 numb: 88</p>	<p>Film plot:</p>  <p>avg: 0.6719 numb: 64</p>
<p><i>Ice age 3</i> (136)</p>	<p>Film plot:</p>  <p>avg: 3.868 numb: 38</p>	<p>Film description:</p>  <p>avg: 4.724 numb: 29</p>	<p>Film recommendation:</p>  <p>avg: 4.512 numb: 43</p>	<p>Film spectrum:</p>  <p>avg: 4.538 numb: 26</p>
<p><i>Lilo and Stitch</i> (421)</p>	<p>Film plot:</p>  <p>avg: 1.653 numb: 167</p>	<p>Film background:</p>  <p>avg: 3.213 numb: 254</p>		
<p><i>Madagascar</i> (331)</p>	<p>Film recommendation:</p>  <p>avg: 3.189 numb: 127</p>	<p>Film spectrum:</p>  <p>avg: 2.906 numb: 53</p>	<p>Film producers:</p>  <p>avg: 0.9385 numb: 65</p>	<p>Film plot:</p>  <p>avg: 2.709 numb: 86</p>
<p><i>Over the hedge</i> (298)</p>	<p>Film plot:</p>  <p>avg: 2.778 numb: 81</p>	<p>Film recommendation:</p>  <p>avg: 3.592 numb: 120</p>	<p>Film spectrum:</p>  <p>avg: 4.247 numb: 97</p>	

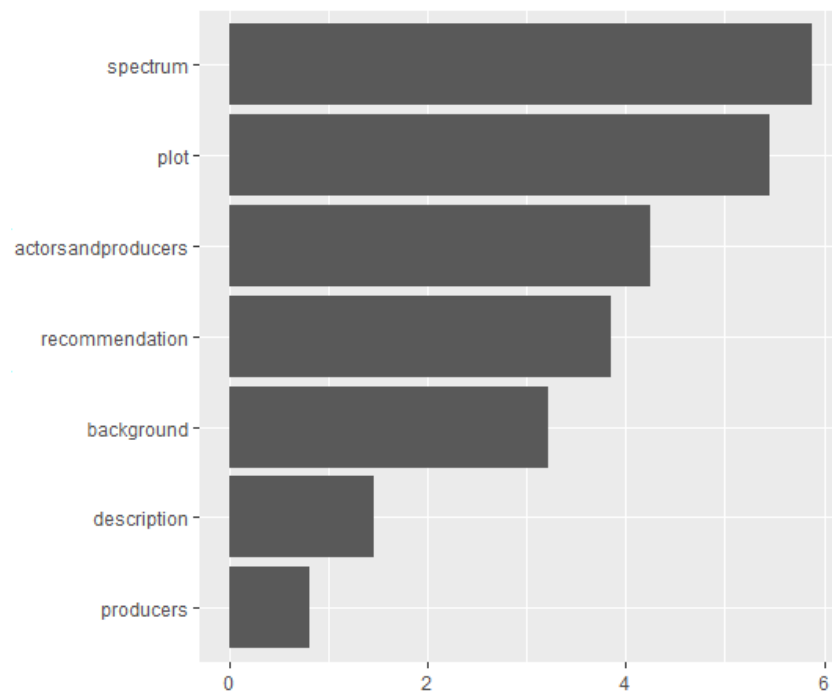
Rio (219)	<p>Film spectrum:</p>  <p>avg: 4.896 numb: 144</p>	<p>Film plot:</p>  <p>avg: 5.32 numb: 75</p>		
Robots (296)	<p>Film recommendation:</p>  <p>avg: 3.866 numb: 112</p>	<p>Film spectrum:</p>  <p>avg: 1.648 numb: 105</p>	<p>Film plot:</p>  <p>avg: 4.747 numb: 79</p>	
Shrek (706)	<p>Film plot:</p>  <p>avg: 3.493 numb: 211</p>	<p>Film spectrum:</p>  <p>avg: 3.63 numb: 216</p>	<p>Film recommendation:</p>  <p>avg: 2.978 numb: 279</p>	
Shrek 2 (651)	<p>Film recommendation:</p>  <p>avg: 3.196 numb: 286</p>	<p>Film spectrum:</p>  <p>avg: 5.006 numb: 177</p>	<p>Film plot:</p>  <p>avg: 4.601 numb: 188</p>	
Sindbad (92)	<p>Film spectrum:</p>  <p>avg: 5.467 numb: 60</p>	<p>Film plot:</p>  <p>avg: 3.031 numb: 32</p>		

Treasure planet (309)	<p>Film plot:</p>  <p>avg: 4.576 numb: 92</p>	<p>Film spectrum:</p>  <p>avg: 3.531 numb: 96</p>	<p>Film recommendation:</p>  <p>avg: 3,24 numb: 121</p>	
---------------------------------	--	--	---	--

We can see on the chart that almost all topics have positive sentiment scores with almost normal distribution which means that people's opinion does not vary much in terms of a given topic..More details are specified on the tables and charts below.

