유튜브 인기 영상 데이터 특성 분석

건축학과 공학전공

2015-16506 이효배

1. 서론

1.1. Youtube 시대의 도래

코로나가 장기화되면서 언택트는 이제 선택이 아닌 필수가 되었다. 많은 사람들이 직접 만나는 것보다 집에서 핸드폰, 컴퓨터를 통해 여가를 즐기거나 사람들과 소통하는 일에 익숙해지고 있다. 온라인 회의, 온라인 강의, 온라인 공연은 SF영화에 등장하던 상상 속의 일이 아니라 이미 우리에게 스며든 일상이 되었다. 따라서 미디어의 중요성은 나날히 커지고 있고, 신문, TV를 통해 일방적으로 정보를 수용하던 사람들은 이제, 다양한 플랫폼을 통해 양방향으로 소통이 가능한 문화를 주도하고 있다. 이런 상황에서 Youtube 는 단순한 여가를 넘어 트렌드를 실시간으로 반영하고, 세계 각국의 사람들을 하나로 연결시켜주는 광장 역할을 한다. 전통적인 미디어가 할 수 없던 즉각적이고 상호적인 소통을 가능하게 하는 Youtube를 분석하는 것이 언택트 시대의 미디어 분석의 출발점이다.

본 레포트에서는 Youtube 에 등재되는 실시간 인기 급상승 영상(i.e. 유튜브 인기 영상)을 분석 대상으로 삼았다. 유튜브 인기 영상은 단 시간에 높은 조회수를 기록하는 영상으로, Youtube에서 따로 항목을 만들어 관리되고 있다. 국가별로 다르게 등록되는 유튜브 인기 영상은, 해당 국가 사람들이 주요 관심 대상이 무엇인지 파악할 수 있으며 이를 통해 문화적, 사회적 트렌드 또한 분석할 수 있다.

1.2. 데이터 소개

데이터사이언스 플랫폼인 Kaggle에서 데이터를 구했다. 유튜브 인기 영상에 관한 자료 중 가장 최근에 갱신되고 풍부한 자료를 담고 있는 Mitchell J의 자료를 사용했다.[1] 데이터에는 캐나다, 독일, 프랑스, 영국, 인도, 일본, 대한민국, 멕시코, 러시아, 미국의 자료들이 담겨있다. 각국의 2017년 11월부터 18년 6월까지 매일 인기 영상 200개가 있으며, 각 영상의 제목, 카테고리, 업로드 날짜, 태그, 조회수, 좋아요수, 싫어요수, 댓글수, 댓글/평가 활성화 여부, 영상 오류 및 삭제 여부 등의 정보가 있다. 분석에 사용할 feature들은 다음과 같다.

a. trending_date
   영상이 유튜브 인기 영상 목록에 등재된 날짜를 의미한다. 트렌드는 매우 빠르게 변하기 때문에, 인기 영상 목록은 날짜에 민감하게 반응한다.

b. category_id
   Youtube에서 정한 Category들 중 해당 영상이 어디에 속하는지를 나타내는 지표이다. R 분석에서 factor로 설정했다. 해당 영상의 성격을 파악할 수 있는 feature이기에 매우 유용한 정보로 판단했고 분석에 주로 활용했다. 국가별로 인기 동영상 중 다수를 차지하는 category들이 달랐는데, 그 이유를 분석하고, 시기별로, 인기 있는 Category가 무엇인지 파악했다.

[1] Mitchel J., 「 Trending Youtube Video Statistics 」, 『Kaggle』, 2018 (https://www.kaggle.com/datasnaek/youtube-new , 2020.11.30)

c. publish_time

해당 영상이 업로드 된 날짜와 시간 정보를 포함한다. 데이터는 UTC 시간대를 사용했기에, 국가별로 시간대를 맞춰서 분석을 진행했으며, 업로드 시간을 크게 3종류(Sleep Time, Work Time, Off Time)로 구분해서 새 범주를 만들었다. 한국, 프랑스 일본처럼 한 종류의 시간대를 쓰는 국가의 경우, 채널과 영상의 종류에 따른 업로드 전략을 파악하는 데에 유효한 feature였다. 이 역시 국가별로 매우 다른 양상을 보였다.

d. tag

주로 영상 설명란에 검색을 가능하게 하는 Hashtag 정보들이 담겨있다. category에 담을 수 없는 영상의 특징들이 tag에 담겨있다고 가정했다. 음악 관련 영상의 경우, 음악의 장르나 가수의 이름 등이 담겨 있고, 뉴스나 정치 관련 영상의 경우에는 특정인의 이름이나 특정 사건의 키워드 등이 담겨 있다. 일반적인 분석을 진행하기에는 실존인물의 이름이나, 회사 이름 등 너무 지엽적인 정보들이 다수 담겨 있어, Word Cloud를 통해 자주 등장하는 단어를 시각화하는 데까지만 진행했다.

