**Want to be a Ted Speaker?**

*"At the end of my life, I want to be able to say I contributed more than I criticized."*— **Brené Brown**

To contribute to the journey of aspiring Ted Speakers, particularly my fellows at Hult SF participating in (TEDx Hult San Francisco, 2020), I used web scarping to gather English transcripts of all Ted-Talks available on Tedtalk.com. While scraping, the elements extracted were Title of Talk, Speaker, Duration, Postviews and the transcript of the talk.

As on the date of scraping, Feb 3, 2020, 3850 Ted-Talks were available. They averaged 11.8 minutes with the shortest one being 1 minute ('Ode to the Only Black Kid in the Class') short and the longest being 47.55 ('Political common ground in a polarized United States') minutes. The popularity and reach of Ted-Talks could be gauged from the fact that average views per talk are over 2 million and the most viewed talk ('Do schools kill creativity') attracted over 63 million eyeballs.

This popularity raises the question what is in their semantic structure (Romanelli et. al, 2014) that makes them so hugely popular. In quest to answers these questions I tokenized the text of each talk into words. Surprisingly one of the most frequently used word was 'I' making up 1.3% of total 6,256,575 words, closely followed by 'You' making up 1.26%. This indicates a pattern of personal references may be in terms of personal stories and narratives and an attempt to connect with the audience by directly referring to them. It is in sharp contrast to other popular mediums like podcasts .

Next, all stop words and some custom words (laughter, applause, and combination of meaningless 2 letter words) were removed and the following word-cloud was plotted.


*Fig 1: Word-cloud of ted talks without stop words*

It is evident from the word-cloud that speakers most often use ‘people’, ‘time’, and ‘world’ in their narratives. However numerically, ‘people’ constitutes only 0.78% of cleaned words. This is not only indicative of misleading nature of word-clouds but also of the fact that ted-talks seem to be spanning across tons of topics with few overlapping words.

Then I looked at correlation among titles and though some titles (‘An 11-year -old’s magical and violin’ and ‘A modern take on piano, violin, cello’) turned out have perfect correlation, most titles were barely correlated. The talks that seemed to be correlated were mostly talks in different languages or years apart in time but on similar themes translated to English.

This motivated the question whether there is some hidden underlying commonality in the talks? To answer this, I used LDA to conduct topic-modelling. Since, there are 6 supposed categories on the Ted website, I used k= 6 in the function LDA. Could ted-talks be clustered in natural groups? Yes and No.

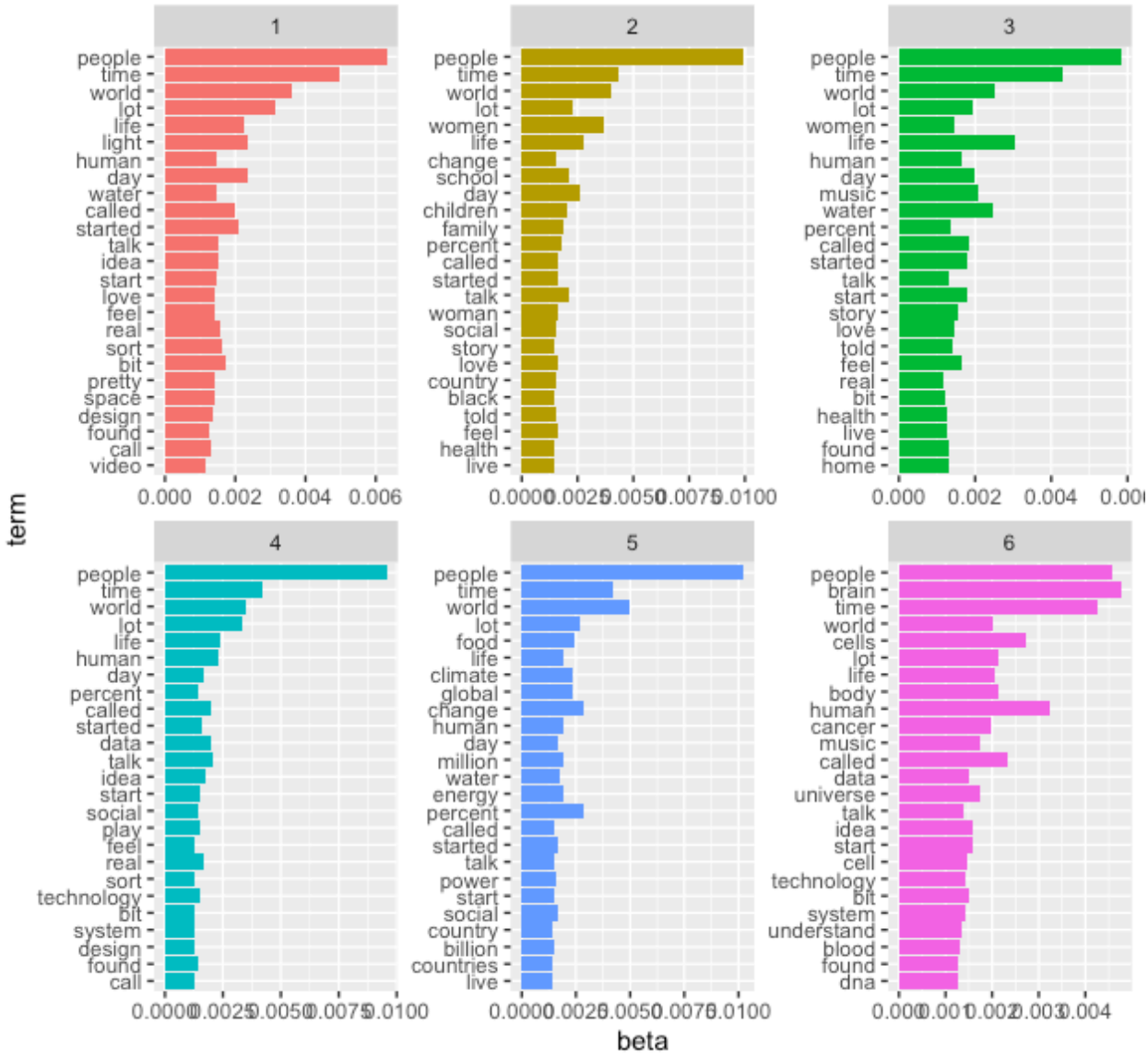



*Figure 2: Topic modelling to extract natural clusters of ted talks*

Yes, because some topics stand out as explicit categories. For example, the bars in 'pink' have terms like 'brain', 'cells', 'cancer' with high betas as compared to other graphs. Thus, they could be seen as cluster of talks focusing on 'science'.

To validate the claim, I reverse engineered them based on gamma to see if the classifications were accurate, I found the algorithm did a decent job. The titles for pink cluster included 'The biology of gender from DNA to brain'. Similarly, blue graph has terms like climate, global, change, water; clearly, they seem to be talking about climate change. The talks in these clusters included 'Visualizing climate change through space and time'.

The yellow graph included words related to social issues: black, women, love, family.

No, because the graphs with not so clear patterns had topics ranging from ‘What’s it like to be a woman in Hollywood ‘ to ‘This is what democracy looks like’. This is in conformity with Ted website; most talks fall under more than 2 categories when we manually filter for talks on the site. Importantly, this finding reinforces the power of Ted-talks. Each talk seems to be a confluence of diverse ideas in  one compact package. Indeed, Ted-Talks are known for sparking interdisciplinary debates.

Now that their individuality is established, it is important to explore similarities and differences in their semantics. Then, using ‘afinn’ lexicon, average score of words for every sentence was computed. Figure 3  indicates that largely the words used were positive with few sentences having extremely negative words. However, talks have varying number of sentences, so the analysis may be good to see average word score and sentiment for a particular sentence number but not for tracking sentiment progression.

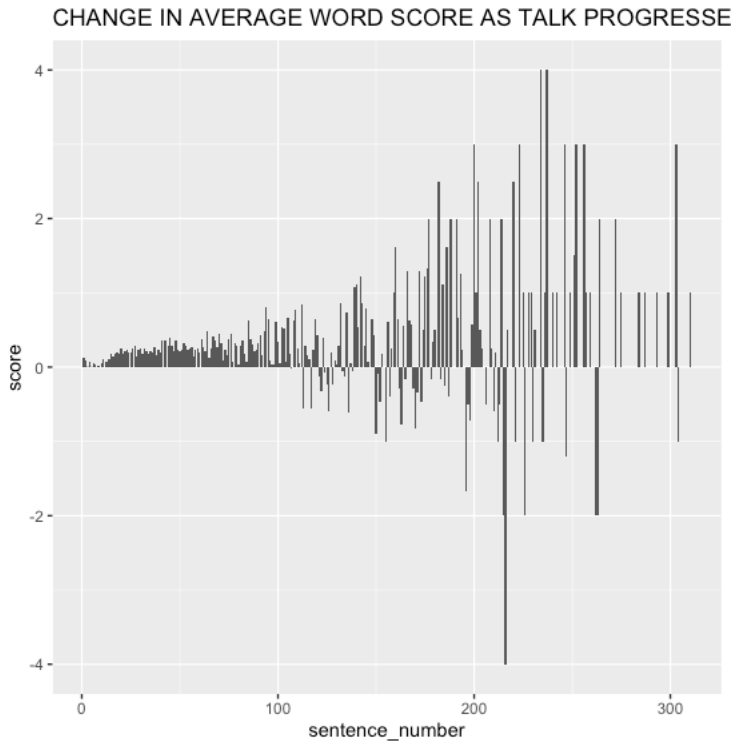



*Figure 3: Average word score sentence number wise*

Additionally, words by themselves are misleading and talk length needs to be normalized, hence the transcripts were tokenized by sentence. The aim is to analyze the progression of sentiments as the talks progress. Therefore, for each of the 3850 talks, sentences were numbered and based on the number of sentences in every talk, $10^{th}$, $20^{th}$,…., $100^{th}$ percentile points were marked. Then, all the sentences falling in $10^{th}$ percentile were grouped and the process was repeated for every consecutive 10 percentiles from $20^{th}$ to $100^{th}$ percentile.

