Statistical Analysis Report for Capstone Project

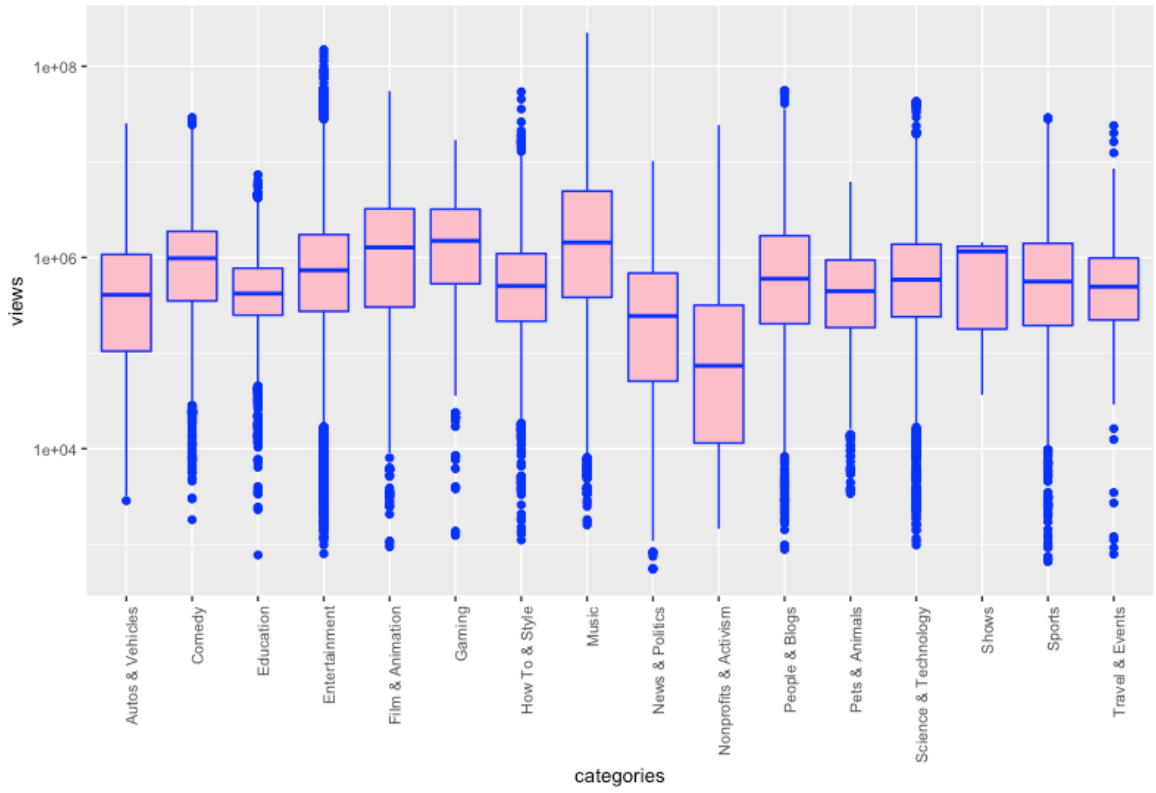
After cleaning and wrangling the original data set, there was a total of 40,949 observations. Because potential clients would be interested in which videos were the most popular, according to total view count, the top 100 viewed videos were isolated so that the tags used in these videos could be explored. This insight will help build a model that can predict the number of views a YouTube video will receive based on the exact tags used.

Therefore, to make the data set more manageable, the 100 most viewed videos from the entire list of 40K+ observations were counted using the following code:

```
USvideos_top100_most_viewed = USvideos_new %>% select(title, tags, views, likes,
comment_count, category_id, categories, publish_date_official, trend_date_official) %>%
  distinct(title, tags, .keep_all = TRUE) %>%
  mutate(Title = title) %>%
  mutate(Tags = tags) %>%
  arrange(desc(views)) %>% head(100)
```