e. views / likes / dislikes / comment_count

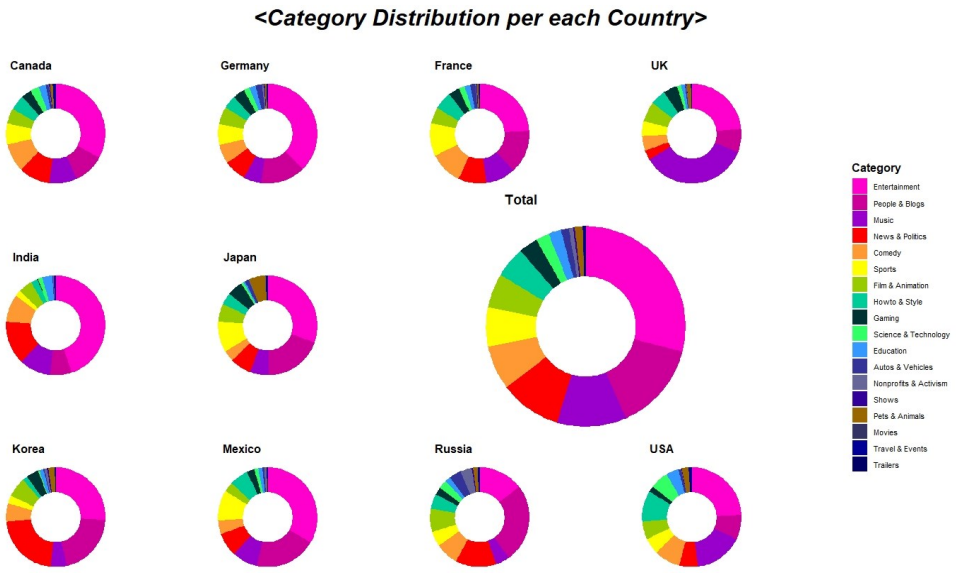
각각 영상 조회수/좋아요수/싫어요수/댓글수를 나타내는 수치정보이다. 인기 동영상인만큼 모두 높은 수의 조회수와 좋아요수를 받았다. 하지만 어떤 성격의 영상이냐에 따라 그 분포에 있어서 큰 차이를 보였다. 댓글수 또한 영상의 성격에 따라 큰 차이를 보였다. 각 category별로 해당 수치 정보들의 상관관계를 파악했다.

f. comment_disabled / ratings_disabled

댓글 또는 좋아요/싫어요를 비활성화하는 영상들이 소수 있었다. 사회적으로 논란이 되는 소재나, 아이가 출연하는 영상 등이 그런 경우가 있는데, 어떤 성격의 영상들이 그런지, Category별로 해당 설정을 비활성화하는 인기영상들의 비율이 어떻게 되는지를 분석했다.

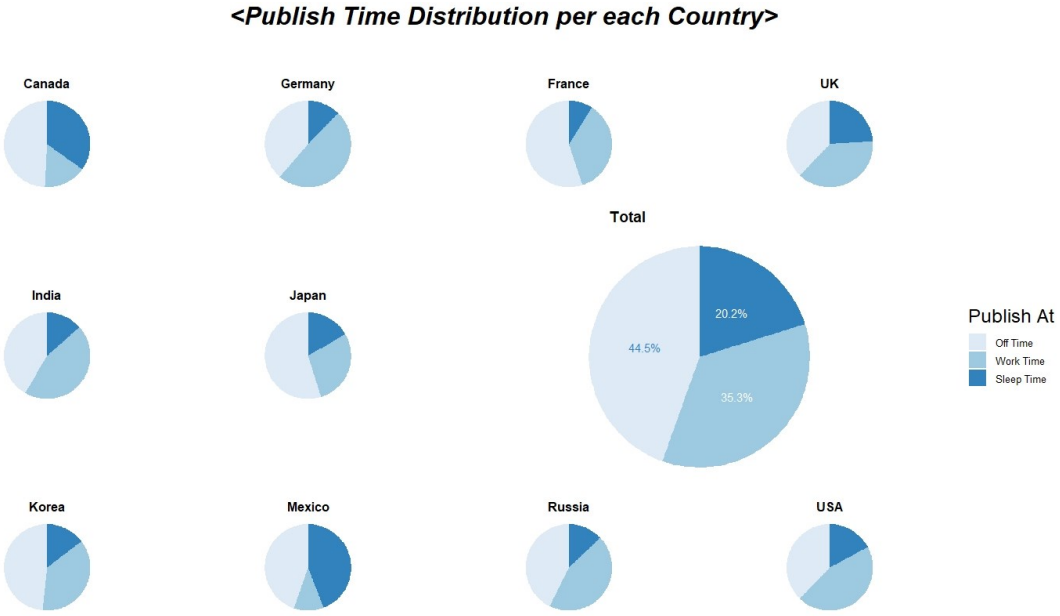
2. 국가별 데이터 분포

2.1. 국가별 Category 분포



국가별로 인기동영상에서의 Category 분포가 매우 다양하게 나타났다. 전체적으로는 Entertainment 가 가장 많고, Music, People & Blogs 가 그 뒤를 따랐다. 영국에서는 Music 이 압도적으로 높은 빈도를 차지했고, 인도에서는 Entertainment 가 거의 절반에 가까운 빈도를 차지했다. 한국에서는 News & Politics 가, 일본에서는 Pets & Animals 가 다른 나라에 비해 빈도를 차지하고 있었다. 해당 국가에서 사람들이 어느 소재에 관심을 갖는지가 반영된 현상이라고 생각한다. 한창 정치적인 사건들이 다수 발생하던 당시의 한국에서는 자연스레 News & Politics 영상들이 사람들에게 많이 공유되었음을 유추할 수 있다. 다른 나라들의 분포 역시 각각의 문화와 사회적 상황이 반영된 결과일 것이다.