Sentiment scores grouped by topic

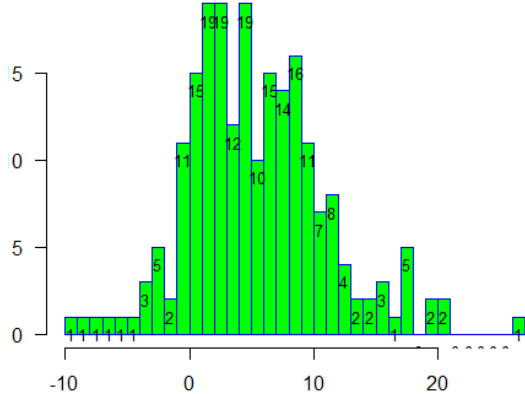
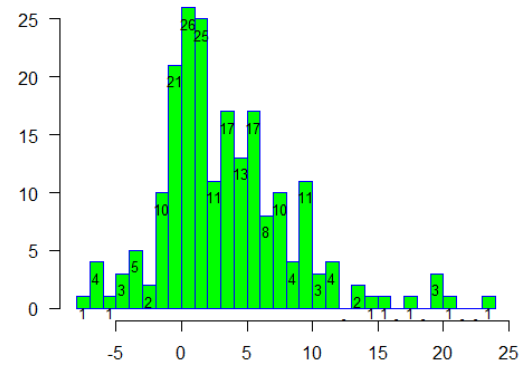
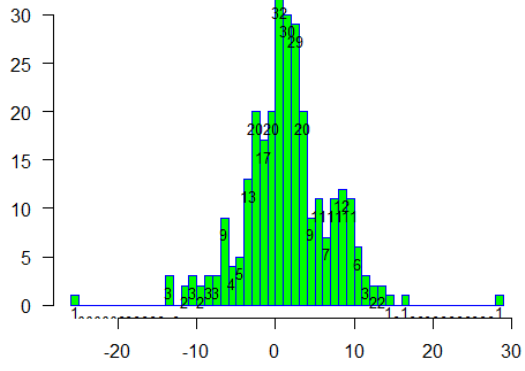
The table below shows average sentiment scores from each topic discussed in table above.

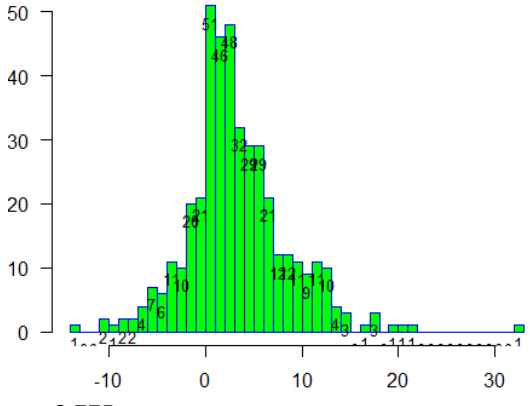
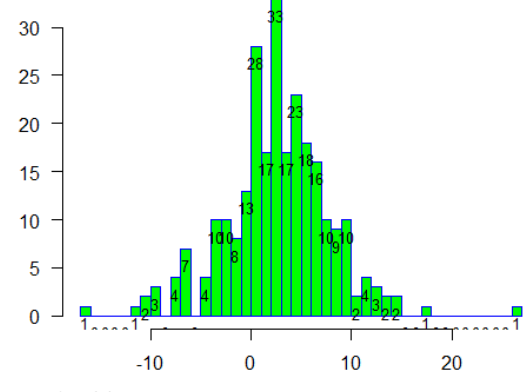
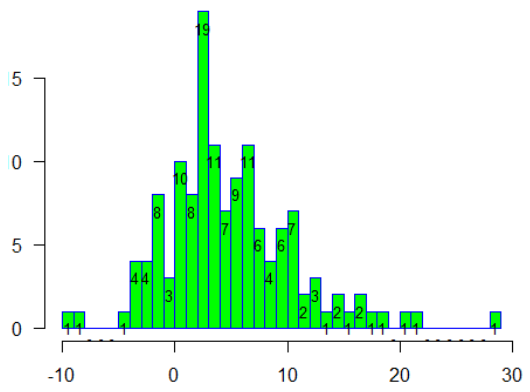
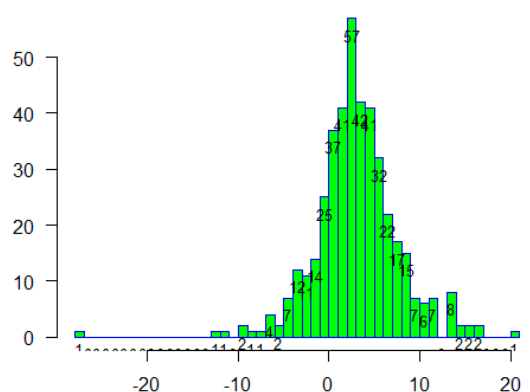


We can conclude that viewers like the spectrum and plot of the film the most, in contrast to producers and description.

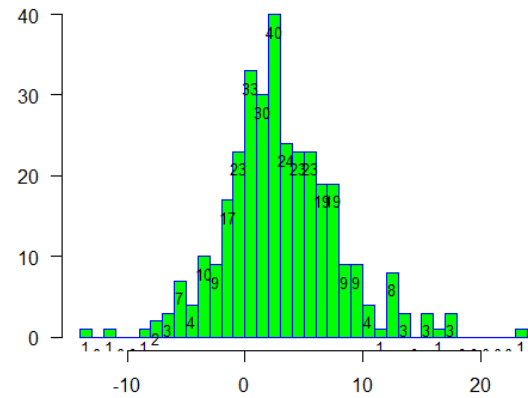
2.13 Sentiment analysis of movies

Table below shows sentiment analysis of movies with their average scores and chart. [Code](#)

