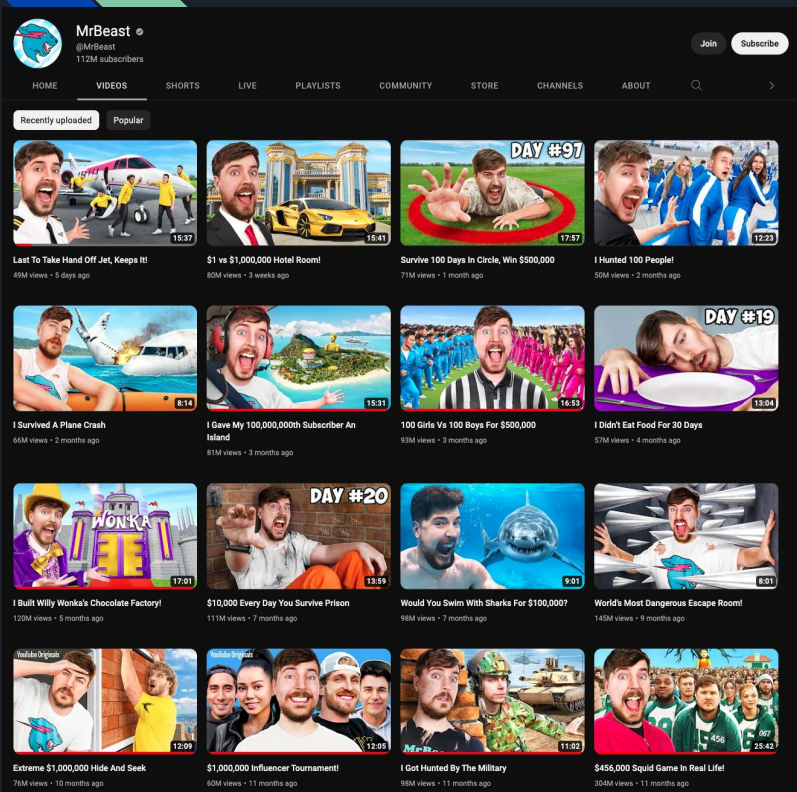


A decorative graphic on the left side of the slide consists of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

How Mr. Beast Grew on Youtube

Jessica Nguyen CDS 301

Who is Mr. Beast?



Being on the YouTube platforms for almost 10 years now, Mr Beast is considered one of the fastest-growing and most successful Youtube channels, hitting 112 million subscribers within his main and averaging 10 million subscribers on each of his four side channels. His content ranges from challenges, gaming, and philanthropy videos. His audiences' ages range between elementary to high school students.

I wanted to ask myself, “*how could I create a youtube channel as successful as Mr. Beast?*”

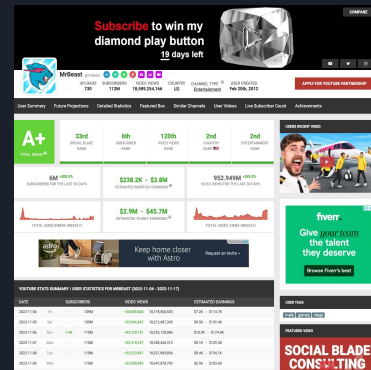
Surprisingly, there is already a website that could answer that question for me called *socialblade.com*. However, there are also limitations to this website.

- ❑ No qualitative data analysis (#4 Tufte's Rule: Integration)
- ❑ No story about interaction between creator vs. viewer being told with exception of “subscriber trend” graphs (#3 Tufte's Rule: Multivariate)
- ❑ No way to find dependent variables influencing each other (#1 Tufte's Rule: Comparisons #2 Tufte's Rule: Casualty)
- ❑ Too many ads

But there are also strengths.

- ✅ Trend line graphs for total videos and total subscribers (#6 Tufte's Rule: Context)
- ✅ Includes information about subscribers, video uploads, views, monthly earnings (#3 Tufte's Rule: Multivariate)

My Goal will be to **better optimize Mr. Beast's youtube channel API** than *socialblade.com* to provide more insightful statistical analysis.