Preliminary analysis has shown that the majority of the top viewed videos fall into mostly two categories – *Music* and *Entertainment.*

	title	tags	views	likes	comment_count	category_id	categories
1	Childish Gambino - This Is America (Official Video)	Childish Gambino|"Rap"|"This Is America"|"mcDJ Reco...	225211923	5023450	517232	10	Music
2	YouTube Rewind: The Shape of 2017 | #YouTubeRewi...	Rewind|"Rewind 2017"|"youtube rewind 2017"|"#You...	149376127	3093544	810698	24	Entertainment
3	Ariana Grande - No Tears Left To Cry	Ariana|"Grande"|"No"|"Tears"|"Left"|"To"|"Cry"|"Univer...	148689896	3094021	242039	10	Music
4	Becky G, Natti Natasha - Sin Pijama (Official Video)	Becky G|"Natti Natasha"|"Natti Natasha Music"|"Natti ...	139334502	1425496	83941	10	Music
5	BTS (방탄소년단) 'FAKE LOVE' Official MV	BIGHIT|"빅히트"|"방탄소년단"|"BTS"|"BANGTAN"|"방탄"|"FA...	123010920	5613827	1228655	10	Music
6	The Weeknd - Call Out My Name (Official Video)	The|"Weeknd"|"Call"|"Out"|"My"|"Name"	122544931	1427436	55320	10	Music
7	Luis Fonsi, Demi Lovato - Échame La Culpa	Luis|"Fonsi"|"Demi"|"Lovato"|"Échame"|"La"|"Culpa"|"U...	102012605	2376636	134224	10	Music
8	Cardi B, Bad Bunny & J Balvin - I Like It [Official Music ...	Cardi B|"I Like It"|"Invasion of Privacy"|"Bad Bunny"|"J ...	94254507	1816753	101077	10	Music
9	Marvel Studios' Avengers: Infinity War Official Trailer	marvel|"comics"|"comic books"|"nerdy"|"geeky"|"supe...	91933007	2625661	350458	24	Entertainment
10	Maluma - El Préstamo (Official Video)	Maluma Music|"Maluma Official Video"|"Maluma Video...	87264467	815369	35945	10	Music
11	BTS (방탄소년단) 'FAKE LOVE' Official MV	BIGHIT|"빅히트"|"방탄소년단"|"BTS"|"BANGTAN"|"방탄"|"FA...	73463137	4924045	1084421	10	Music
12	Taylor Swift - Delicate	Taylor Swift|"Delicate"|"Big"|"Machine"|"Records"|"LLC...	71560694	1928392	162990	10	Music
13	Calvin Harris, Dua Lipa - One Kiss (Official Video)	calvin harris|"calvin harris one kiss"|"calvin harris dua...	71017021	828626	26279	10	Music
14	TWICE What is Love? M/V	TWICE What is Love|"TWICE What is Love?"|"TWICE 왓이...	69295519	1324609	238744	10	Music
15	Maroon 5 - Girls Like You ft. Cardi B	Maroon|"Girls"|"Like"|"You"|"Interscope"|"Records*"|"P...	66529577	2488565	142410	10	Music
16	Drake - Nice For What	Drake|"Nice"|"For"|"What"|"Young"|"Money"|"Hip"|"Hop"	60635812	994986	55653	10	Music
17	Maluma - Marinero (Official Video)	Maluma Marinero|"Maluma Marinero official video"|"M...	59877217	853193	48391	10	Music
18	VENOM - Official Trailer (HD)	Venom|"Venom Movie"|"Venom (2018)"|"Marvel"|"Mar...	59254638	1295189	139879	24	Entertainment
19	Bruno Mars - Finesse (Remix) [Feat. Cardi B] [Official ...	Bruno Mars|"Finesse"|"Cardi B"|"Finesse Remix"|"Brun...	57951412	1919583	133601	10	Music
20	To Our Daughter	Kylie Jenner|"Kylie"|"Travis Scott"|"Baby"|"Annoucement"	56111957	0	0	22	People & Blogs
21	Ed Sheeran - Happier (Official Video)	edsheeran|"ed sheeran"|"acoustic"|"live"|"cover"|"offici...	55054077	1378923	67092	10	Music
22	Selena Gomez - Back To You (Lyric Video)	selena gomez|"13 reasons why"|"back to you"|"selena...	54863912	922355	41774	1	Film & Animation
23	42 HOLY GRAIL HACKS THAT WILL SAVE YOU A FORT...	5-Minute Crafts|"DIY"|"Do it yourself"|"crafts"|"trucos"...	54155921	378111	24679	26	How To & Style
24	Marvel Studios' Avengers: Infinity War - Official Trailer	marvel|"comics"|"comic books"|"nerd"|"geek"|"superh...	52404970	1565579	194290	24	Entertainment
25	Ed Sheeran - Happier (Official Video)	edsheeran|"ed sheeran"|"acoustic"|"live"|"cover"|"offici...	49353979	1346647	66081	10	Music
26	Sanju | Official Teaser | Ranbir Kapoor | Rajkumar Hir...	Sanju Teaser|"Official Teaser"|"Sanju Official Teaser"|"...	48654951	811144	48941	24	Entertainment