Using the 'bing' lexicon we find that net sentiment (positive-negative) is initially negative becoming more and more negative till the $40^{th}$ percentile (Figure 4) of sentences and then slowly begins to become positive and by the end of talk people are left on high note. Well this pattern could mean that on an average ted talks begin with a problem, a grappling issue and by the end the speaker suggests a solution.

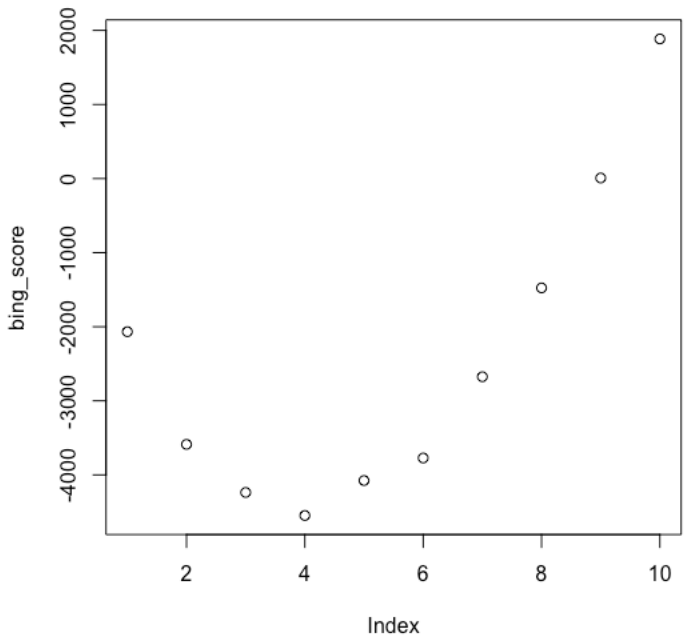



*Figure 4:Change in sentiment score as talks progress using 'bing'*

However, considering just two sentiments in a dataset comprising vast pool of topics seems a bit restrictive. Hence, 'nrc' lexicon was used to find a pattern in the narrative of talks. Figure5 indicates that positive words dominate the discourse throughout the talk taking a dip at around 30% of talk. It is around the same time that proportion of joy related words drop and fear related

words increase. From 30th to 80th percentile, the relative proportions of different emotions seem consistent. However, when about 80% of talk is over the proportion of negative words significantly drops, whereas proportion of joy, anticipation and positivity increase. Both lexicons reinforce the problem-solution structure of talks.

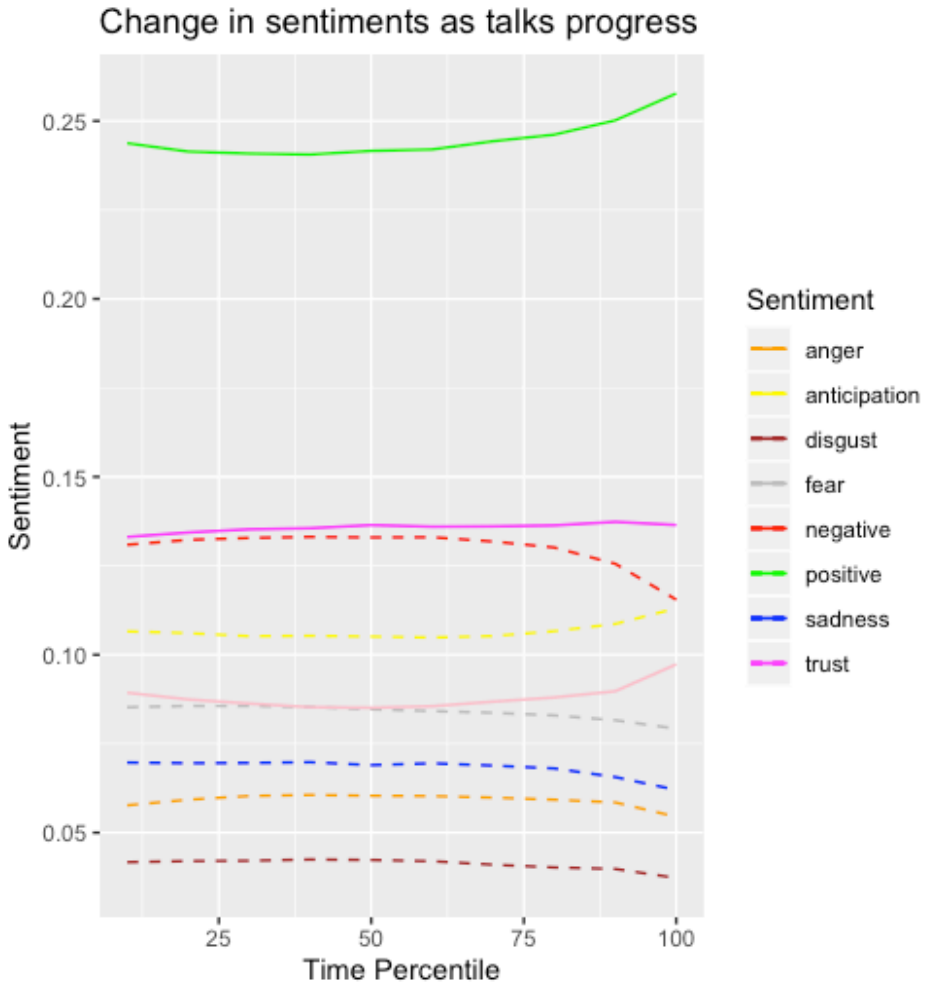



*Figure 5: Change in sentiments as the talks progress*

The burning question from these findings is that are these patterns consistent across most-viewed and least-viewed talks? For the purpose dataset was divided into two groups- least-viewed- 10thpercentile views (views<475,936) and most viewed- 90th percentile views (views > 3,899,093). Though they seemed to be similar in terms of average duration (~11 minutes), sentiments, but difference is evident in Figure 6.

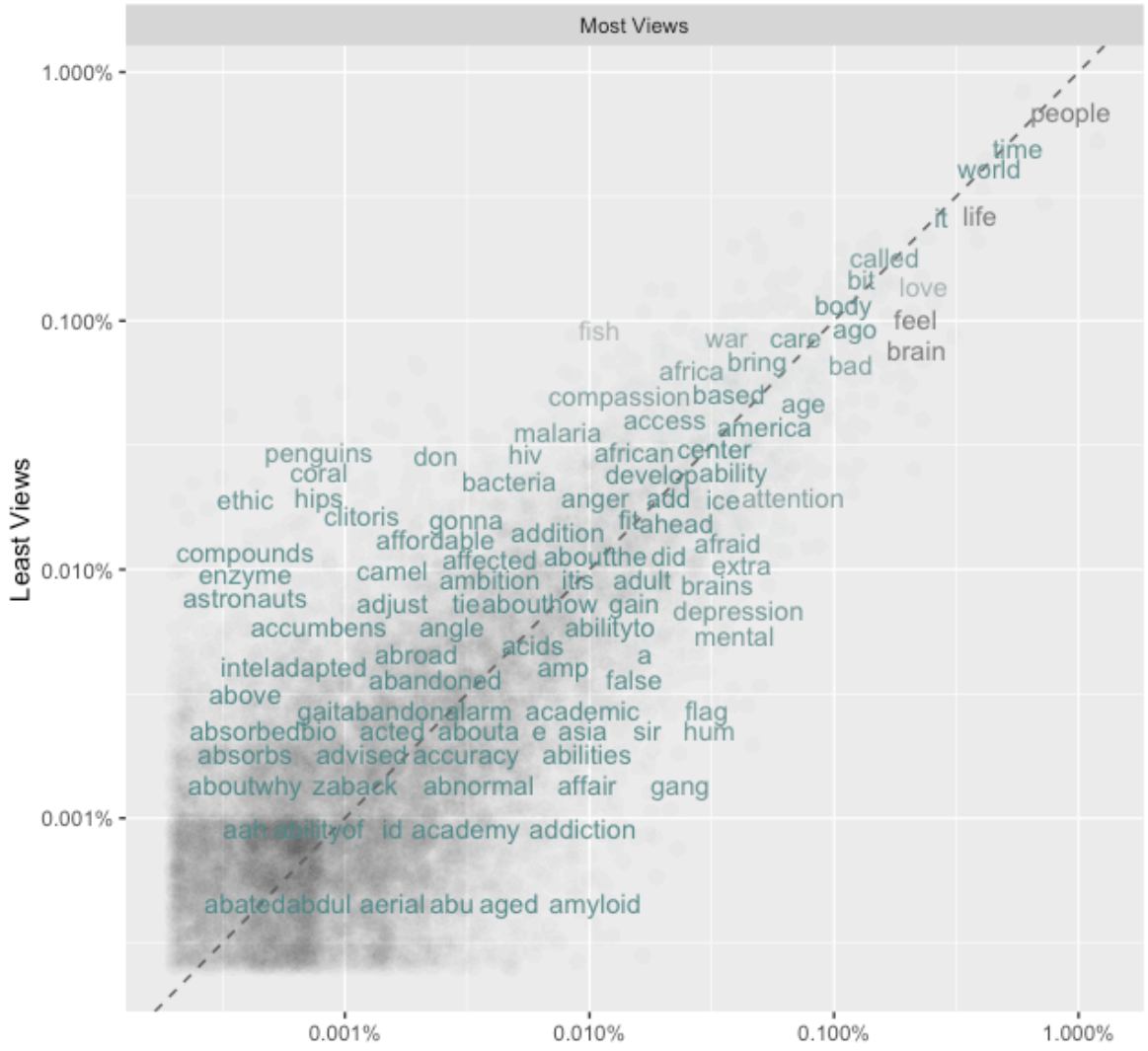



*Fig 6: Word comparison between least and most viewed talks*

Most common words (people, time, etc.) from Fig 1 are also common between most and least-viewed talks (Eg: people, closer to line). However, least-viewed topics had niche words like penguin, coral, enzyme, accumbens (I had to google it to ensure if it is even a word! (Nucleus accumbens, 2020), which could be perceived as very specific interest areas. On the contrary, most-viewed talks had words as abilities, academic, depression which seemed to be of interest to wider public being more personal and partly due to increasing calls towards mental health issues and happiness drives.