Literature Review

Through my findings within Mr Beast's YouTube channel, I found that the correlation between Mr. Beast's most commonly used words within his YouTube titles is not a strong causative variable to explain the growth of his YouTube channel. This was supported by the lack of his most popular keywords being within his best viewed YouTube videos, best liked YouTube videos, and best commented YouTube videos, **with only 8 of his best performing videos keywords being contained within his 15 most commonly used words in Youtube titles.** This translates to a 53% probability that a commonly used Mr. Beast video title will become a best performing video. How Mr. Beast can better optimize his YouTube to continue growing is by integrating more words such as **circle, leave, buried, real, alive, spent** within his YouTube titles.

I've utilized all 5 of Tufte's Rules of Visualization (Comparisons, Causality, Multivariate, Integration, Documentation, Context) within this project. Through my exploratory data analysis, I was able to take qualitative data to better answer the question to whether Mr. Beast's growth of his channel is due to his choice in YouTube titles. Within the data, there are both qualitative data (video titles) and quantitative data (video likes, video comments, video views) with a stronger emphasis on the video titles as shown by the word clouds.

Dataset and Packages

library(dplyr)
library(readr)
library(devtools)
library(AnomalyDetection)
library(ggplot2)
library(tm)
library(stringi)
library(lattice)
library(udpipes)
library(wordcloud)
library(SnowballC)
library(highcharter)
library(tidyverse)

For the word cloud graph
For the scatter, line graphs
For reading external files

```
key <- ("A1zaSyC1ETodFrV5QzHh81TghzflQpxXa8Mmsuc") #api key
MrBeastChannelID <- "UCX6G03DkccsYNErH8uQuVA" #channel id
base <- "https://www.googleapis.com/youtube/v3/" #baseurl

# Construct the API call: https://www.yuichitsuka.com/youtube-data-extract-r/
api_params <-
  paste0("key=", key),
  paste0("id=", MrBeastChannelID),
  "part=snippet,contentDetails,statistics",
  sep = "&"
api_call <- paste0(base, "channels", "?", api_params)
api_result <- GET(api_call)
json_result <- httr::content(api_result, "text", encoding="UTF-8")

#format channel information from API into dataframe
channel_json <- fromJSON(json_result, flatten = T)
channel.df <- as.data.frame(channel.json)

#take 'upload' playlist ID from the channel dataframe in order to get video information
playlist_id <- channel.df$items.contentDetails.relatedPlaylists.uploads

#temporary variables
nextPageToken <- ""
upload.df <- NULL
pageInfo <- NULL

#go through pages to get all the video information
while (!is.null(nextPageToken)) {
  # Construct the API call
  api_params <-
    paste0("key=", key),
    paste0("playlistId=", playlist_id),
    "part=snippet,contentDetails",
    "maxResults=50",
    sep = "&"

  # Add the page token for page 2 onwards
  if (nextPageToken != "") {
    api_params <- paste0(api_params,
      "&pageToken=", nextPageToken)
  }

  api_call <- paste0(base, "playlists/items", "?", api_params)
  api_result <- GET(api_call)
  json_result <- httr::content(api_result, "text", encoding="UTF-8")
  upload_json <- fromJSON(json_result, flatten = T)

  nextPageToken <- upload_json$nextPageToken
  pageInfo <- upload_json$pageInfo

  curr.df <- as.data.frame(upload_json$items)
  if (is.null(upload.df)) {
    upload.df <- curr.df
  } else {
    upload.df <- bind_rows(upload.df, curr.df)
  }
}

#upload.df is the dataframe that contains all the video information (titles, description, when it was posted)
```