As illustrated in the boxplot above, certain categories received a higher amount of views across the entire data set. Because this provides limited value to clients whose videos are in categories outside of *Music* and *Entertainment* – for example, videos in categories such as *News & Politics* or *Science & Technology* – the data is restructured to show the top 100 viewed videos from within each of the 32 categories (from category 1 to category 44). See the code used below.

```
USvideos_top100_most_viewed_category1 = USvideos_new %>% select(title, tags, views, likes,
comment_count, category_id, categories, publish_date_official, trend_date_official) %>%
  distinct(title, tags, .keep_all = TRUE) %>%
  mutate(Title = title) %>%
  mutate(Tags = tags) %>%
  arrange(desc(views)) %>%
  filter(category_id == 1) %>% head(100)

…
USvideos_top100_most_viewed_category44 = USvideos_new %>% select(title, tags, views,
likes, comment_count, category_id, categories, publish_date_official, trend_date_official) %>%
  distinct(title, tags, .keep_all = TRUE) %>%
  mutate(Title = title) %>%
  mutate(Tags = tags) %>%
  arrange(desc(views)) %>%
  filter(category_id == 44) %>% head(100)
```

When the data was wrangled according to category id, it was discovered that some categories did not have enough observations to contain 100 videos. Thus, those categories will be excluded from further analysis. The 12 categories that feature 100 videos or more, and will be used to create the predictive model, are: *Film & Animation, Music, Pets & Animals, Sports, Gaming, People & Blogs, Comedy, Entertainment, News & Politics, How To & Style, Education*, and *Science & Technology*.

After there was a data set created for each of the 12 categories, it was now time to focus on the "tags" variable. Because clients need to know which tags to use to drive the most views to their videos, the frequency with which each tag appeared needed to be calculated. While wrangling the original data set, it was clear that the "tags" variable would be messy to work with. Some tags were misspelled, others were in a language other than English, and some contained characters or punctuation issues.

To count the frequency with which each tag appears in each category data set, each individual tag was separated into its own column while keeping it in its corresponding row to match the video the tag belonged to. Cells containing no tags were automatically filled in with NA. Here is an example showing how the tags were separated in the "Music" category (category_id = 10).

```
Tags_Separated_10 <- USvideos_top100_most_viewed_category10 %>% separate(Tags,
c("tag1", "tag2", "tag3", "tag4", "tag5", "tag6", "tag7", "tag8", "tag9", "tag10", "tag11", "tag12",
"tag13", "tag14", "tag15", "tag16", "tag17", "tag18", "tag19", "tag20", "tag21", "tag22",
"tag23", "tag24", "tag25", "tag26", "tag27", "tag28", "tag29", "tag30", "tag31", "tag32",
"tag33", "tag34", "tag35", "tag36", "tag37", "tag38", "tag39", "tag40", "tag41", "tag42",
"tag43", "tag44", "tag45", "tag46", "tag47", "tag48", "tag49", "tag50", "tag51", "tag52"), "\\|",
convert = TRUE)
```

To summarize all of the tag frequencies into a single table, regular expressions were defined. Here is an example for the tags counted in the "Music" category (category_id = 10).

```
tags_10 <- Tags_Separated_10
tag_cols <- which(regexpr("tag[0123456789]",
                          colnames(tags_10)) > 0)
all_tags_10 <- character()
for (i in 1:length(tag_cols)) {
  all_tags_10 <- c(all_tags_10, tags_10[tag_cols[i]])
}
all_tags_10 <-
  unlist(all_tags_10) %>%
  as.data.frame() %>%
  rename(tag_names = names(.)[1]) %>%
  arrange(tag_names)
```

	all_tags	Freq
1	Pop	23
2	live	7
3	official	7
4	BANGTAN	6
5	BIGHIT	6
6	BTS	6
7	official video	6
8	Rap	6
9	방탄	6
10	방탄소년단	6
11	빅히트	6
12	acoustic	5
13	cover	5
14	ed sheeran	5
15	edsheeran	5
16	lyrics	5
17	Records	5
18	remix	5
19	session	5
20	[none]	4
21	Animation	4
22	Cardi B	4

```
tag_freq_10 <-
  table(all_tags_10) %>%
  as.data.frame() %>%
  arrange(desc(Freq)) %>%
  mutate(all_tags_10 = as.character(all_tags_10))
```