2.2. 국가별 Publish Time

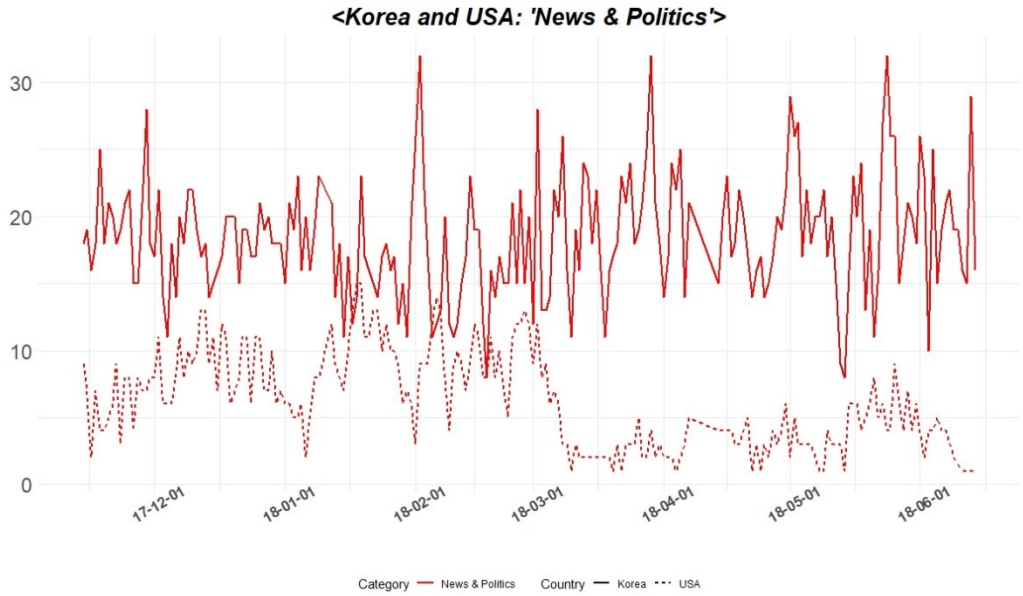


국가별로 인기동영상이 업로드 되는 시간을 분석했다. 주목할 점은 사람들이 쉬는 시간에 더 많은 영상이 업로드 된다는 점이었다. 이는 더 높은 조회수를 통해 수익을 창출하려는 유튜버들의 목적을 고려할 때 당연한 결과이다. 국가별로도 차이가 있었는데, 프랑스·일본에서는 영상 절반 이상이 쉬는 시간에 업로드 되는 반면, 영국 독일에서는 일하는 시간과 잠자는 시간에도 다수의 영상들이 업로드되었다.

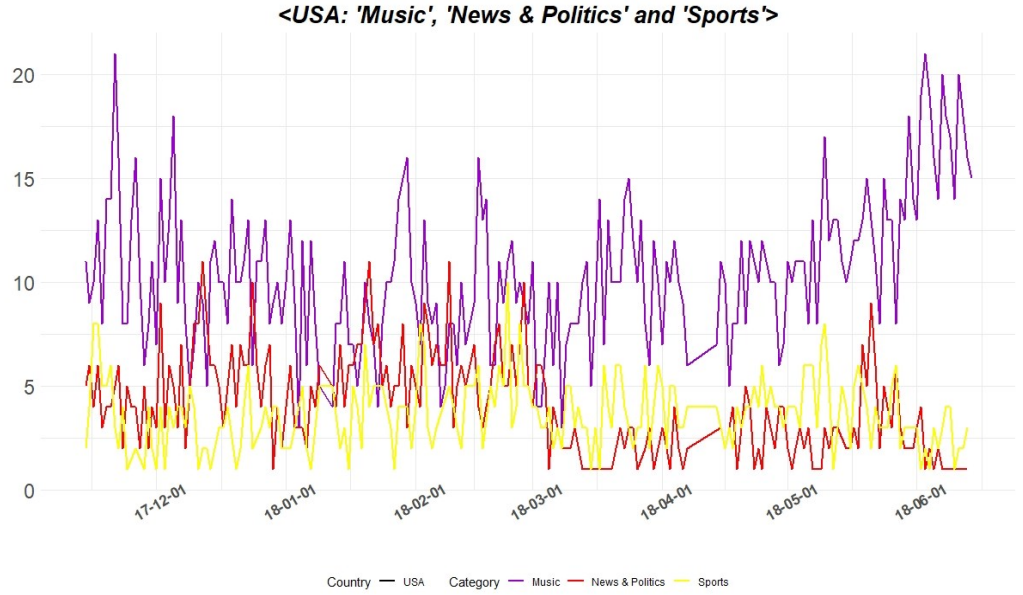
2.3. 미국과 한국의 시기별 News & Politics 분포

인기 영상은 트렌드가 매우 민감하게 반영되는 지표인 만큼, 시계열로 정보를 나열했을 때, 꽤 큰 폭으로 상승과 하강이 반복됨을 보였다. 장기간 증가 혹은 감소 추세를 보인다고 할 수 없었으며, 그날 그날의 이슈들에 즉각 반응하는 그래프 양상을 보였다.

모든 국가의 모든 Category 가 대체로 유사한 그래프를 그렸지만, 그 중에서도 한국과 미국의 News & Politics 그래프에서 특징적인 정보들이 있었다.