#Now, I need to get the video statistics such as the likes, view count, and comment count.

```
#create empty vectors to be the column names
items.id <- c()
items.statistics.viewCount <- c()
items.statistics.likeCount <- c()
items.statistics.commentCount <- c()

for (x in 1:730) { #loop through all 730 Mr Beast videos
  vid_api_params <-
    paste0("key=", key),
    paste0("id=", upload.df$contentDetails.videoId[x]),
    "part=statistics",
    sep = "&"
  vid_api_call <- paste0(base, "videos", "?", vid_api_params)
  vid_api_result <- GET(vid_api_call)
  vid_json_result <- httr::content(vid_api_result, "text", encoding="UTF-8")
  vid_json <- fromJSON(vid_json_result, flatten = T)
  vid.df <- as.data.frame(vid_json)

  #If there are any variables within the row that are blank, say they are NA

  if(!("items.statistics.likeCount" %in% colnames(vid.df)) == FALSE){
    items.statistics.likeCount[x] <- NA
  }
  else if("items.id" %in% colnames(vid.df) == FALSE)
  {
    items.id[x] <- NA
  }
  else if("items.statistics.viewCount" %in% colnames(vid.df) == FALSE){
    items.statistics.viewCount[x] <- NA
  }
  else if ("items.statistics.commentCount" %in% colnames(vid.df) == FALSE){
    items.statistics.commentCount[x] <- NA
  }
  else{
    items.id[x] <- vid.df$items.id[1] #if all the variables within the row is present, then add that variable into the vector
    items.statistics.commentCount[x] <- vid.df$items.statistics.commentCount[1]
    items.statistics.viewCount[x] <- vid.df$items.statistics.viewCount[1]
    items.statistics.likeCount[x] <- vid.df$items.statistics.likeCount[1]
  }
}




vid.df <- cbind(items.id, items.statistics.viewCount, items.statistics.likeCount, items.statistics.commentCount) #bind the vector to get a matrix
vid.df[is.na(vid.df)] <- 0 #make all NA into 0

upload.df <- cbind(upload.df, vid.df) #combine the video information (titles, description, when it was posted) and video statistics (likes, view count, and comment count)

write.csv(upload.df, file = "upload.csv")
```

The Completed, Created Dataset

8 columns and 730 rows

items.id	items.statistics.viewCount	items.statistics.likeCount	items.statistics.commentCount	snippet.publishedAt	snippet.description	snippet.title	snippet.playlistId	
1	kX3nB4PpJko	50139135	2370866	77669	2022-11-12T21:00:00Z	Close this and watch to the end to see who wins! New Merch - https://shopmrbeast.com/ At the end I said we would offset all carbon emissions but what we actually did was offset it by 10x! SUBSCRIBE OR I TAKE YOUR DOG  ----- follow all of these or i will kick you • Facebook - https://www.facebook.com/MrBeast6000/ • Twitter - https://twitter.com/MrBeast • Instagram - https://www.instagram.com/mrbeast • Im Hiring! - https://www.mrbeastjobs.com/ -----	Last To Take Hand Off Jet, Keeps It!	UUX6OQ3DkcsbYNE6H8uQQUwA
2	YivcFJOE-OE	93869725	5409195	33870	2022-11-02T21:00:01Z	New Merch - https://shopmrbeast.com/ SUBSCRIBE OR I TAKE YOUR DOG  ----- follow all of these or i will kick you • Facebook - https://www.facebook.com/MrBeast6000/ • Twitter - https://twitter.com/MrBeast • Instagram - https://www.instagram.com/mrbeast • Im Hiring! - https://www.mrbeastjobs.com/ -----	Giving iPhones Instead Of Candy on Halloween	UUX6OQ3DkcsbYNE6H8uQQUwA
3	logcY_4xQJo	80845611	3566546	103190	2022-10-22T21:00:00Z	The hotel at the end is worth the wait! Download the Experian App: https://smart.link/n3op1gefxtzjn or Visit Experian.com/Boost. *Results will vary. Not all payments are boost-eligible or considered by lenders. Credit Card offers are not available Check out Conrad Maldives Rangali Island! Facebook @conradmaldivesarangaliland Youtube https://www.youtube.com/c/ConradMaldivesRangaliIsland New Merch - https://shopmrbeast.com/ SUBSCRIBE OR I TAKE YOUR DOG  ----- follow all of these or i will kick you • Facebook - https://www.facebook.com/MrBeast6000/ • Twitter - https://twitter.com/MrBeast • Instagram - https://www.instagram.com/mrbeast • Im Hiring! - https://www.mrbeastjobs.com/ -----	\$1 vs \$1,000,000 Hotel Room!	UUX6OQ3DkcsbYNE6H8uQQUwA
4	S_CUEOBZ0P4	79919105	7158110	34282	2022-10-21T20:00:06Z	Winning totally Rocks!		