These apparent differences made me curious to explore are the less popular talks semantically structurally different as well.

It is interesting to look at Fig 7a) tracks sentiments of least-viewed talks and Fig 7b) tracks of most-viewed talks. Least-viewed commence on a more positive note than most-viewed. But, for least-viewed talks, positivity decreases during the course of talk; whereas most viewed talks vary with positivity, picking up towards end. However, anticipation invoking narratives are consistently higher for most-viewed talks. Intriguingly, most-viewed talks show a dip in trust at around 60% talk and at the same time negative words suddenly pick up, gradually tapering off at end. I believe that's a major part of their success ! Set the stage, keep audience anxious, create tension (that 60$^{th}$ percentile where trust trades off for negativity), keep them hooked and end on a markedly high note.

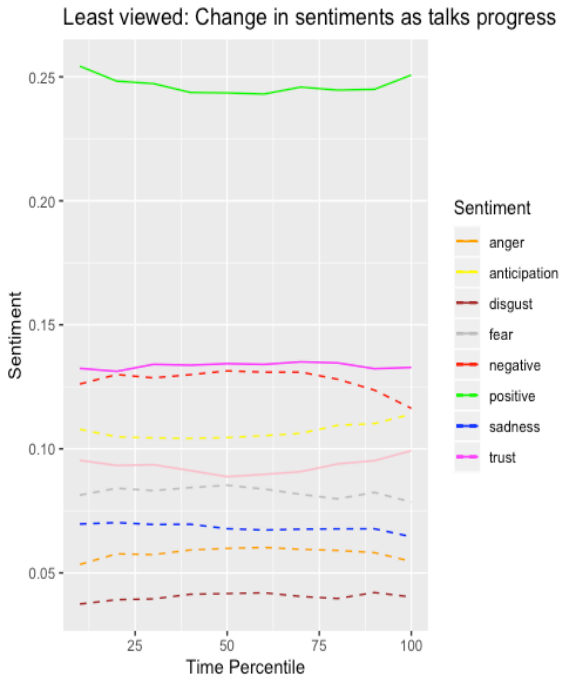



*Figure 7a: Least-viewed: changes in sentiments*

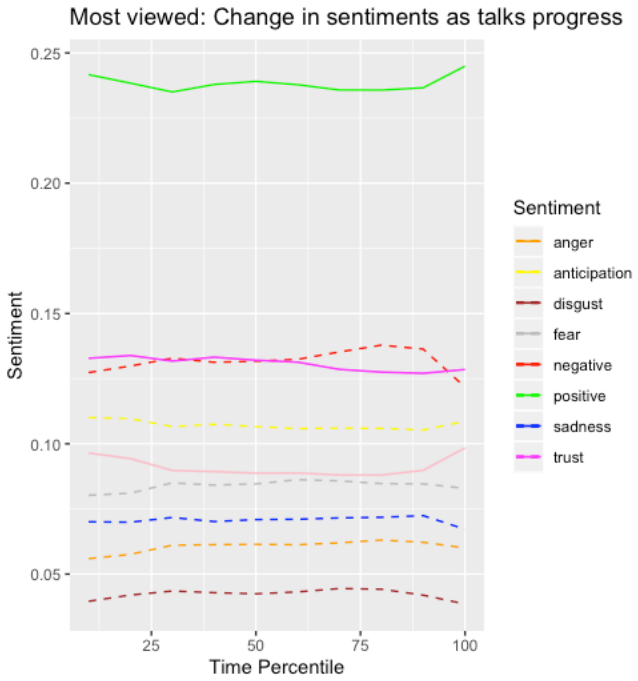



*Figure 7b: Most-viewed: changes in sentiments*

In sum, every Ted speaker is a Mozart, creating own symphony through use of specific words, narrating personal anecdotes, connecting with audience on human level by bringing up a problem , riding on the sentiment waves and ending on a high note.

References:

Nucleus accumbens. (2020, February 14). Retrieved from https://en.wikipedia.org/wiki/Nucleus_accumbens)

Romanelli, F., Cain, J., & McNamara, P. J. (2014). Should TED talks be teaching us something?. *American journal of pharmaceutical education*, *78*(6).

Silge, J., & Robinson, D. (2016). tidytext: Text mining and analysis using tidy data principles in R. *Journal of Open Source Software*, *1*(3), 37.

Tedx Hult San Francisco. (2020, February 14). Retrieved from https://www.ted.com/tedx/events/37582

# APPENDIX

```
> library(rvest)
> library(tidyverse)
> library(tidytext)
> library(tidyr)
> library(sentimentr)
> library(wordcloud)
> library(RColorBrewer)
> library(topicmodels)
> library(scales)
> library(ggplot2)
> library(topicmodels)
> url_vec_mainpage <- c() ####creating empty vector
> for(i in 1:107) { ###for loop to scrape main pages
+   url_vec_mainpage[i] <-
paste0("https://www.ted.com/talks?language=en&page=",as.character(i),"&sort=newest")
+ }
> firststep<- read_html(url_vec_mainpage[1])
> url<- firststep%>%
+   html_nodes("a.ga-link")%>%
+   html_attr("href")
> urlfinal<- str_detect(url, '=en')
> urlfinal<-url[urlfinal]
> urlnew<- unique(urlfinal)
> ## modifying url by string manipulation
> modifiedurl<- c()
> for (i in 1:length(urlnew)){
+   modifiedurl[i]<- paste(c("https://www.ted.com",urlnew[i]), collapse= "")
+ }
> modifiedurl2<- str_replace(modifiedurl, "\\?language=en","/transcript?language=en")
> modifiedurl2
 [1]
"https://www.ted.com/talks/debbie_millman_how_symbols_and_brands_shape_our_humanity/tr
anscript?language=en"
 [2]
"https://www.ted.com/talks/noelle_martin_online_predators_spread_fake_porn_of_me_here_s_h
ow_i_fought_back/transcript?language=en"
 [3]
"https://www.ted.com/talks/antara_raychaudhuri_and_iseult_gillespie_the_legend_of_annapurna
_hindu_goddess_of_nourishment/transcript?language=en"
 [4]
"https://www.ted.com/talks/victoria_gill_what_a_nun_can_teach_a_scientist_about_ecology/tran
script?language=en"
```

[5]
"https://www.ted.com/talks/lisa_godwin_how_teachers_can_help_students_navigate_trauma/transcript?language=en"
[6]
"https://www.ted.com/talks/alex_gendler_epic_engineering_building_the_brooklyn_bridge/transcript?language=en"
[7]
"https://www.ted.com/talks/amane_dannouni_how_online_marketplaces_can_help_local_economies_not_hurt_them/transcript?language=en"
[8]
"https://www.ted.com/talks/allison_ramsey_and_mary_staicu_the_accident_that_changed_the_world/transcript?language=en"
[9]
"https://www.ted.com/talks/alicia_eggert_imaginative_sculptures_that_explore_how_we_perceive_reality/transcript?language=en"
[10]
"https://www.ted.com/talks/robert_reffkin_5_ways_to_create_stronger_connections/transcript?language=en"
[11]
"https://www.ted.com/talks/chieh_huang_how_to_know_if_it_s_time_to_change_careers/transcript?language=en"
[12]
"https://www.ted.com/talks/emily_oster_3_things_new_parents_should_consider_before_going_back_to_work/transcript?language=en"
[13]
"https://www.ted.com/talks/thasunda_duckett_6_ways_to_improve_your_relationship_with_money/transcript?language=en"
[14]
"https://www.ted.com/talks/leeann_renninger_the_secret_to_giving_great_feedback/transcript?language=en"
[15]
"https://www.ted.com/talks/liz_fosslien_how_to_embrace_emotions_at_work/transcript?language=en"
[16]
"https://www.ted.com/talks/rahaf_harfoush_how_burnout_makes_us_less_creative/transcript?language=en"
[17]
"https://www.ted.com/talks/patrick_mcginnis_how_to_make_faster_decisions/transcript?language=en"
[18]
"https://www.ted.com/talks/alex_gendler_everything_changed_when_the_fire_crystal_got_stolen/transcript?language=en"
[19]
"https://www.ted.com/talks/lucy_king_how_bees_can_keep_the_peace_between_elephants_and_humans/transcript?language=en"