2016 년 국정 농단이라는 큰 이슈가 있었던 이래로 한국 유튜브에서는 정치관련 컨텐츠가 사람들에게 많은 관심을 받았다. 이는 앞서 살펴 본, 국가별 Category id 그래프에서도 확인할 수 있었지만, 일별 'News & Politics' 카테고리에 해당하는 인기동영상수를 표현한 그래프에서도 더 확연히 드러났다. 아래 점선 그래프가 미국에서의 'News & Politics' 인기동영상수의 변동추이이다. 계속 높은 위치에 있는 한국 그래프와 달리 미국 그래프는 18 년 3 월부터 급락하여 계속 낮은 위치에 있다. 유튜브를 통해 뉴스와 정치를 접하는 방식이 두 국가가 매우 달랐다. 한국은 실제로 정치를 소재로 하는 유튜브가 인기동영상에 많이 포진해 있는 반면, 미국은 평창올림픽 등 스포츠, 연예 관련 뉴스가 주로 인기동영상에 있었다. 18 년 3 월 이후 급락하는 것도 평창 올림픽이 끝나는 시기와 일치한다.

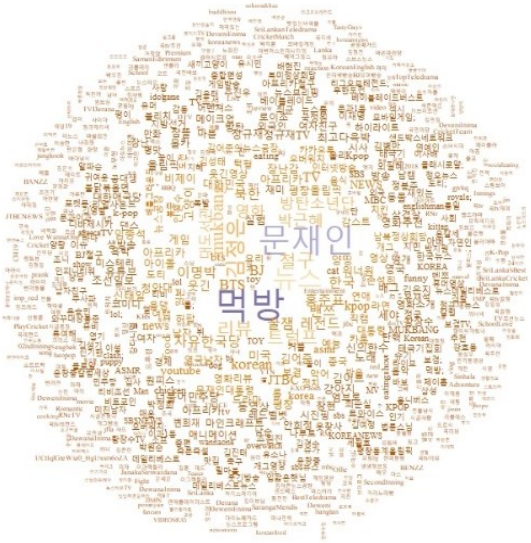


이는 다음 그래프에서도 확인할 수 있다. 18 년 6 월에는 북미정상회담이라는 정치적으로 매우 큰 이슈가 있었음에도 오히려 'Music' Category 가 크게 상승함을 확인할 수 있다.

2.4. 미국과 한국의 Tag 분포

<USA: Word Cloud of Tags>

<Korea: Word Cloud of Tags>

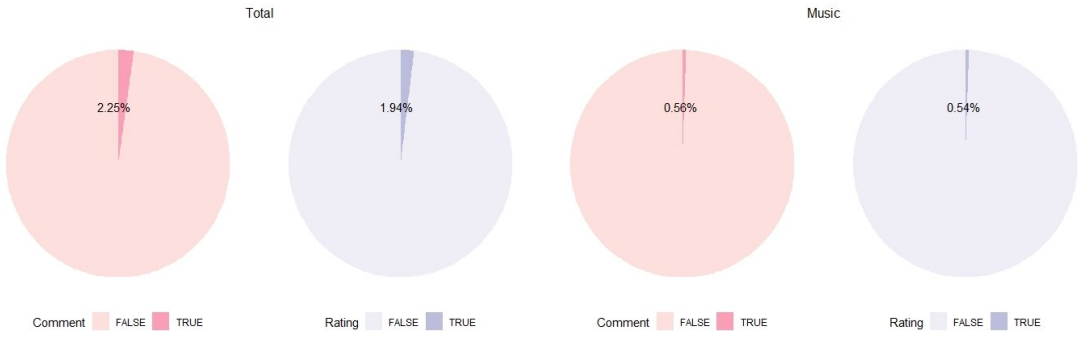


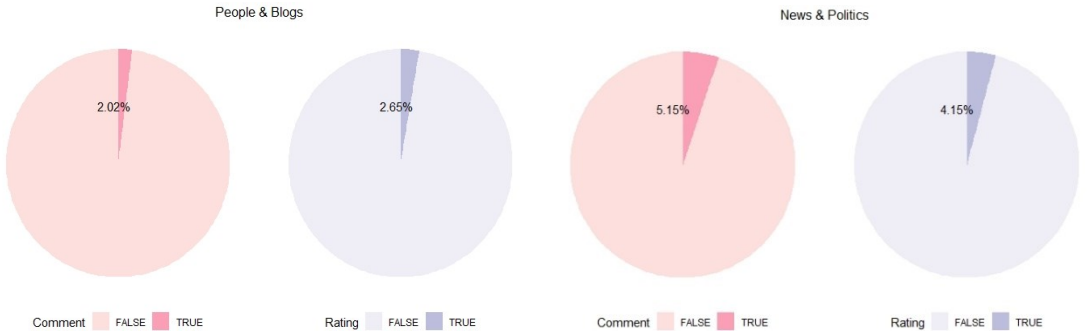
두 국가의 유튜브 환경 차이는 Tag 데이터를 이용해 만든 Word Cloud 에서도 잘 드러났다. 미국은 funny, comedy, Pop, humor 등 연예, 오락 관련 Tag 들이 다수 등장하는 데에 반해, 한국은 '문재인', '김정은', '박근혜', '트럼프' 등의 정치인들의 실존 이름과 '자유한국당', '북한' 등 정치 관련 Tag 들이 다수 등장했다. 이를 통해 한국이 미국보다 유튜브를 정치적 의견을 표출하는 공간으로 더 사용하는 경향이 있다고 이야기할 수 있다.