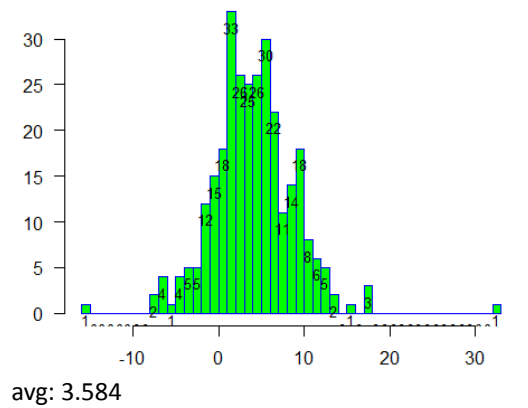
Movie name	Customer sentiment
<i>Bolt</i> (214)	 <p>avg: 4.944</p>
<i>Brother bear</i> (206)	 <p>avg: 2.947</p>
<i>Chicken little</i> (293)	 <p>avg: 0.959</p>

<p><i>Ice age</i> (423)</p>	 <p>avg: 2.775</p>
<p><i>Ice age 2</i> (259)</p>	 <p>avg: 2.166</p>
<p><i>Ice age 3</i> (136)</p>	 <p>avg: 4.382</p>
<p><i>Lilo and Stitch</i> (421)</p>	 <p>avg: 2.594</p>

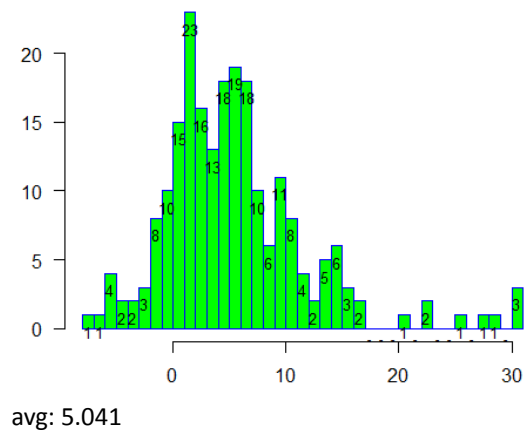
Madagascar
(331)



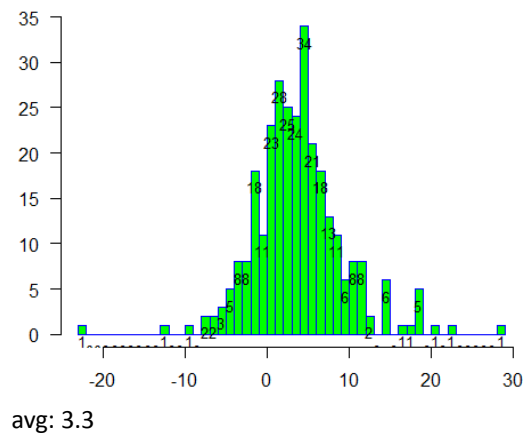
Over the hedge
(298)



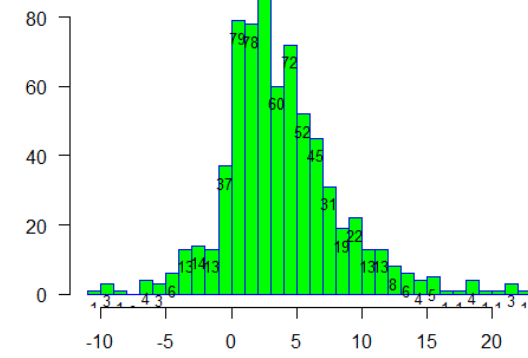
Rio
(219)



Robots
(296)

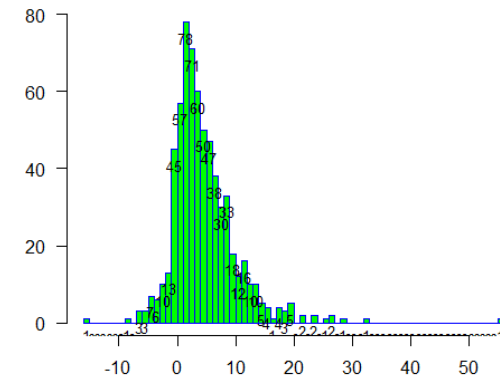


Shrek
(706)



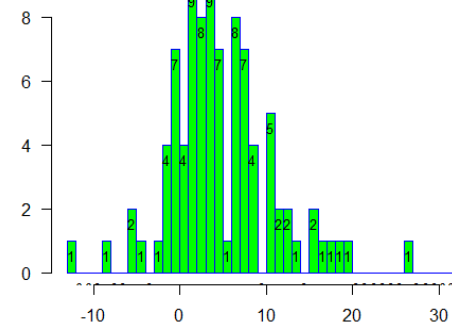
avg: 3.331

Shrek 2
(651)



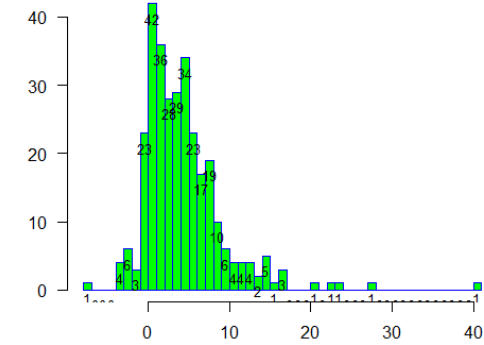
avg: 4.094

Sindbad
(92)