[20]
"https://www.ted.com/talks/smruti_jukur_johari_what_if_the_poor_were_part_of_city_planning/transcript?language=en"
[21]
"https://www.ted.com/talks/jennifer_vail_the_science_of_friction_and_its_surprising_impact_on_our_lives/transcript?language=en"
[22]
"https://www.ted.com/talks/matthew_a_wilson_the_health_benefits_of_clowning_around/transcript?language=en"
[23]
"https://www.ted.com/talks/jay_van_bavel_do_politics_make_us_irrational/transcript?language=en"
[24]
"https://www.ted.com/talks/paul_mceuen_and_marc_miskin_tiny_robots_with_giant_potential/transcript?language=en"
[25]
"https://www.ted.com/talks/lisa_janae_bacon_the_life_legacy_assassination_of_an_african_revolutionary/transcript?language=en"
[26]
"https://www.ted.com/talks/david_ikard_the_real_story_of_rosa_parks_and_why_we_need_to_confront_myths_about_black_history/transcript?language=en"
[27]
"https://www.ted.com/talks/rayma_suprani_dictators_hate_political_cartoons_so_i_keep_drawing_them/transcript?language=en"
[28]
"https://www.ted.com/talks/alex_rosenthal_the_chasm_think_like_a_coder_ep_6/transcript?language=en"
[29]
"https://www.ted.com/talks/ellen_agler_parasitic_worms_hold_back_human_progress_here_s_how_we_can_end_them/transcript?language=en"
[30]
"https://www.ted.com/talks/sylvain_duranton_how_humans_and_ai_can_work_together_to_create_better_businesses/transcript?language=en"
[31]
"https://www.ted.com/talks/jessica_ochoa_hendrix_how_virtual_reality_turns_students_into_scientists/transcript?language=en"
[32]
"https://www.ted.com/talks/christopher_bahl_a_new_type_of_medicine_custom_made_with_tiny_proteins/transcript?language=en"
[33]
"https://www.ted.com/talks/kenny_coogan_licking_bees_and_pulping_trees_the_reign_of_a_wasp_queen/transcript?language=en"
[34]
"https://www.ted.com/talks/melody_smith_how_bones_make_blood/transcript?language=en"

```
[35]
"https://www.ted.com/talks/werner_reich_how_the_magic_of_kindness_helped_me_survive_the
_holocaust/transcript?language=en"
[36]
"https://www.ted.com/talks/angelicque_white_what_ocean_microbes_reveal_about_the_changin
g_climate/transcript?language=en"
> all_pages <- c()
> for(i in 1:107){
+   firststep<- read_html(url_vec_mainpage[i])
+   url<- firststep%>%
+     html_nodes("a.ga-link")%>%
+     html_attr("href")
+   urlfinal<- str_detect(url, '=en')
+   urlfinal
+   urlfinal<-url[urlfinal]
+   urlfinal
+   urlnew<- unique(urlfinal)
+   modifiedurl<- c()
+   for (i in 1:length(urlnew)){
+     modifiedurl[i]<- paste(c("https://www.ted.com",urlnew[i]), collapse= "")
+   }
+   modifiedurl2<- str_replace(modifiedurl, "\\?language=en","/transcript?language=en")
+   all_pages <- append(all_pages, modifiedurl2)
+ }
> all_pages
 [1]
"https://www.ted.com/talks/debbie_millman_how_symbols_and_brands_shape_our_humanity/tr
anscript?language=en"
 [2]
"https://www.ted.com/talks/noelle_martin_online_predators_spread_fake_porn_of_me_here_s_h
ow_i_fought_back/transcript?language=en"
 [3]
"https://www.ted.com/talks/antara_raychaudhuri_and_iseult_gillespie_the_legend_of_annapurna
_hindu_goddess_of_nourishment/transcript?language=en"
 [4]
"https://www.ted.com/talks/victoria_gill_what_a_nun_can_teach_a_scientist_about_ecology/tran
script?language=en"


> finaltext<- c()
> Postviews <- c()
> duration <- c()
> speaker <- c()
> titles <- c()
> sentence<- c()
> ##for loop to extract releavnt variables from each talk
```

```
> for (i in 1:length(all_pages)) {
+   try({
+     text<- read_html(all_pages[i])%>%
+       html_nodes("div.Grid__cell p")
+
+     text2<-str_replace_all(text,"[\r\n\t]" , "")
+     text2<- str_replace_all(text2, "<p>", "")
+     text2<- str_replace_all(text2, "</p>", "")
+
+     finaltext[i]<-paste(text2, collapse=" ")
+     for (i in 1:length(finaltext)){
+       sentence[i]<- unlist(strsplit(finaltext[i], "\\."))
+     }
+
+     Views<- read_html(all_pages[i])%>%
+       html_nodes("div.Grid__cell span")
+     Views2<-str_replace_all(Views,"[\r\n\t]" , "")
+     Postviews[i] <- str_remove_all(Views2[1],"[^0-9]")
+     Postviews[i] <- str_remove(Postviews[i], '[0-9]{3}$')
+
+     time<- "([0-9]+)[:]([0-9]{2})"
+     duration[i] <- str_extract(Views[3], time)
+     duration[i]<- (as.double(str_extract(duration[i], "[0-9]+"))) * 60 +
(as.double(str_extract(duration[i], "[0-9]{2}$")))
+
+     speaker_ext<- read_html(all_pages[i])%>%
+       html_nodes("title")
+     speaker_ext<- speaker_ext[1]
+     speaker_ext<- str_remove_all(speaker_ext, "<title>")
+     speaker[i] <- str_extract(speaker_ext, "([A-z]+)[ ]([:alpha:]+)")
+
+     title<-speaker_ext[1]
+     title<- str_remove(speaker_ext, "([A-z]+)[ ]([:alpha:]+[:])")
+     titles[i] <- str_trim(str_remove(title, "[|].*"))
+   })
+ }
There were 50 or more warnings (use warnings() to see the first 50)
> ##creating a dataframe to include title, speaker, duration, views and transcript of each talk
> tedtalk_df<- data.frame(titles, speaker, duration, Postviews, finaltext,
stringsAsFactors=FALSE)
> str(tedtalk_df)
'data.frame':    3850 obs. of  5 variables:
 $ titles   : chr  "How symbols and brands shape our humanity" "Online predators spread fake
porn of me. Here's how I fought back" "Antara Raychaudhuri and  The legend of Annapurna,
Hindu goddess of nourishment" "What a nun can teach a scientist about ecology" ...
 $ speaker  : chr  "Debbie Millman" "Noelle Martin" "Antara Raychaudhuri" "Victoria Gill" ...
```

```
 $ duration : chr  "852" "706" "280" "839" ...
 $ Postviews: chr  "165162" "40695" "0" "532800" ...


 $ finaltext: chr  "Thirteen point eight billion years ago,the universe as we know itbegan with a
big bang,and everything that we k"| __truncated__ "[This talk contains graphic languageand
descriptions of sexual abuse] Can I get a show of handswho here has eve"| __truncated__ "Lord
Shiva— primordial destroyer of evil,slayer of demons,protector, and omniscient observer of the
universe—wa"| __truncated__ "OK, I would like to introduce all of youbeautiful, curious-
minded peopleto my favorite animal in the world.This"| __truncated__ ...
> tedtalk_df$titles<- as.character(tedtalk_df$titles)
> tedtalk_df$speaker<- as.character(tedtalk_df$speaker)
> tedtalk_df$finaltext<- as.character(tedtalk_df$finaltext)
> tedtalk_df$duration <- as.numeric(tedtalk_df$duration)
> tedtalk_df$Postviews<- as.numeric(tedtalk_df$Postviews)
> str(tedtalk_df)
'data.frame':    3850 obs. of  5 variables:
 $ titles   : chr  "How symbols and brands shape our humanity" "Online predators spread fake
porn of me. Here's how I fought back" "Antara Raychaudhuri and  The legend of Annapurna,
Hindu goddess of nourishment" "What a nun can teach a scientist about ecology" ...
 $ speaker  : chr  "Debbie Millman" "Noelle Martin" "Antara Raychaudhuri" "Victoria Gill" ...
 $ duration : num  852 706 280 839 920 292 747 276 629 173 ...
 $ Postviews: num  165162 40695 0 532800 494988 ...
 $ finaltext: chr  "Thirteen point eight billion years ago,the universe as we know itbegan with a
big bang,and everything that we k"| __truncated__ "[This talk contains graphic languageand
descriptions of sexual abuse] Can I get a show of handswho here has eve"| __truncated__ "Lord
Shiva— primordial destroyer of evil,slayer of demons,protector, and omniscient observer of the
universe—wa"| __truncated__ "OK, I would like to introduce all of youbeautiful, curious-
minded peopleto my favorite animal in the world.This"| __truncated__ ...

#### saving the scrapped data in csv for further analysis
write.csv2(tedtalk_df, file= 'tedtalkfilecsv.csv')

##########----------ANALYSIS----------
## I. Summary
data2 <- read.csv2('/Users/mvs/Desktop/Ted Talk/tedtalkfilecsv.csv')
data2$titles<- as.character(data2$titles)
daat2$speaker<- as.character(data2$speaker)
data2$finaltext<- as.character(data2$finaltext)
data2$duration <- as.numeric(data2$duration)
data2$Postviews<- as.numeric(data2$Postviews)
str(data2)

max(data2$duration, na.rm = TRUE)
[1] 2853
```