3. 국가 및 Category 별 views / likes / dislikes / comments 분포

3.1. Category 별 comment/rating 비활성화 비율

<Total: Disable Percentage>

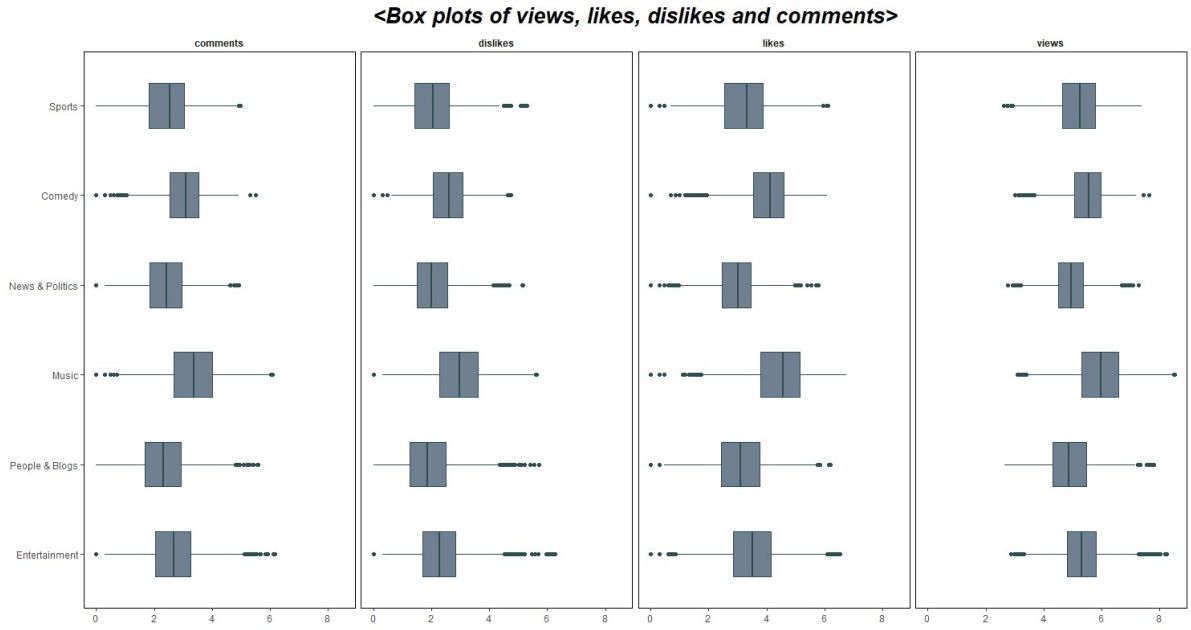




먼저 Category 별로 데이터 댓글 기능 혹은 좋아요/싫어요 기능의 차단 유무를 파악했다. 전체 인기 영상 중 약 2.25%가 댓글 기능을, 약 1.94%가 좋아요/싫어요 기능을 차단했다.

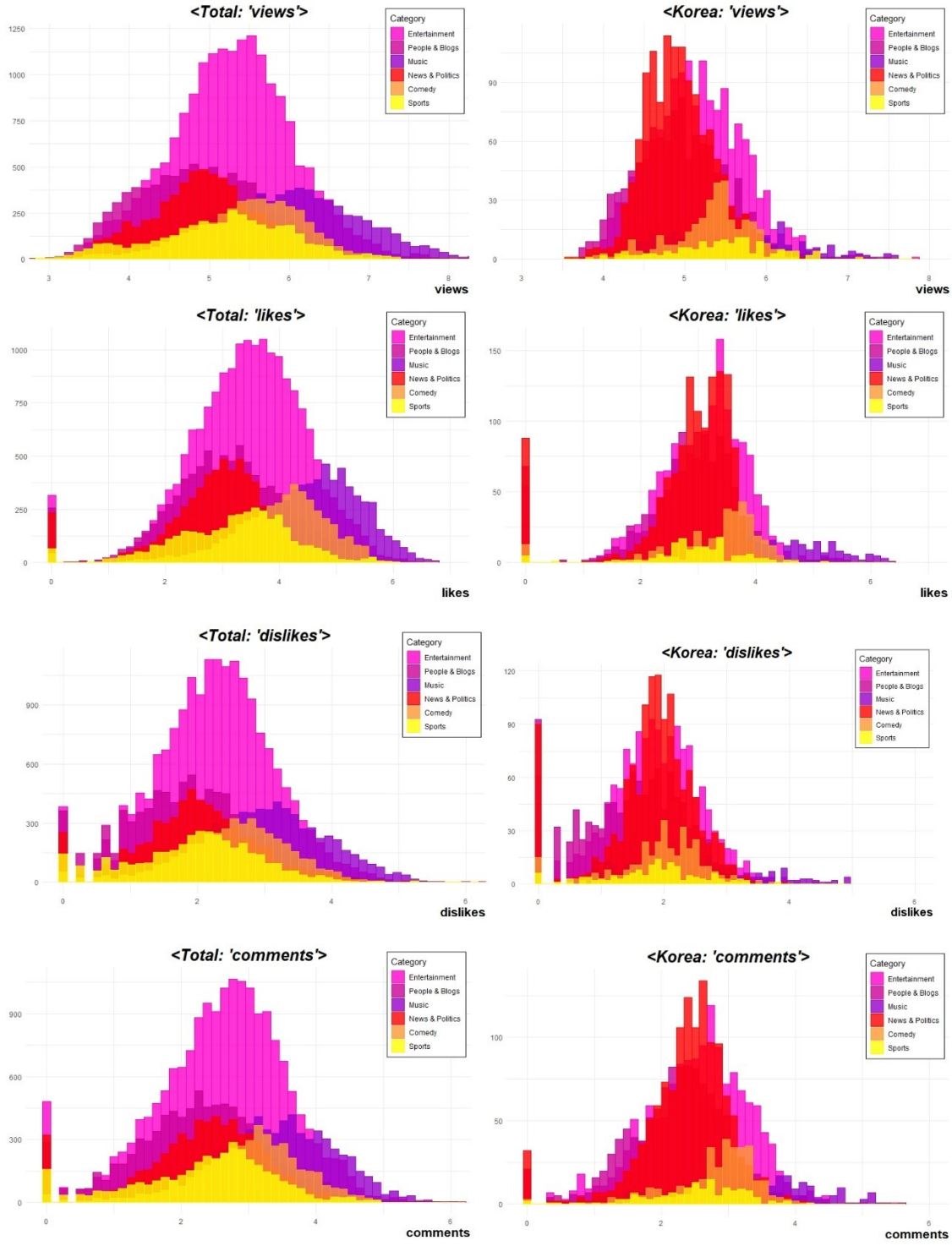
하지만 Category 별로 나눠서 파악했을 때는 다른 양상을 보였는데, Music 은 이 기능들을 차단한 영상이 거의 없었다. 음악을 홍보하고 팬덤과 소통하는 매개체로써 유튜브를 사용하는 Music 유튜버들은 댓글과 좋아요 기능이 매우 중요하기 때문이다. 반면, News & Politics 는 이런 기능을 차단한 영상의 비율이 타 영상들보다 높았다. 정치적 혹은 사회적으로 논란이 되는 영상의 경우 다수의 싫어요와 부정적인 댓글들을 받을 가능성이 높다. 이를 사전에 차단하고자 해당 기능들을 차단한 영상이 다른 Category 에 비해 많은 것이라고 생각한다.