To make the tag frequencies into usable data by combining like terms, correcting spelling errors, and adjusting spacing, the following code was run. It should be noted that Google Translate software was used to translate the Korean characters. After comparing the two tables, it is clear that when the tags were consolidated, some tags increased in frequency count. (See the “pop” tag as an example.)

	tag	count
1	Hannah Stocking	4
2	pop	28
3	live	15
4	official	45
5	rap	9
6	acoustic	5
7	cover	5
8	Ed Sheeran	22
9	lyric	17
10	BTS	42

```
tag_freq_summary_10 <- data.frame("tag" = character(), "count" =
integer(), stringsAsFactors = FALSE)

# "Hannah Stocking"
tag_freq_summary_10[1, "tag"] <- as.character("Hannah Stocking")
tag_freq_summary_10[1, "count"] <- sum(tag_freq_10[which(grepl("Hannah Stocking",
tag_freq_10$all_tags, ignore.case = TRUE) | grepl("HannahStocking", tag_freq_10$all_tags,
ignore.case = TRUE)), "Freq"])

# "pop"
tag_freq_summary_10[2, "tag"] <- as.character("pop")
tag_freq_summary_10[2, "count"] <- sum(tag_freq_10[which(grepl("Pop",
tag_freq_10$all_tags, ignore.case = TRUE) & !grepl("popp", tag_freq_10$all_tags, ignore.case =
TRUE)), "Freq"])

# "live"
tag_freq_summary_10[3, "tag"] <- as.character("live")
tag_freq_summary_10[3, "count"] <- sum(tag_freq_10[which(grepl("live",
tag_freq_10$all_tags, ignore.case = TRUE) & !grepl("we can't live", tag_freq_10$all_tags,
ignore.case = TRUE)), "Freq"])

# "official"
tag_freq_summary_10[4, "tag"] <- as.character("official")
tag_freq_summary_10[4, "count"] <- sum(tag_freq_10[which(grepl("official",
tag_freq_10$all_tags, ignore.case = TRUE) | grepl("官方", tag_freq_10$all_tags, ignore.case =
TRUE)), "Freq"])

# "rap"
tag_freq_summary_10[5, "tag"] <- as.character("rap")
tag_freq_summary_10[5, "count"] <- sum(tag_freq_10[which((grepl("Rap",
tag_freq_10$all_tags, ignore.case = TRUE) | grepl("Hip", tag_freq_10$all_tags, ignore.case =
```

```
TRUE)) & !grepl("bubble wrap", tag_freq_10$all_tags, ignore.case = TRUE) & !grepl("Rapt",
tag_freq_10$all_tags, ignore.case = TRUE) & !grepl("Shakira ft. Maluma Trap",
tag_freq_10$all_tags, ignore.case = TRUE) & !grepl("Trap", tag_freq_10$all_tags, ignore.case =
TRUE) & !grepl("realtionship", tag_freq_10$all_tags, ignore.case = TRUE)), "Freq"])

# "acoustic"
tag_freq_summary_10[6, "tag"] <- as.character("acoustic")
tag_freq_summary_10[6, "count"] <- sum(tag_freq_10[which(grepl("acoustic",
tag_freq_10$all_tags, ignore.case = TRUE)), "Freq"])

# "cover"
tag_freq_summary_10[7, "tag"] <- as.character("cover")
tag_freq_summary_10[7, "count"] <- sum(tag_freq_10[which(grepl("cover",
tag_freq_10$all_tags, ignore.case = TRUE)), "Freq"])

# "Ed Sheeran"
tag_freq_summary_10[8, "tag"] <- as.character("Ed Sheeran")
tag_freq_summary_10[8, "count"] <- sum(tag_freq_10[which(grepl("ed sheeran",
tag_freq_10$all_tags, ignore.case = TRUE) | grepl("Ed Skrein", tag_freq_10$all_tags,
ignore.case = TRUE) | grepl("edsheeran", tag_freq_10$all_tags, ignore.case = TRUE)), "Freq"])

# "lyric"
tag_freq_summary_10[9, "tag"] <- as.character("lyric")
tag_freq_summary_10[9, "count"] <- sum(tag_freq_10[which(grepl("lyric",
tag_freq_10$all_tags, ignore.case = TRUE)), "Freq"])

# "BTS"
tag_freq_summary_10[10, "tag"] <- as.character("BTS")
tag_freq_summary_10[10, "count"] <- sum(tag_freq_10[which(grepl("BANGTAN",
tag_freq_10$all_tags, ignore.case = TRUE) | grepl("BIGHIT", tag_freq_10$all_tags, ignore.case =
TRUE) | grepl("BTS", tag_freq_10$all_tags, ignore.case = TRUE) | grepl("방탄",
tag_freq_10$all_tags, ignore.case = TRUE) | grepl("방탄소년단", tag_freq_10$all_tags,
ignore.case = TRUE) | grepl("빅히트", tag_freq_10$all_tags, ignore.case = TRUE) | grepl("FAKE
LOVE", tag_freq_10$all_tags, ignore.case = TRUE) | grepl("FAKE_LOVE", tag_freq_10$all_tags,
ignore.case = TRUE)), "Freq"])
```