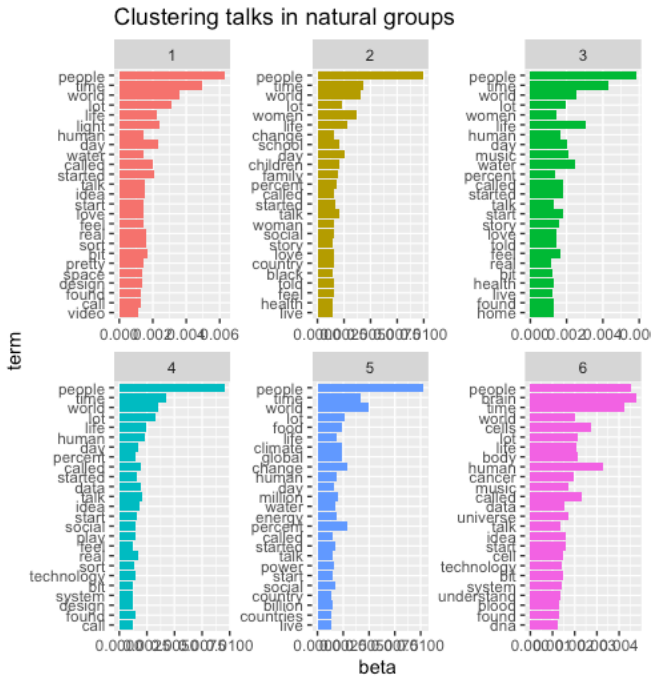
```
 data2[which(data2$duration== 2853),names(data2) %in% c("titles","speaker", "duration" ,
"Postviews" ,"finaltext")]
```
titles speaker duration Postviews
1248 Gretchen Carlson, Political common ground in a polarized United States Gretchen Carlson
2853 1041140

```
> ## II. ----------------------------------Finding words frequency-----------------------------------------
---
> ######## customized stop words to remove two letter wired words
> data2_unnest<- data2%>%
+   unnest_tokens(word, finaltext)
> View(data2_unnest)
>
> word_counts<- data2_unnest%>%
+   anti_join(stop_words)%>%
+   count(titles, word, sort=TRUE)%>%
+   ungroup()
Joining, by = "word"
>
> two_letter_df<- word_counts%>%
+   filter(nchar(word)==2)
> two_letter_words<- two_letter_df$word
> # making data frame for 2 letter words to remove as custom stop words
> two_letter_words_df<- data_frame(word= two_letter_words, lexicon= c("custom"))
> custom_stop_words<-bind_rows(data_frame(word=c("applause",
"laughter"),lexicon=c("custom", "custom")),two_letter_words_df, stop_words)
>
##III------------------------Word cloud-----------------------------------------------------------------
colorlist = c("green","blue","pink","yellow","orange","purple")

word_cloud_ted<- data2_unnest%>%
  anti_join(custom_stop_words)%>%
  count(word)%>%
  with(wordcloud(word, n, max.words= 75, colors = colorlist))
```


```
##############################################################################
####Exploring correlations
> title_words<- data2%>%
+   unnest_tokens(word, finaltext)%>%
+   anti_join(stop_words)%>%
+   count(titles,word, sort=TRUE)
Joining, by = "word"
> ungroup()
Error in UseMethod("ungroup") :
  no applicable method for 'ungroup' applied to an object of class "NULL"
>
> total_words<- title_words%>%
+   group_by(titles)%>%
+   summarize(total= sum(n))
>
> title_words<- left_join(title_words, total_words)
Joining, by = "titles"
>
> title_words<-title_words%>%
+   bind_tf_idf(word, titles, n)%>%
+   arrange(desc(tf_idf))
>
> ###pairwise corr2el
>
> install.packages('widyr')
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.6/widyr_0.1.2.tgz'
Content type 'application/x-gzip' length 246602 bytes (240 KB)
```

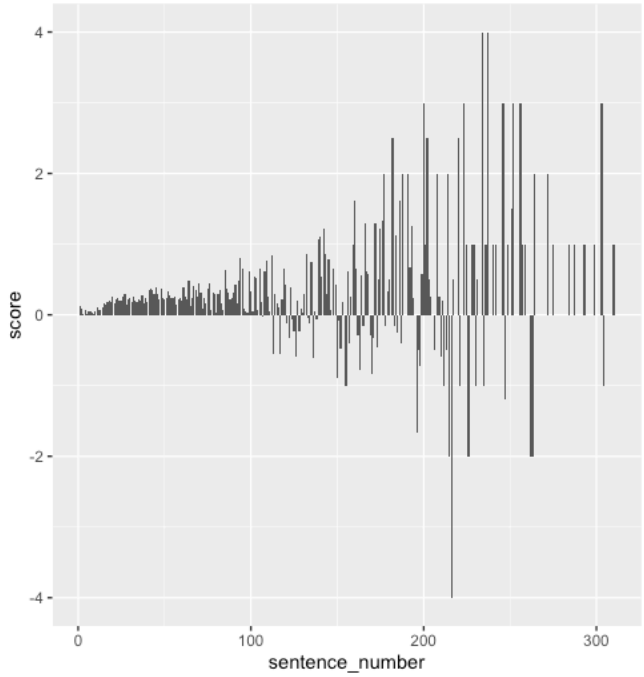
```
==========================================================
downloaded 240 KB


The downloaded binary packages are in
        /var/folders/z7/9cqygcv10zv8jr2b57_kb3p40000gn/T//RtmpS2hsZD/downloaded_packag
es
> library(widyr)
>
> title_corr<- title_words%>%
+   pairwise_cor(titles, word, n, sort= TRUE)
> title_corr
# A tibble: 14,818,650 x 3
   item1                                item2                                correlation
   <chr>                                <chr>                                      <dbl>
 1 An 11-year-old prodigy performs old-school jazz       Dancing with light
1.000
 2 "Robert Gupta +  On violin and cello, \"Passacaglia\… Dancing with light
1.000
 3 A dance in a hurricane of paper, wind and light       Dancing with light
1.000
 4 Dancing with light                                    An 11-year-old prodigy performs old-school jazz
1.000
 5 "Robert Gupta +  On violin and cello, \"Passacaglia\… An 11-year-old prodigy performs old-
school jazz           1.000
 6 A dance in a hurricane of paper, wind and light       An 11-year-old prodigy performs old-
school jazz           1.000
 7 Dancing with light                                    "Robert Gupta +  On violin and cello,
\"Passacaglia…       1.000
 8 An 11-year-old prodigy performs old-school jazz       "Robert Gupta +  On violin and cello,
\"Passacaglia…       1.000
 9 A dance in a hurricane of paper, wind and light       "Robert Gupta +  On violin and cello,
\"Passacaglia…       1.000
10 Dancing with light                                    A dance in a hurricane of paper, wind and light
1.000
# … with 14,818,640 more rows

> ## VII.-------topic modelling-----------------------------------------------------------------------
>
> word_counts<- data2_unnest%>%
+   anti_join(custom_stop_words)%>%
+   count(titles, word, sort=TRUE)%>%
+   ungroup()
Joining, by = "word"
>
> titles_dtm<- word_counts%>%
```

```
+   cast_dtm(titles,word, n)
>
> titles_lda<- LDA(titles_dtm, k=6, control = list(seed=1234))
>
> titles_topics<- tidy(titles_lda, mtarix="beta")
>
> top_terms<- titles_topics%>%
+   group_by(topic)%>%
+   top_n(25, beta)%>%
+   ungroup()%>%
+   arrange(topic,-beta)
>
> top_terms%>%
+   mutate(term= reorder(term, beta))%>%
+   ggplot(aes(term, beta, fill= factor(topic)))+
+   geom_col(show.legend=FALSE)+
+   facet_wrap(~topic, scales="free")+
+   coord_flip()+
+   ggtitle("Clustering talks in natural groups")
>
```




```
> title_gamma<- tidy(titles_lda, matrix= "gamma")
>
> title_classification<-
+   title_gamma%>%
+   group_by(document)%>%
+   top_n(1, gamma)%>%
+   ungroup()
>
```

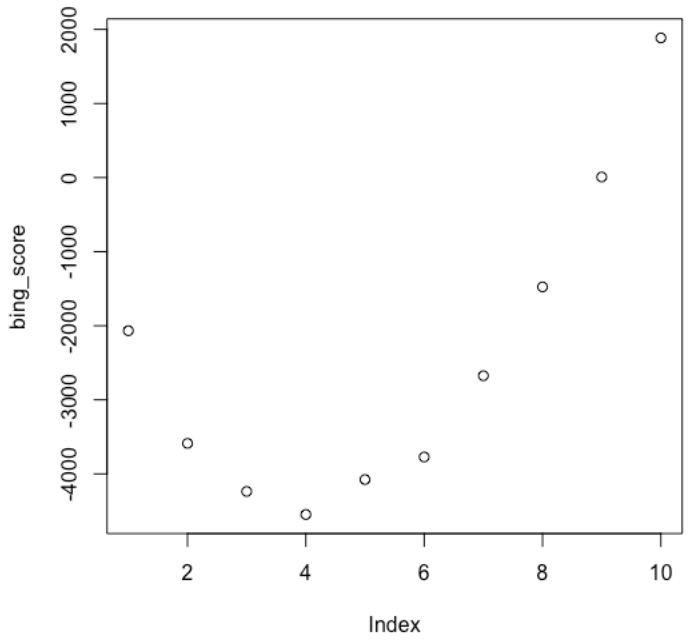
```
> grouped_title_classification<- title_classification%>%
+   group_by(topic)%>%
+   summarise(n())
>
> ##Example of topic modelling
> title_classification$document[1]
[1] "My year of saying yes to everything"

> ted_sentence<- data2%>%
+   unnest_tokens(senetence, finaltext, token='sentences')
Warning message:
Factor `speaker` contains implicit NA, consider using `forcats::fct_explicit_na`
> sentence_number_ted<- ted_sentence%>%
+   group_by(titles)%>%
+   mutate(sentencenumber=row_number())%>%
+   ungroup()
>
> sentence_number_ted_unnest<- sentence_number_ted%>%
+   unnest_tokens(word,senetence)%>%
+   anti_join(custom_stop_words)
Joining, by = "word"
>
> afinn<-sentence_number_ted_unnest%>%
+   inner_join(get_sentiments("afinn"))%>%
+   group_by(sentence_number=sentencenumber)%>%
+   summarise(score=(sum(value)/length(value)))
Joining, by = "word"
>
> afinn%>%
+   ggplot(aes(sentence_number,score))+geom_col(show.legend=FALSE)  ###not sure what it is
>
```