3.2. Category 별 view / likes / dislikes / comments 상관관계



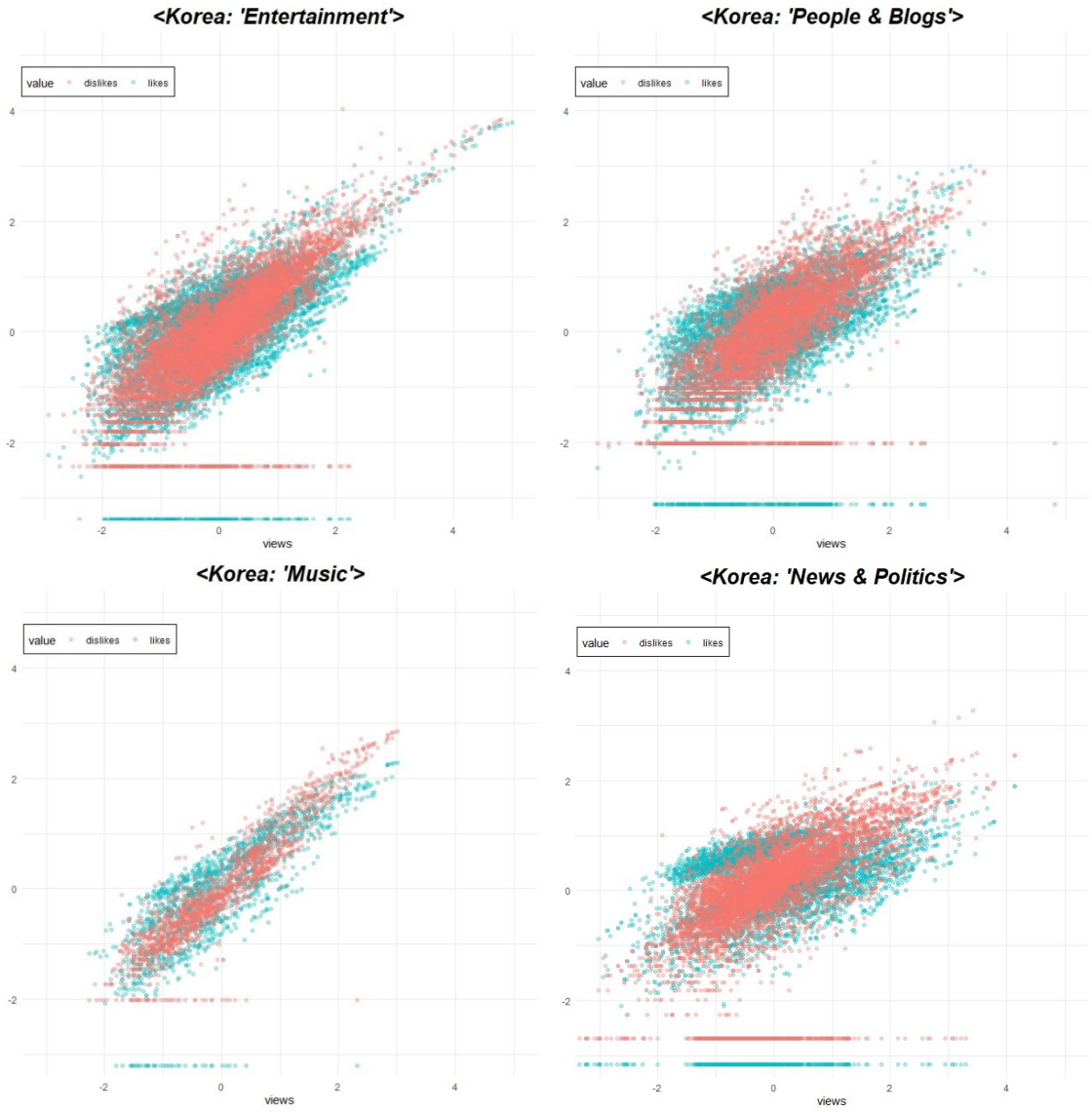
세계적으로 가장 많은 수로 분포되어 있는 상위 6 개 Category('Entertainment', 'People & Blogs', 'Music', 'News & Politics', 'Comedy', 'Sports')들별 각 수치 데이터(views, likes, dislikes, comments)의 Box plot 을 확인했다. 4 개 수치데이터를 모두 영상에 대한 관심도를 나타내는 지표로 판단했을 때, Category 별로 유사한 형태를 보였다. Music 이 6 개 Category 중 대체로 높은 관심을 받는 것으로 사료된다. 이는 유튜브가 음악산업의 미디어 마케팅에서 중요한 위치를 차지하고 있음을 시사한다.