Word Cloud of Commonly Used Words in Video Titles

```
#creating wordcloud for mr. beast's titles with numbers
titles_n <- read_file('/Users/jessicanguyen/mrbeast/MrBeastYTAnalytics/mrbeast_titles.txt')

tdocs<-Corpus(VectorSource(titles_n))

## pre-process the texts
tdocs <- tm_map(tdocs, function(x) stri_replace_all_regex(x, "<.+?>", " "))
tdocs <- tm_map(tdocs, function(x) stri_replace_all_fixed(x, "\\n", " "))

# consider all raw text
tdocs <- tm_map(tdocs, PlainTextDocument)
# remove punctuation
tdocs <- tm_map(tdocs, removePunctuation)
# set capital letters to lower letters
tdocs <- tm_map(tdocs, content_transformer(tolower))
# remove common words ("the", "and")
tdocs <- tm_map(tdocs, removeWords, stopwords("english"))

tdtm<-DocumentTermMatrix(tdocs)
tdtm<-TermDocumentMatrix(tdocs)

tmywords<-tdtm$dimnames$Terms

tdtm$dimnames$Docs<-as.character(c(1:length(text)))
rowTotals<- apply(tdtm, 1, sum)
tdtm.new <- tdtm[rowTotals> 0,]

tdictionary<-tdtm$dimnames$Terms

tfreq <- colSums(as.matrix(tdtm))
tord <- order(tfreq,decreasing=TRUE)
t_dict <- as.data.frame(tfreq[tord])

tm <- as.matrix(tdtm)
tv <- sort(colSums(tm), decreasing=TRUE)
t_myNames <- names(tv)
t_dtmnew <- data.frame(word=t_myNames, freq=tv)
layout(matrix(c(1, 2), nrow=1), heights=c(1, 4))
par(mar=rep(0, 4))
plot.new()
text(x=0.5, y=0.5, "Most Common Words and Numbers Used in Mr. Beast's Video Titles")
wordcloud(t_dtmnew$word, family = "serif", font = 6, colors=colorRampPalette(brewer.pal(9,"Greens"))(32)[seq(8,32,6)], t_dtmnew$freq, min.freq = 20)
```

Most Common Words and Numbers Used in Mr. Beast's Video Titles

youtube
make
100000
video
10000
battle
youtubers
people
challenge
money
minecraft
last
much
hours
2016