## V.---------- Looking beyond words----------

```
ted_sentence<- data2%>%
  unnest_tokens(senetence, finaltext, token='sentences')

sentence_number_ted<- ted_sentence%>%
  group_by(titles)%>%
  mutate(sentencenumber=row_number())%>%
  ungroup()

sentence_number_ted_unnest<- sentence_number_ted%>%
  unnest_tokens(word,senetence)%>%
  anti_join(custom_stop_words)

max_sentence_quant_bing<-c()
for(i in 1:10){
  max_sentence_quant_bing[i]<-sentence_number_ted_unnest%>%
    group_by(titles)%>%
    filter(sentencenumber<= quantile(sentencenumber,0.1*i))%>%
    filter(sentencenumber> quantile(sentencenumber,0.1*(i-1)))%>%
    ungroup()%>%
    inner_join(get_sentiments("bing"))%>%
    count(sentiment)%>%
    spread(sentiment,n,fill = 0)%>%
    mutate(sentiment=positive-negative)%>%
    select(sentiment)%>% unlist()
```

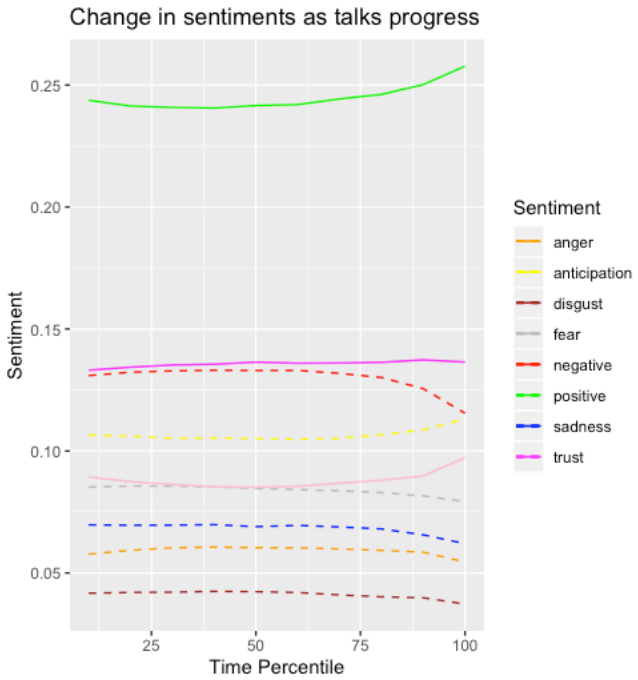
```
}
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
>
```




```
> df<- data.frame(matrix(, nrow=0, ncol=10))
> for(i in 1:10){
+   df <- rbind(df, sentence_number_ted_unnest %>%
+                 anti_join(custom_stop_words)%>%
+                 group_by(titles)%>%
+                 filter(sentencenumber<= quantile(sentencenumber,0.1*i))%>%
+                 filter(sentencenumber> quantile(sentencenumber,0.1*(i-1))-1)%>%
+                 ungroup()%>%
+                 inner_join(get_sentiments("nrc"))%>%
+                 count(sentiment) %>%
+                 select(n) %>% t())
+ }
Joining, by = "word"
```

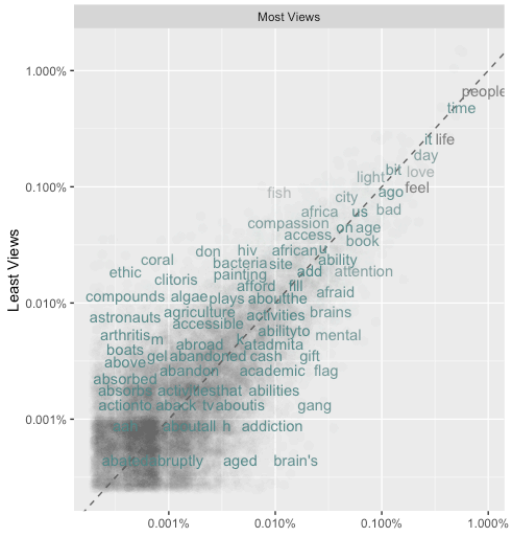
```
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
> colnames(df) <- c("anger", "anticipation",  "disgust", "fear", "joy", "negative", "positive",
"sadness", "surprise", "trust")
> df$sum<- rowSums(df)
> df$angerprop<- df$anger/df$sum
> df$anticipationprop<- df$anticipation/df$sum
> df$disgustprop<- df$disgust/df$sum
> df$fearprop<- df$fear/df$sum
> df$joyprop<- df$joy/df$sum
> df$negativeprop<- df$negative/df$sum
> df$positiveprop<- df$positive/df$sum
> df$sadnessprop<- df$sadness/df$sum
> df$surpriseprop<- df$surprise/df$sum
> df$trustprop<- df$trust/df$sum
>
> ggplot() +
+   geom_line(data = df, aes(x = seq(10, 100, by=10), y = angerprop, color =
"anger"),linetype="dashed") +
+   geom_line(data = df, aes(x = seq(10, 100, by=10), y = disgustprop, color =
"disgust"),linetype="dashed") +
+   geom_line(data = df, aes(x = seq(10, 100, by=10), y = anticipationprop, color =
"anticipation"), linetype="dashed") +
+   geom_line(data = df, aes(x = seq(10, 100, by=10), y = fearprop, color = "fear"),
linetype="dashed") +
+   geom_line(data = df, aes(x = seq(10, 100, by=10), y = joyprop, color = "joy"), color = "pink")
+
+   geom_line(data = df, aes(x = seq(10, 100, by=10), y = negativeprop, color =
"negative"),linetype="dashed") +
```

```
+   geom_line(data = df, aes(x = seq(10, 100, by=10), y = positiveprop, color = "positive")) +
+   geom_line(data = df, aes(x = seq(10, 100, by=10), y = sadnessprop, color = "sadness"),
linetype="dashed") +
+   geom_line(data = df, aes(x = seq(10, 100, by=10), y = trustprop, color = "trust")) +
+   scale_color_manual(values = c(
+     'anger' = 'orange',
+     'disgust'='brown',
+     'anticipation'= 'yellow',
+     'fear'='gray',
+     'joy'='pink',
+     'negative'='red',
+     'positive'='green',
+     'sadness'='blue',
+     'trust'='magenta'
+   )) +
+   labs(color = 'Sentiment') +
+   xlab('Time Percentile') +
+   ylab('Sentiment') +
+   ggtitle("Change in sentiments as talks progress")
>
```