최근 '방탄소년단', '블랙핑크' 등 세계적으로 인기를 얻는 K-Pop 가수들도 뮤직비디오나 공연 영상을 주 홍보수단으로 삼고 있다.



다음은 각 수치 데이터의 히스토그램을 통해 각 Category 별로 데이터가 어떻게 분포해 있는지를 확인하고자 했다. 각 수치데이터들 모두 로그화했을 때 정규분포 형태를 띠었다. Category 별로 색 구분을 했을 때 Box plot 에서 확인했듯이 Music 이 각 수치 상에서 높은 위치에 있었다.

한국 데이터와도 비교를 해보았다. 앞서 살펴봤듯이 한국은 News & Politics 가 인기동영상에 다수 있었으며 People & Blogs 와 유사하게 분포했다. 한국에서도 그 수는 적었으나 Music 이 각 수치상에서 높은 위치에 있었다.

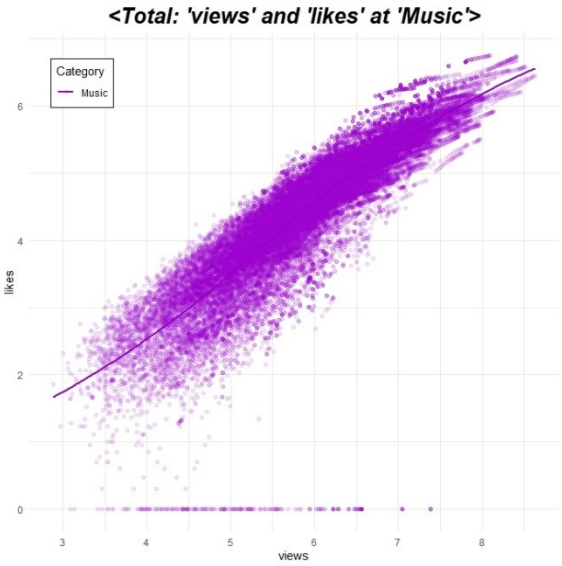


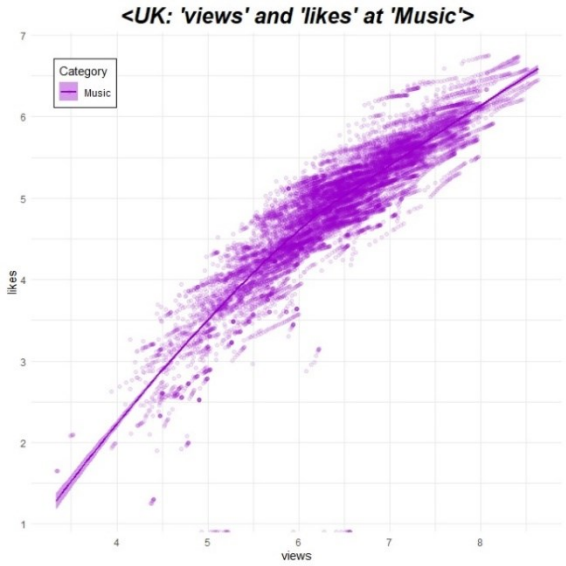
그 다음 한국에서의 각 Category 별 views 와 likes, dislikes 간의 상관관계를 파악했다. likes 와 dislikes 모두 views 와 양의 상관관계를 갖는 분포를 보였다. 그래프를 그리기 전에는 likes 가 더 강한 상관관계를 가질 것이라고 예상했는데, 오히려 dislikes 가 더 강한 상관관계를 보였다. 사람들이 싫어요를 누르는 영상이 관심이 없어서가 아니라 불쾌감을 주거나 논란이 되는 영상임을 유추할 수 있다.

또 그래프를 그리기 전에는 News & Politics 가 조회수당 싫어요를 제일 많이 받을 것이라고 예상했지만, 실제로는 Category 별로 큰 차이가 없었다. 오히려, Entertainment 와 Music 이 views 와 dislikes 산포도 그래프 기울기가 더 큰 것으로 볼 때, 관심과 동시에 부정적인 피드백을 더 많이 받는 것으로 판단된다.