avg: 4.62

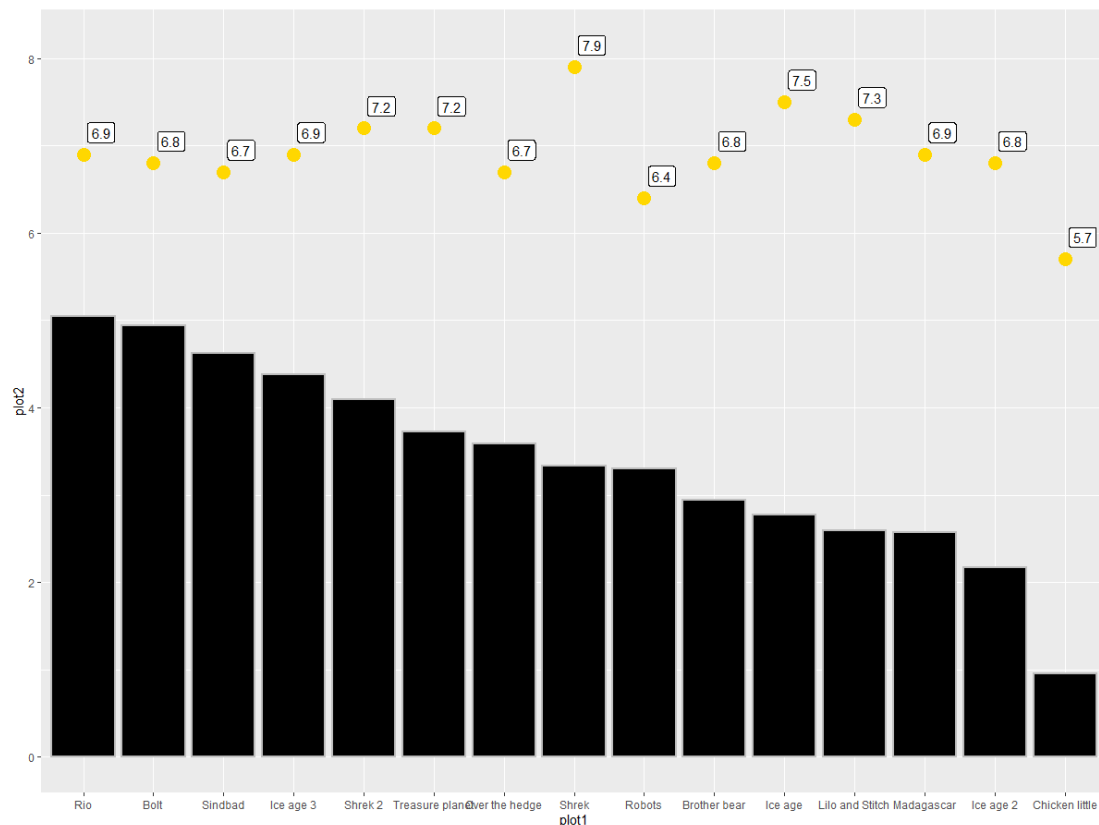
Treasure planet
(309)



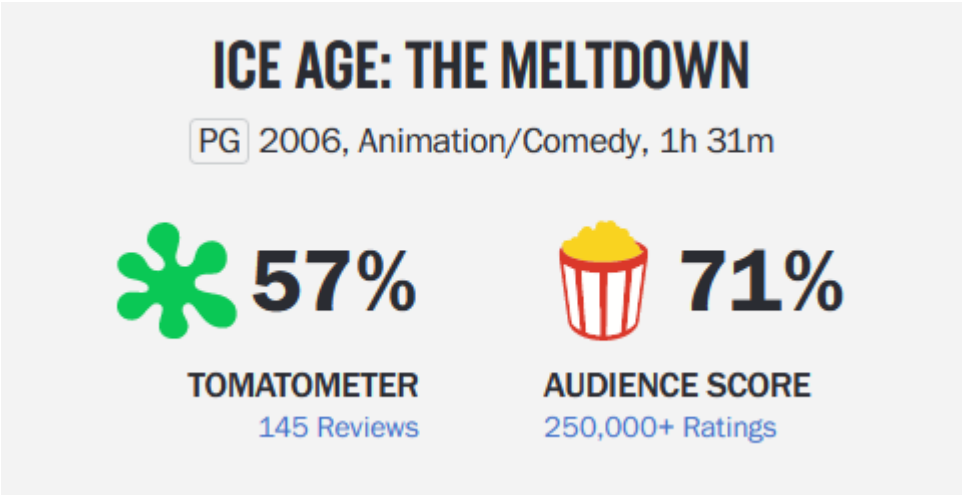
avg: 3.728

To better visualize the sentiment of films below is a chart comparing average scores of films to IMDB ratings. Nonetheless we can see that every chart above contains very big outliers which suggests that almost every film has mostly its superfans or rarely big haters.