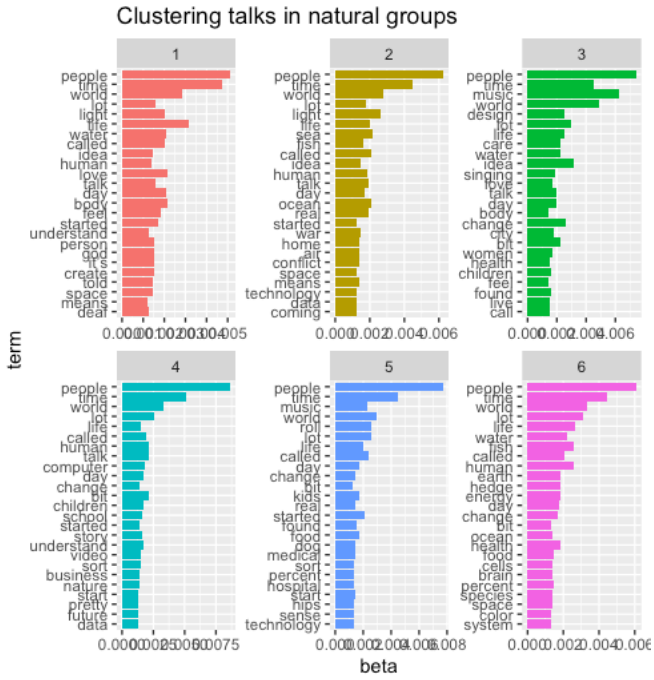

```
> least_dataset<- data2[which(data2$Postviews< 475936),names(data2) %in%
c("titles","speaker", "duration" , "Postviews" ,"finaltext")]
> least_tidy<- least_dataset%>%
+   unnest_tokens(word, finaltext)%>%
+   anti_join(custom_stop_words)
Joining, by = "word"
```

```
> most_dataset<- data2[which(data2$Postviews > 3899093),names(data2) %in%
c("titles","speaker", "duration" , "Postviews" ,"finaltext")]
> most_tidy<- most_dataset%>%
+   unnest_tokens(word, finaltext)%>%
+   anti_join(custom_stop_words)
Joining, by = "word"
> frequency <- bind_rows(mutate(least_tidy, subgroup="Least Views"),
+                        mutate(most_tidy, subgroup= "Most Views"))%>%#closing bind_rows
+   mutate(word=str_extract(word, "[a-z']+")) %>%
+   count(subgroup, word) %>%
+   group_by(subgroup) %>%
+   mutate(proportion = n/sum(n))%>%
+   select(-n) %>%
+   spread(subgroup, proportion) %>%
+   gather(subgroup, proportion, `Most Views`)
> head(frequency)
# A tibble: 6 x 4
  word   `Least Views` subgroup    proportion
  <chr>          <dbl> <chr>           <dbl>
1 '         NA         Most Views  0.0000196
2 a          0.0000499  Most Views  0.000168
3 a's       NA         Most Views  0.0000118
4 aaa       NA         Most Views  0.00000392
5 aachen    0.00000499 Most Views NA
6 aah       0.00000997 Most Views  0.00000392
> ggplot(frequency, aes(x=proportion, y=`Least Views`,
+                       color = abs(`Least Views`- proportion)))+
+   geom_abline(color="grey40", lty=2)+
+   geom_jitter(alpha=.01, size=2.5, width=0.3, height=0.3)+
+   geom_text(aes(label=word), check_overlap = TRUE, vjust=1) +
+   scale_x_log10(labels = percent_format())+
+   scale_y_log10(labels= percent_format())+
+   scale_color_gradient(limits = c(0,0.001), low = "darkslategray4", high = "gray75")+
+   facet_wrap(~subgroup, ncol=1)+
+   theme(legend.position = "none")+
+   labs(y= "Least Views", x=NULL)
Warning messages:
1: Removed 61339 rows containing missing values (geom_point).
2: Removed 61340 rows containing missing values (geom_text).
>
```




```
> least_dataset_unnest<- least_dataset%>%
+   unnest_tokens(word, finaltext)
> word_counts<- least_dataset_unnest%>%
+   anti_join(custom_stop_words)%>%
+   count(titles, word, sort=TRUE)%>%
+   ungroup()
Joining, by = "word"
> ####### tf idf
> least_tf_idf<- least_dataset_unnest%>%
+   anti_join(custom_stop_words)%>%
+   count(titles, word, sort=TRUE)
Joining, by = "word"
> ungroup()
Error in UseMethod("ungroup") :
  no applicable method for 'ungroup' applied to an object of class "NULL"
>
> least_total_words<- least_tf_idf%>%
+   group_by(titles)%>%
+   summarize(total= sum(n))
>
> least_tf_idf<- left_join(least_tf_idf, least_total_words)
Joining, by = "titles"
>
> least_tf_idf<-least_tf_idf%>%
+   bind_tf_idf(word, titles, n)%>%
+   arrange(desc(tf_idf))
> ##### topic modelling
>
> least_word_counts<- least_dataset_unnest%>%
+   anti_join(custom_stop_words)%>%
+   count(titles, word, sort=TRUE)%>%
+   ungroup()
```

```
Joining, by = "word"
>
> least_titles_dtm<- least_word_counts%>%
+   cast_dtm(titles,word, n)
>
> least_titles_lda<- LDA(least_titles_dtm, k=6, control = list(seed=1234))
>
> least_titles_topics<- tidy(least_titles_lda, mtarix="beta")
>
> least_top_terms<- least_titles_topics%>%
+   group_by(topic)%>%
+   top_n(25, beta)%>%
+   ungroup()%>%
+   arrange(topic,-beta)
>
> least_top_terms%>%
+   mutate(term= reorder(term, beta))%>%
+   ggplot(aes(term, beta, fill= factor(topic)))+
+   geom_col(show.legend=FALSE)+
+   facet_wrap(~topic, scales="free")+
+   coord_flip()+
+   ggtitle("Clustering talks in natural groups")
>
```

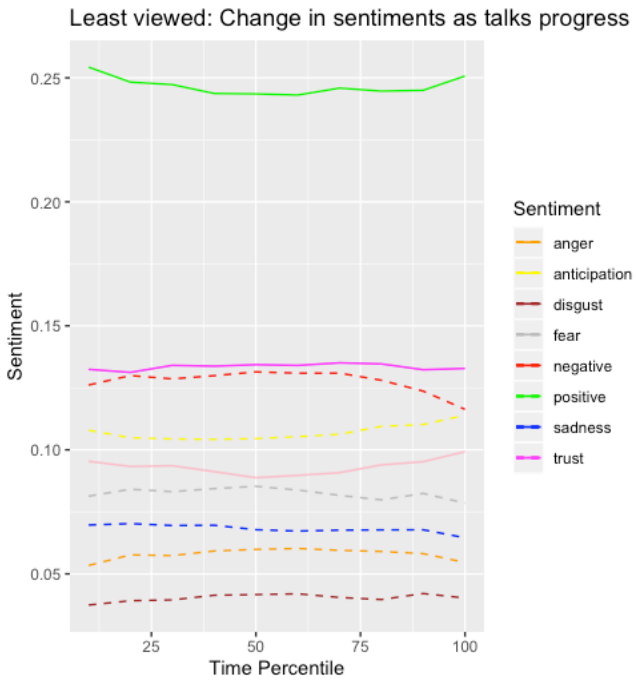



```
> least_title_gamma<- tidy(least_titles_lda, matrix= "gamma")
>
> least_title_classification<-
```

```
+   least_title_gamma%>%
+   group_by(document)%>%
+   top_n(1, gamma)%>%
+   ungroup()
>
> least_grouped_title_classification<- least_title_classification%>%
+   group_by(topic)%>%
+   summarise(n())
>
> ted_sentence_least<- least_dataset%>%
+   unnest_tokens(senetence, finaltext, token='sentences')
>
> sentence_number_ted_least<- ted_sentence_least%>%
+   group_by(titles)%>%
+   mutate(sentencenumber=row_number())%>%
+   ungroup()
>
> sentence_number_ted_unnest_least<- sentence_number_ted_least%>%
+   unnest_tokens(word,senetence)%>%
+   anti_join(custom_stop_words)
Joining, by = "word"
>
> df_least<- data.frame(matrix(, nrow=0, ncol=10))
>
>
> for(i in 1:10){
+   df_least <- rbind(df_least, sentence_number_ted_unnest_least %>%
+               anti_join(custom_stop_words)%>%
+               group_by(titles)%>%
+               filter(sentencenumber<= quantile(sentencenumber,0.1*i))%>%
+               filter(sentencenumber> quantile(sentencenumber,0.1*(i-1))-1)%>%
+               ungroup()%>%
+               inner_join(get_sentiments("nrc"))%>%
+               count(sentiment) %>%
+               select(n) %>% t())
+ }
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
```