Comparison of Graphs: Best Viewed Videos vs. Best Liked Videos vs. Best Liked Videos

```
#best videos via views
upload <- upload[order(upload$items.statistics.viewCount, decreasing = TRUE),] #making sure it is in highest to lowest order
best_viddf_views <- data.frame(upload$snippet.title[1:10], upload$items.statistics.viewCount[1:10]) #capturing only top ten videos

plot2 <- ggplot(data = best_viddf_views, mapping = aes(x = reorder(as.factor(upload$snippet.title.1.10.), -upload$items.statistics.viewCount.1.10.), y = upload$items.statistics.viewCount.1.10.))
geom_bar(stat = 'identity', fill = 'lightblue') +
  labs(x = "YouTube Titles", y = "View Count") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

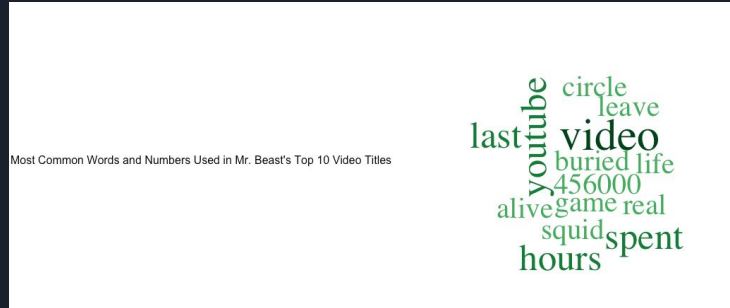
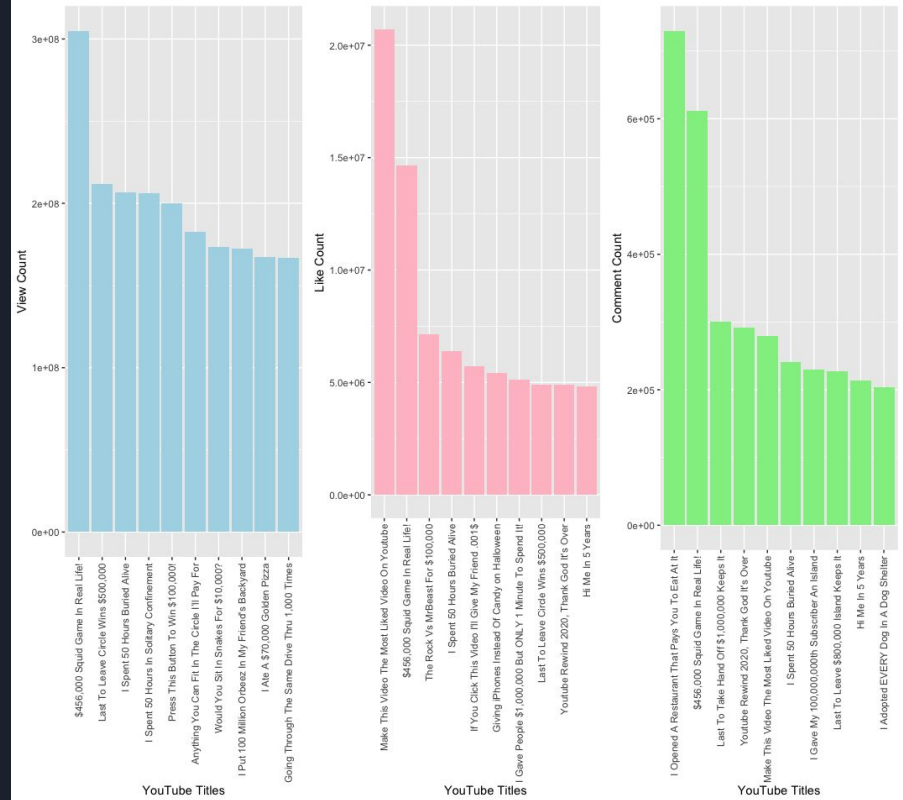
#best videos via likes
upload <- upload[order(upload$items.statistics.likeCount, decreasing = TRUE),] #making sure it is in highest to lowest order
best_viddf_likes <- data.frame(upload$snippet.title[1:10], upload$items.statistics.likeCount[1:10]) #capturing only top ten videos

plot3 <- ggplot(data = best_viddf_likes, mapping = aes(x = reorder(as.factor(upload$snippet.title.1.10.), -upload$items.statistics.likeCount.1.10.), y = upload$items.statistics.likeCount.1.10.))
geom_bar(stat = 'identity', fill = 'pink') +
  labs(x = "YouTube Titles", y = "Like Count") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

#best videos via comments
upload <- upload[order(upload$items.statistics.commentCount, decreasing = TRUE),] #making sure it is in highest to lowest order
best_viddf_comments <- data.frame(upload$snippet.title[1:10], upload$items.statistics.commentCount[1:10]) #capturing only top ten videos

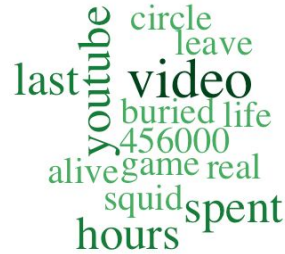
plot1 <- ggplot(data = best_viddf_comments, mapping = aes(x = reorder(as.factor(upload$snippet.title.1.10.), -upload$items.statistics.commentCount.1.10.), y = upload$items.statistics.commentCount.1.10.))
geom_bar(stat = 'identity', fill = 'lightgreen') +
  labs(x = "YouTube Titles", y = "Comment Count") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

plot_grid(plot2, plot3, plot1, ncol = 3, nrow = 1)
```



Comparison of Mr Beast's Common Titles and Best Performing Titles

Most Common Words and Numbers Used in Mr. Beast's Top 10 Video Titles



Words included in the best performing videos that are not included within the most commonly used videos: circle, leave, buried, squid (Squid Games), real, alive, spent (8 words in total out of 14)

Most Common Words and Numbers Used in Mr. Beast's Video Titles



8/15 words here that match the words within the 'Most common words and numbers used in Mr. Beast's Top 10 Video Titles'



Citations

Code for extracting YouTube API into R-studio: <https://www.yuichiotsuka.com/youtube-data-extract-r/>

Inspiration for analyzing YouTube analytics:

https://www.youtube.com/watch?v=D56_Cx36oGY&t=500s