Average sentiment analysis scores mapped against IMDB ratings

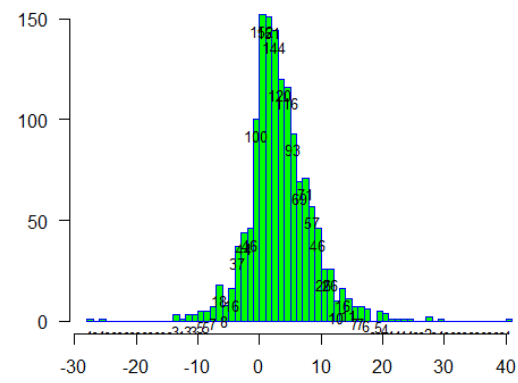


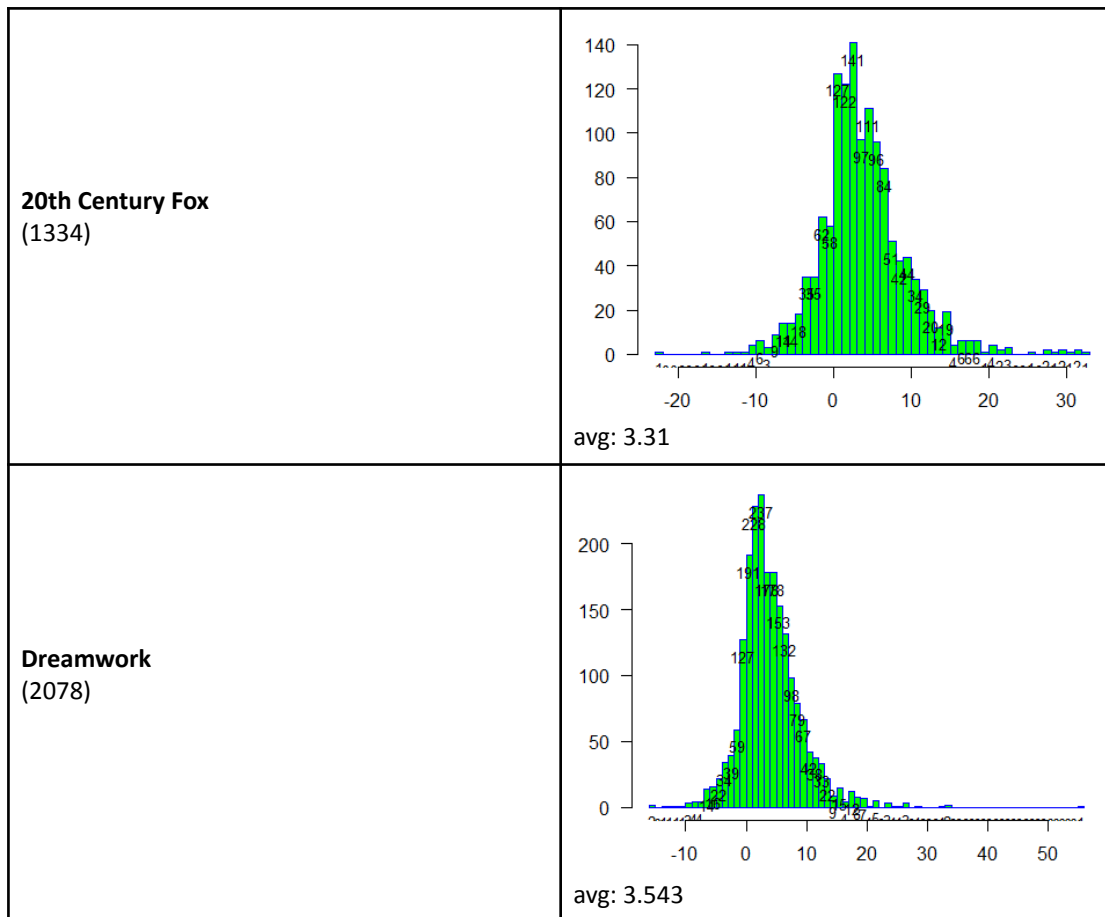
In this chart we see that the sentiment analysis results don't correspond too closely with how the movies are rated on the IMDB site. This may be because the written reviews are only about 0.1% of the "rating" reviews placed on site. The larger customer consensus may vary from the opinion of the small subset of customers willing to place written reviews. Moreover, this may be due to the existence of two groups of viewers: critics and casual fans. Critics are obviously more likely to place written reviews, with the casual fans less likely to do so. Some review sites, such as RottenTomatoes.com even present two separate scores - for the critics and the audience. Below we see the RottenTomatoes rating for the movie Ice age 2 with a similar discrepancy:



2.14 Sentiment Analysis of production companies

The table below shows the chart and average score of 3 production companies(each one has produced 5 films in our corpus). [Code](#)

Production company name (# total reviews)	Customer sentiment
Disney (1443)	 <p>avg: 2.904</p>



The best results have Dreamworks with also the highest amount of words used.

3. Summary

3.1 Conclusions

The analysis produced interesting results that answered most of our research questions. Inter-corporal hierarchical and cosine clustering revealed similarities between the movies - with certain accuracy grouping them by the studios that produced them, but also by general theme, topic or style. We were able to outline distinct topics that appear across most reviews and gauge customer engagement for

different topics across all movies. We devised a formula which can determine customer engagement based on the number of reviews and their word length. Finally, we have learnt that customer sentiment based on the written reviews is not an accurate indicator of the overall customer-perceived quality of the film.

3.2 Business insights

- Judging a film's success based only on popularity (number of reviews) may obscure the actual factor of customer engagement, as presented by the PAEI table and chart. (This is true at least for the engagement of a subset customers that place written reviews)
- Frequent word associations provide insight into which factors the customers paid most attention to (for example "movie - funny"). Useful to know which factors to emphasize in the production of further films.
- For children's animated films the critic scores may not be tied too closely to audience scores and generally will tend to be lower. This means that films receiving lower critic scores (for instance during pre-screening) will not necessarily impact the film's commercial success.
- IMDB.com is a rich and useful source of information, both concerning user ratings and reviews, however the review data has to be scraped by means prohibited by the site's terms of service, hence the data isn't "freely available" and analysis thereof may be tied to extra costs.