Once the “tags” variable was cleaned up, the original tags in the tag columns were replaced with the “tag_freq_summary” chosen term. All other, non-used tags are labeled as NA. This process is used for each of the 12 data sets representing the different video categories.

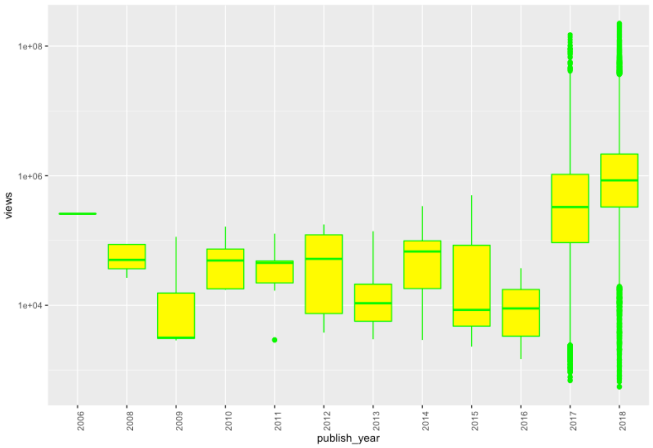
Further statistical analysis revealed that the range for the publish year differed from that of the trending year. As can be seen below, the trending year only covers 2017 and 2018 while the publish year accounts for 2006 and 2008 – 2018. This is important because there would be no data for the trending year before 2017 so a comparison of these two time points cannot be

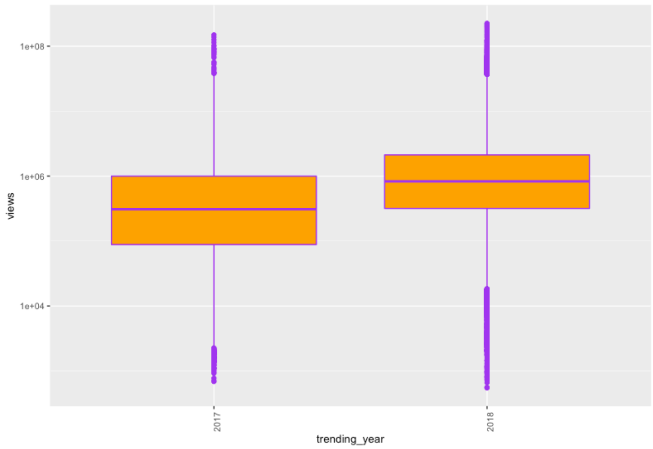
made beyond these two years. Likewise, to include the single observation in 2006 for the publish year would skew the data given that over 30K observations were made in 2018. See the boxplots below to compare the number of video views by the year they were first published to the number of video views by the year they trended.

```
> table(USvideos_new$publish_year)

 2006  2008  2009  2010  2011  2012  2013  2014  2015  2016  2017  2018
    1    11    14    19    27    24    44    32    35    35 10428 30279
> table(USvideos_new$trending_year)

 2017  2018
 9600 31349
```





To account for this, only videos that were published in 2017 and 2018 were included in the final analysis. The time-series plot below compares the official publish date to the official trending date for the top 100 most viewed videos.