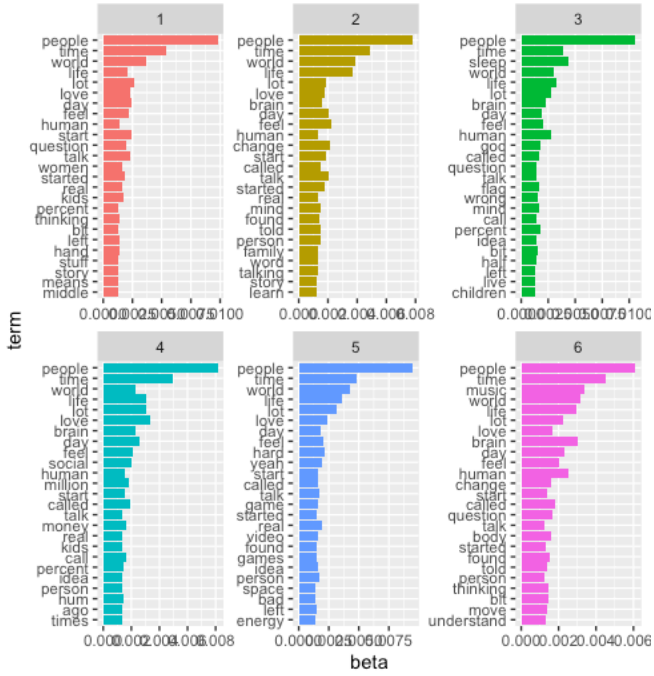
```
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
> colnames(df_least) <- c("anger", "anticipation",  "disgust", "fear", "joy", "negative", "positive",
"sadness", "surprise", "trust")
> df_least$sum<- rowSums(df_least)
> df_least$angerprop<- df_least$anger/df_least$sum
> df_least$anticipationprop<- df_least$anticipation/df_least$sum
> df_least$disgustprop<- df_least$disgust/df_least$sum
> df_least$fearprop<- df_least$fear/df_least$sum
> df_least$joyprop<- df_least$joy/df_least$sum
> df_least$negativeprop<- df_least$negative/df_least$sum
> df_least$positiveprop<- df_least$positive/df_least$sum
> df_least$sadnessprop<- df_least$sadness/df_least$sum
> df_least$surpriseprop<- df_least$surprise/df_least$sum
> df_least$trustprop<- df_least$trust/df_least$sum
>
> ggplot() +
+   geom_line(data = df_least, aes(x = seq(10, 100, by=10), y = angerprop, color =
"anger"),linetype="dashed") +

+   geom_line(data = df_least, aes(x = seq(10, 100, by=10), y = disgustprop, color =
"disgust"),linetype="dashed") +
+   geom_line(data = df_least, aes(x = seq(10, 100, by=10), y = anticipationprop, color =
"anticipation"), linetype="dashed") +
+   geom_line(data = df_least, aes(x = seq(10, 100, by=10), y = fearprop, color = "fear"),
linetype="dashed") +
+   geom_line(data = df_least, aes(x = seq(10, 100, by=10), y = joyprop, color = "joy"), color =
"pink") +
+   geom_line(data = df_least, aes(x = seq(10, 100, by=10), y = negativeprop, color =
"negative"),linetype="dashed") +
+   geom_line(data = df_least, aes(x = seq(10, 100, by=10), y = positiveprop, color = "positive"))
+
+   geom_line(data = df_least, aes(x = seq(10, 100, by=10), y = sadnessprop, color = "sadness"),
linetype="dashed") +
+   geom_line(data = df_least, aes(x = seq(10, 100, by=10), y = trustprop, color = "trust")) +
+   scale_color_manual(values = c(
+     'anger' = 'orange',
+     'disgust'='brown',
```

```
+     'anticipation'= 'yellow',
+     'fear'='gray',
+     'joy'='pink',
+     'negative'='red',
+     'positive'='green',
+     'sadness'='blue',
+     'trust'='magenta'
+   )) +
+   labs(color = 'Sentiment') +
+   xlab('Time Percentile') +
+   ylab('Sentiment') +
+   ggtitle("Least viewed: Change in sentiments as talks progress")
>
```




```
> most_dataset_unnest<- most_dataset%>%
+   unnest_tokens(word, finaltext)
> ### ########################################word freq
> most_dataset_unnest<- most_dataset%>%
+   unnest_tokens(word, finaltext)
> ####### tf idf
> most_tf_idf<- most_dataset_unnest%>%
+   anti_join(custom_stop_words)%>%
+   count(titles,word, sort=TRUE)
Joining, by = "word"
> ungroup()
Error in UseMethod("ungroup") :
  no applicable method for 'ungroup' applied to an object of class "NULL"
>
> most_total_words<- most_tf_idf%>%
```

```
+   group_by(titles)%>%
+   summarize(total= sum(n))
>
> most_tf_idf<- left_join(most_tf_idf, most_total_words)
Joining, by = "titles"
>
> most_tf_idf<-most_tf_idf%>%
+   bind_tf_idf(word, titles, n)%>%
+   arrange(desc(tf_idf))
>
> ##### topic modelling
>
> most_word_counts<- most_dataset_unnest%>%
+   anti_join(custom_stop_words)%>%
+   count(titles, word, sort=TRUE)%>%
+   ungroup()
Joining, by = "word"
>
> most_titles_dtm<- most_word_counts%>%
+   cast_dtm(titles,word, n)
>
> most_titles_lda<- LDA(most_titles_dtm, k=6, control = list(seed=1234))
>
> most_titles_topics<- tidy(most_titles_lda, mtarix="beta")
>
> most_top_terms<- most_titles_topics%>%
+   group_by(topic)%>%
+   top_n(25, beta)%>%
+   ungroup()%>%
+   arrange(topic,-beta)
>
> most_top_terms%>%
+   mutate(term= reorder(term, beta))%>%
+   ggplot(aes(term, beta, fill= factor(topic)))+
+   geom_col(show.legend=FALSE)+
+   facet_wrap(~topic, scales="free")+
+   coord_flip()
```

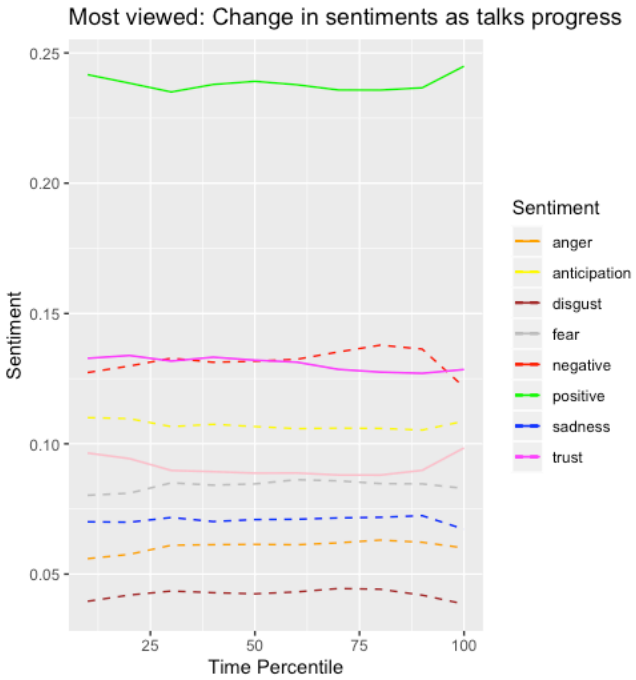



```
> most_title_gamma<- tidy(most_titles_lda, matrix= "gamma")
>
> most_title_classification<-
+   most_title_gamma%>%
+   group_by(document)%>%
+   top_n(1, gamma)%>%
+   ungroup()
>
> most_title_classification
# A tibble: 385 x 3
   document                                  topic gamma
   <chr>                                     <int> <dbl>
 1 Why we make bad decisions                     1 1.000
 2 What makes us feel good about our work?       1 1.000
 3 Can we eat to starve cancer?                  1 1.000
 4 Why work doesn't happen at work               1 1.000
 5 How to gain control of your free time         1 1.000
 6 What's wrong with what we eat                 1 1.000
 7 Which country does the most good for the world?  1 1.000
 8 Why the universe seems so strange             1 1.000
 9 Why we have too few women leaders             1 1.000
10 We need to talk about an injustice            1 1.000
# … with 375 more rows
>
> most_grouped_title_classification<- most_title_classification%>%
+   group_by(topic)%>%
+   summarise(n())
```

```
>
> ted_sentence_most<- most_dataset%>%
+   unnest_tokens(senetence, finaltext, token='sentences')
>
> sentence_number_ted_most<- ted_sentence_most%>%
+   group_by(titles)%>%
+   mutate(sentencenumber=row_number())%>%
+   ungroup()
>
> sentence_number_ted_unnest_most<- sentence_number_ted_most%>%
+   unnest_tokens(word,senetence)%>%
+   anti_join(custom_stop_words)
Joining, by = "word"
>
> df_most<- data.frame(matrix(, nrow=0, ncol=10))
> for(i in 1:10){
+   df_most <- rbind(df_most, sentence_number_ted_unnest_most %>%
+                    anti_join(custom_stop_words)%>%
+                    group_by(titles)%>%
+                    filter(sentencenumber<= quantile(sentencenumber,0.1*i))%>%
+                    filter(sentencenumber> quantile(sentencenumber,0.1*(i-1))-1)%>%
+                    ungroup()%>%
+                    inner_join(get_sentiments("nrc"))%>%
+                    count(sentiment) %>%
+                    select(n) %>% t())
+ }
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
Joining, by = "word"
```

```
> colnames(df_most) <- c("anger", "anticipation",  "disgust", "fear", "joy", "negative",
"positive", "sadness", "surprise", "trust")
> df_most$sum<- rowSums(df_most)
> df_most$angerprop<- df_most$anger/df_most$sum
> df_most$anticipationprop<- df_most$anticipation/df_most$sum
> df_most$disgustprop<- df_most$disgust/df_most$sum
> df_most$fearprop<- df_most$fear/df_most$sum
> df_most$joyprop<- df_most$joy/df_most$sum
> df_most$negativeprop<- df_most$negative/df_most$sum
> df_most$positiveprop<- df_most$positive/df_most$sum
> df_most$sadnessprop<- df_most$sadness/df_most$sum
> df_most$surpriseprop<- df_most$surprise/df_most$sum
> df_most$trustprop<- df_most$trust/df_most$sum
> ggplot() +
+   geom_line(data = df_most, aes(x = seq(10, 100, by=10), y = angerprop, color =
"anger"),linetype="dashed") +
+   geom_line(data = df_most, aes(x = seq(10, 100, by=10), y = disgustprop, color =
"disgust"),linetype="dashed") +
+   geom_line(data = df_most, aes(x = seq(10, 100, by=10), y = anticipationprop, color =
"anticipation"), linetype="dashed") +
+   geom_line(data = df_most, aes(x = seq(10, 100, by=10), y = fearprop, color = "fear"),
linetype="dashed") +
+   geom_line(data = df_most, aes(x = seq(10, 100, by=10), y = joyprop, color = "joy"), color =
"pink") +
+   geom_line(data = df_most, aes(x = seq(10, 100, by=10), y = negativeprop, color =
"negative"),linetype="dashed") +
+   geom_line(data = df_most, aes(x = seq(10, 100, by=10), y = positiveprop, color = "positive"))
+
+   geom_line(data = df_most, aes(x = seq(10, 100, by=10), y = sadnessprop, color = "sadness"),
linetype="dashed") +
+   geom_line(data = df_most, aes(x = seq(10, 100, by=10), y = trustprop, color = "trust")) +
+   scale_color_manual(values = c(
+     'anger' = 'orange',
+     'disgust'='brown',
+     'anticipation'= 'yellow',
+     'fear'='gray',
+     'joy'='pink',
+     'negative'='red',
+     'positive'='green',
+     'sadness'='blue',
+     'trust'='magenta'
+     )) +
+   labs(color = 'Sentiment') +
+   xlab('Time Percentile') +
+   ylab('Sentiment') +
+   ggtitle("Most viewed: Change in sentiments as talks progress")
```

>


Most viewed: Change in sentiments as talks progress
Sentiment
0.25
0.20
0.15
0.10
0.05
25
50
75
100
Time Percentile
Sentiment
anger
anticipation
disgust
fear
negative
positive
sadness
trust