4. Appendix

Appendix 1: Review scraping code (python):

```
import requests
import csv
from bs4 import BeautifulSoup

s = requests.Session()
# get first/full page

url = 'https://www.imdb.com/title/tt0327084/reviews/?ref_=tt_ql_urv'

r = s.get(url)
soup = BeautifulSoup(r.text, 'html.parser')
```

```

items = soup.find_all('a', {'class': 'title'})
for number, title in enumerate(items, 1):
    print(number, '>', title.text.strip())

# get next page(s)

with open('C:\\Users\\adria\\Desktop\\overthehedge.txt', 'w', encoding="utf-8") as f:
    #csvwriter = csv.writer(f)
    for _ in range(12):

        div = soup.find('div', {'data-key': True})
        #print('---', div['data-key'], '---')

        url = 'https://www.imdb.com/title/tt0327084/reviews/_ajax'

        payload = {
            'ref_': 'tt_ql_urv',
            'paginationKey': div['data-key']
        }

        #headers = {'X-Requested-With': 'XMLHttpRequest'}

        r = s.get(url, params=payload) #, headers=headers)
        soup = BeautifulSoup(r.text, 'html.parser')

        items = soup.find_all('div', {'class': 'content'})
        for number, content in enumerate(items, 1):
            f.write(content.text.strip())

```

Appendix 2: Preprocessing:

```

file_loc
"C:\\Users\\NorbiX\\Desktop\\studia\\sem5\\DatamininginBusiness\\Dorpusproject1csvka"
<-

docs<- VCorpus(DirSource(directory=file_loc,encoding="UTF-8"))
writeLines(as.character(docs[[1]]))
docs <- tm_map(docs,removePunctuation)
docs <- tm_map(docs, removeNumbers)
for (j in seq(docs)) {
  docs[[j]] <- gsub("/", " ", docs[[j]])
  docs[[j]] <- gsub("@", " ", docs[[j]])
  docs[[j]] <- gsub("-", " ", docs[[j]])
  docs[[j]] <- gsub("'", " ", docs[[j]])
  docs[[j]] <- gsub("'", " ", docs[[j]])
  docs[[j]] <- gsub("...", " ", docs[[j]])
  docs[[j]] <- gsub(":", " ", docs[[j]])
  docs[[j]] <- gsub(")", " ", docs[[j]])
  docs[[j]] <- gsub("'''", " ", docs[[j]])
}

docs <- tm_map(docs, tolower)
docs <- tm_map(docs, removeWords, stopwords("English"))
StW<-read.table("C:/Users/NorbiX/Desktop/studia/sem5/DatamininginBusiness/StopWords.txt")

```

```

StWW<-as.character(StW$V1)
docs <- tm_map(docs, removeWords, StWW)
docs <- tm_map(docs, stripWhitespace)
for (j in seq(docs)) {
  docs[[j]]<-stemDocument(docs[[j]], language = "english")
}
writeLines(as.character(docs[[1]]))

write.csv(docs[[10]],file="C:/Users/Norbix/Desktop/studia/sem5/DatamininginBusiness/finalcsvcorpus
project1/docs10.csv")
docs[[6]]

```

Appendix 3: Top 3 associations with the top 3 frequent words:

```

freqr <- colSums(as.matrix(dtm))
freq <- sort(freqr, decreasing=TRUE)
head(freq, 3)
findAssocs(dtm,"bolt",0.25)

```

Appendix 4: Latent semantic analysis:

```

raw.sum=apply(dtm,1,FUN=sum)
raw.sum

mmm<-nrow(dtm[raw.sum==0,])
mmm

```



```

if (mmm==0) {
  dtm2<-dtm
  NN<-nrow(dtm)
  NN
} else {
  dtm2<-dtm[raw.sum!=0,]
  NN<-nrow(dtm2)
}
dtm2

system.time({
  tunes <- FindTopicsNumber(
    dtm = dtm2,
    topics = c(2:15),
    metrics = c("Griffiths2004", "CaoJuan2009", "Arun2010"),
    method = "Gibbs",
    control = list(seed = 12345),
    mc.cores = 4L,
    verbose = TRUE
  )
})
FindTopicsNumber_plot(tunes)

```

```

burnin <- 4000
iter <- 2000
thin <- 500
seed <-list(2003,5,63,100001,765)
nstart <- 5
best <- TRUE
k <- 3

```

```

ldaOut <-LDA(dtm2, k, method="Gibbs", control=list(nstart=nstart, seed = seed, best=best, burnin =
burnin, iter = iter, thin=thin))
str(ldaOut)
# ____ topics keywords_____
ldaOut.terms <- as.matrix(terms(ldaOut, 10))
ldaOut.terms

```

Appendix 5: Semantic analysis of topics:

```

ldaOut.topics <- as.matrix(topics(ldaOut))
rownames(ldaOut.topics)<-as.character(rownames(dtm2))
ldaOut.topics
nrow(ldaOut.topics)

```

```

Comment<-seq(1, NN, by=1)
Comment
wf=data.frame(Comment=Comment, Topics=ldaOut.topics)
wf
#_____
#_____ Building Sub-Corpus of Topic 1_____

topic1<-wf[wf[2] == 1,]      #find Topic 1
topic1$Comment
length(topic1$Comment)

#number of comments with Topic 1
kk1<-nrow(topic1)
kk1
kk<-nrow(dtm2)
kk

list1<-c()
i=1
while(i<=kk) {
  if (wf[i,2]==1) {          #find Topic 1
    list1<-c(list1,i)}
  i=i+1
}
length(list1)

wf1=NULL
for (i in 1:kk) {
  for (j in 1:kk1) {
    if (i==list1[j]){
      c <- data.frame(file=as.character(wf[list1[j],1]),document=as.character(corp[[i]]))
      wf1=rbind(wf1,c)
    }
  }
}
wf1
wf1$document[1]

#_____ Corpus Creating_____
Topic_1_docs <- Corpus(VectorSource(as.character(wf1$document))) #Corpus for Topic 1

neg=scan("negative-words.txt", what="character", comment.char=";" )
pos=scan("positive-words.txt", what="character", comment.char=";" )

#_____ Initialization of the Sentiment analysis Procedure_____

score.sentiment = function(docs, pos.words, neg.words, .progress='none')
{
  scores = laply(docs_s, function(docs, pos.words, neg.words) {

    word.list = str_split(docs, '\\s+')
    words = unlist(word.list)

    # compare our words to the dictionaries of positive & negative terms
    pos.matches = match(words, pos.words)
    neg.matches = match(words, neg.words)
  })
}

```

```

# match() returns the position of the matched term or NA
# we just want a TRUE/FALSE:
pos.matches = !is.na(pos.matches)
neg.matches = !is.na(neg.matches)

# and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():
score = sum(pos.matches) - sum(neg.matches)

return(score)
}, pos.words, neg.words, .progress=.progress )

scores.df = data.frame(score=scores, text=docs)
return(scores.df)
}

```

_____ Topic 1 Sentiment Scoring _____

```

result=c()

docs<-Topic_1_docs
m1=c()
for (j in seq(docs)) {
  docs_s=as.character(docs[[j]])
  print(docs_s)
  result = score.sentiment(docs_s, pos, neg)
  newRow1 <- data.frame(Doc=j,Score = result$score, Documents = result$text)
  #print(newRow1)
  m1<- rbind(m1,newRow1)
  #print(m1)
}
head(m1)
m1[1:3,]

```

_____ Statistics _____

```

summary(m1$Score)
minn<-min(m1$Score)
minn
maxx<-max(m1$Score)
maxx
mmm<-maxx-minn
mmm

```

_____ Topic 1 _____ Histograms _____

_____ Histogram_1 _____

```

h<-hist(m1$Score,
  main="Histogram for the Sentiment by Topic _____",
  xlab="Scores",
  ylab="Number of of Opinions",
  right=FALSE,
  border="blue",
  col="green",
  freq=TRUE,
  las=1,
  xlim=c(minn,maxx),
  breaks=mmm
)

```

```
text(h$mids,h$counts,labels=h$counts, adj=c(0.5, -0.5),cex = 0.8, pos = 1)
m1$Score
h$count
```

Appendix 6: Semantic analysis of movies:

```
file_loc <-
"C:\\Users\\Norbi\\Desktop\\studia\\sem5\\DatamininginBusiness\\temp\\temp\\bolt.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:214)

file_loc <-
"C:\\Users\\Norbi\\Desktop\\studia\\sem5\\DatamininginBusiness\\temp\\temp\\brotherbear.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:206)

file_loc <-
"C:\\Users\\Norbi\\Desktop\\studia\\sem5\\DatamininginBusiness\\temp\\temp\\chickenlittle.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:293)

file_loc <-
"C:\\Users\\Norbi\\Desktop\\studia\\sem5\\DatamininginBusiness\\temp\\temp\\iceage.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:423)

file_loc <-
"C:\\Users\\Norbi\\Desktop\\studia\\sem5\\DatamininginBusiness\\temp\\temp\\iceage2.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:259)

file_loc <-
"C:\\Users\\Norbi\\Desktop\\studia\\sem5\\DatamininginBusiness\\temp\\temp\\iceage3.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:136)

file_loc <-
"C:\\Users\\Norbi\\Desktop\\studia\\sem5\\DatamininginBusiness\\temp\\temp\\lilo.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:421)

file_loc <-
"C:\\Users\\Norbi\\Desktop\\studia\\sem5\\DatamininginBusiness\\temp\\temp\\madagascar.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
```

```

x$doc_id<-(1:331)
file_loc <- "C:\\Users\\Norbix\\Desktop\\studia\\sem5\\DatamininginBusiness\\temp\\temp\\overthehedge.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:298)
file_loc <- "C:\\Users\\Norbix\\Desktop\\studia\\sem5\\DatamininginBusiness\\temp\\temp\\rio.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:219)
file_loc <- "C:\\Users\\Norbix\\Desktop\\studia\\sem5\\DatamininginBusiness\\temp\\temp\\robots.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:297)
file_loc <- "C:\\Users\\Norbix\\Desktop\\studia\\sem5\\DatamininginBusiness\\temp\\temp\\shrek.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:706)
file_loc <- "C:\\Users\\Norbix\\Desktop\\studia\\sem5\\DatamininginBusiness\\temp\\temp\\shrek2.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:651)
file_loc <- "C:\\Users\\Norbix\\Desktop\\studia\\sem5\\DatamininginBusiness\\temp\\temp\\sindbad.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:92)
file_loc <- "C:\\Users\\Norbix\\Desktop\\studia\\sem5\\DatamininginBusiness\\temp\\temp\\treasure.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:309)
x <- x %>% select(doc_id,text)
corp <- VCorpus(DataframeSource(x))

```

```

neg=scan("negative-words.txt", what="character", comment.char=";")
pos=scan("positive-words.txt", what="character", comment.char=";")

```

_____ Initialization of the Sentiment analysis Procedure _____

```

score.sentiment = function(docs, pos.words, neg.words, .progress='none')
{
  scores = laply(docs_s, function(docs, pos.words, neg.words) {

    word.list = str_split(docs, '\\s+')
    words = unlist(word.list)

    # compare our words to the dictionaries of positive & negative terms
    pos.matches = match(words, pos.words)
    neg.matches = match(words, neg.words)

    # match() returns the position of the matched term or NA

```

```

# we just want a TRUE/FALSE:
pos.matches = !is.na(pos.matches)
neg.matches = !is.na(neg.matches)

# and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():
score = sum(pos.matches) - sum(neg.matches)

return(score)
}, pos.words, neg.words, .progress=.progress )

scores.df = data.frame(score=scores, text=docs)
return(scores.df)
}

# _____Topic 1 Sentiment Scoring _____

result=c()

#docs<-Topic_1_docs # You need to replace it into Topic_2_docs, ...Topic_5_docs in the next steps
docs<-corp
m1=c()
for (j in seq(docs)) {
  docs_s=as.character(docs[[j]])
  print(docs_s)
  result = score.sentiment(docs_s, pos, neg)
  newRow1 <- data.frame(Doc=j,Score = result$score, Documents = result$text)
  #print(newRow1)
  m1<- rbind(m1,newRow1)
  #print(m1)
}
head(m1)
m1[1:3,]
# _____Statistics _____
summary(m1$Score)
minn<-min(m1$Score)
minn
maxx<-max(m1$Score)
maxx
mmm<-maxx-minn
mmm
# _____Topic 1 _____Histograms _____

# _____Histogram_1 _____
h<-hist(m1$Score,
  main="Histogram for the Sentiment by Topic _____",
  xlab="Scores",
  ylab="Number of of Opinions",
  right=FALSE,
  border="blue",
  col="green",
  freq=TRUE,
  las=1,
  xlim=c(minn,maxx),
  breaks=mmm
)
text(h$mids,h$counts,labels=h$counts, adj=c(0.5, -0.5),cex = 0.8, pos = 1)

```

```
m1$Score
h$count
```

Appendix 7: Semantic analysis of production companies:

```
file_loc <-
"C:\\Users\\Norbi\\Desktop\\studia\\sem5\\DatamininginBusiness\\wytwurniefilmowe\\disney.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:1443)
file_loc <-
"C:\\Users\\Norbi\\Desktop\\studia\\sem5\\DatamininginBusiness\\wytwurniefilmowe\\20th_Century_FOX.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:1334)
file_loc <-
"C:\\Users\\Norbi\\Desktop\\studia\\sem5\\DatamininginBusiness\\wytwurniefilmowe\\DreamWorks.csv"
# change TRUE to FALSE if you have no column headings in the CSV
x <- read.csv(file_loc, header = TRUE)
x$doc_id<-(1:2078)
x <- x %>% select(doc_id,text)
corp <- VCorpus(DataframeSource(x))
neg=scan("negative-words.txt", what="character", comment.char=";" )
pos=scan("positive-words.txt", what="character", comment.char=";" )

# _____ Initialization of the Sentiment analysis Procedure _____

score.sentiment = function(docs, pos.words, neg.words, .progress='none')
{
  scores = laply(docs_s, function(docs, pos.words, neg.words) {

    word.list = str_split(docs, '\\s+')
    words = unlist(word.list)

    # compare our words to the dictionaries of positive & negative terms
    pos.matches = match(words, pos.words)
    neg.matches = match(words, neg.words)

    # match() returns the position of the matched term or NA
    # we just want a TRUE/FALSE:
    pos.matches = !is.na(pos.matches)
    neg.matches = !is.na(neg.matches)

    # and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():
    score = sum(pos.matches) - sum(neg.matches)
```

```

    return(score)
  }, pos.words, neg.words, .progress=.progress )

scores.df = data.frame(score=scores, text=docs)
return(scores.df)
}

#_____Topic 1 Sentiment Scoring_____

result=c()

#docs<-Topic_1_docs # You need to replace it into Topic_2_docs, ...Topic_5_docs in the next steps
docs<-corp
m1=c()
for (j in seq(docs)) {
  docs_s=as.character(docs[[j]])
  print(docs_s)
  result = score.sentiment(docs_s, pos, neg)
  newRow1 <- data.frame(Doc=j,Score = result$score, Documents = result$text)
  #print(newRow1)
  m1<- rbind(m1,newRow1)
  #print(m1)
}
head(m1)
m1[1:3,]
#_____Statistics_____
summary(m1$Score)
minn<-min(m1$Score)
minn
maxx<-max(m1$Score)
maxx
mmm<-maxx-minn
mmm
#_____Topic 1____Histograms_____

#_____Histogram_1_____
h<-hist(m1$Score,
  main="Histogram for the Sentiment by Topic _____",
  xlab="Scores",
  ylab="Number of of Opinions",
  right=FALSE,
  border="blue",
  col="green",
  freq=TRUE,
  las=1,
  xlim=c(minn,maxx),
  breaks=mmm
)
text(h$mids,h$counts,labels=h$counts, adj=c(0.5, -0.5),cex = 0.8, pos = 1)
m1$Score
h$count

```


Appendix 8: Clusterization of topics:

```
setwd("C:\\Users\\Norbi\\Desktop\\studia\\sem5\\DatamininginBusiness\\topikidocclusteringu\\treasure")
wd<-"C:/Users/Norbix/Desktop/studia/sem5/DatamininginBusiness/topikidocclusteringu/treasure"
docs <- Corpus(DirSource(wd))
dtm <- DocumentTermMatrix(docs)
dtm
getwd()
filenames <- list.files(getwd(),pattern="*.txt")
filenames <-c(filenames)
filenames
rownames(dtm)
rownames(dtm)<-filenames
d1 <- dist(dtm, method="euclidian")
# make the clustering
fit <- hclust(d=d1, method="complete")
fit
plot.new()
plot(fit, hang=-1, cex=1)
```