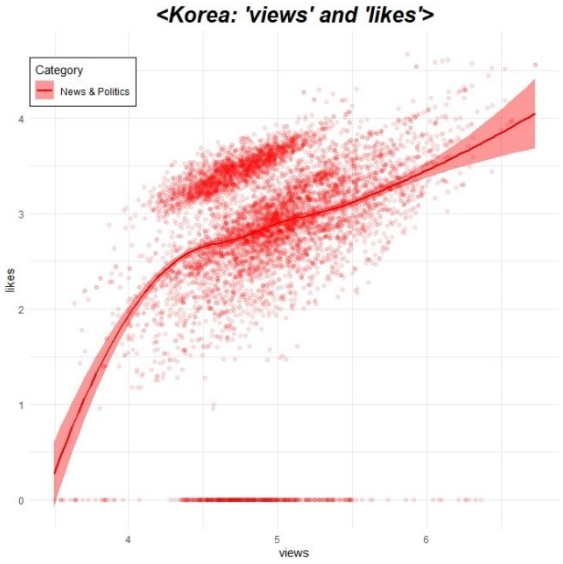
3.3. 국가별 Category 에 따른 산점도

마지막으로 국가별, Category 별 수치형 데이터 간의 상관관계를 분석했다. 많은 수의 조합을 만들어 검토할 수 있었고 이 중 주목할 만한 군집을 보이는 그래프를 본 레포트에 수록했다.



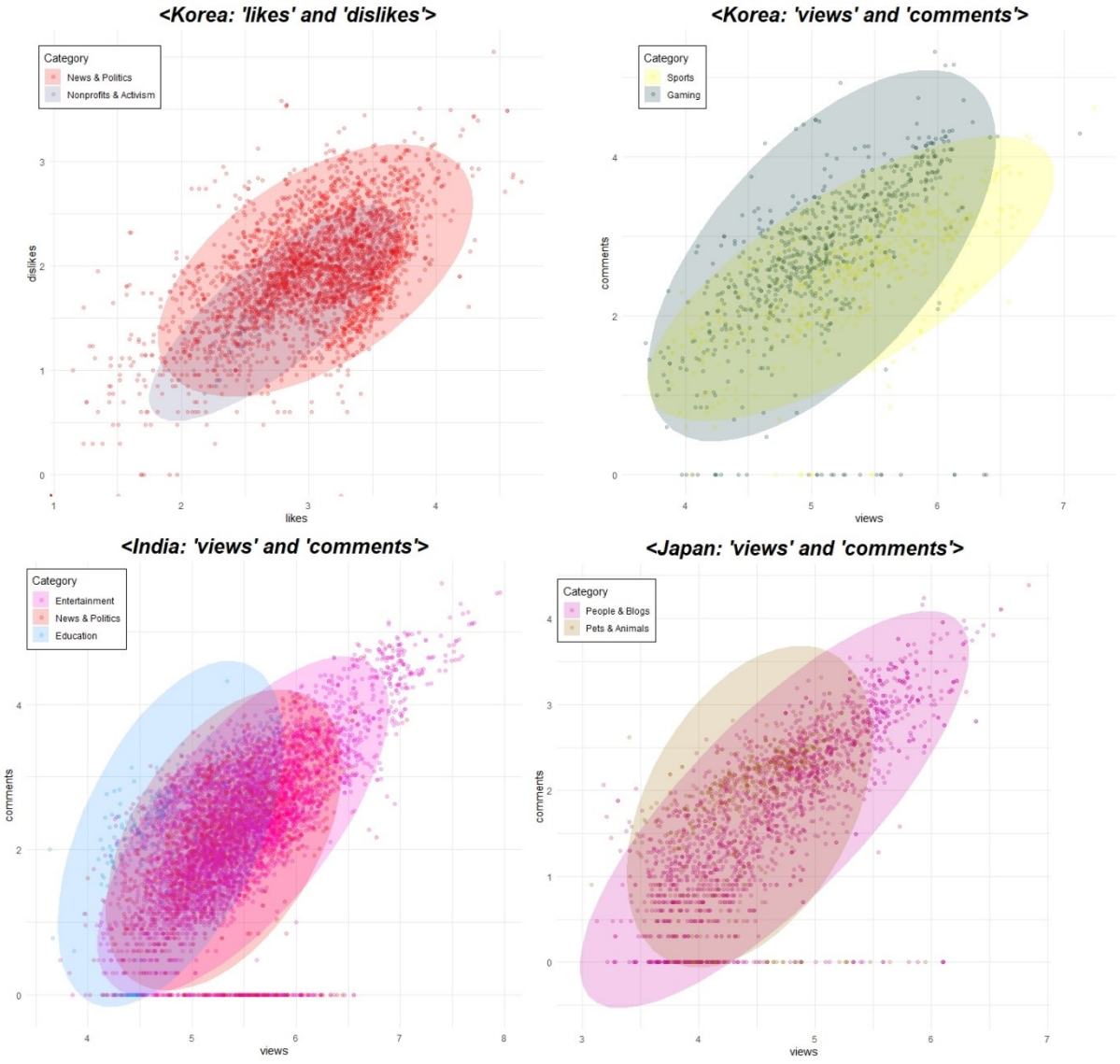


우선 영국에서의 Music 의 경우 타 국가들에 비해 views 와 likes 간의 강한 선형관계를 보였다. 주목할 점은 부분마다 직선을 그리는 형태의 산점도가 나타났는데, 이는 같은 영상이 시간이 지나며 더 높은 조회수와 좋아요수를 받으며 나타난 그림이었다. Music 컨텐츠가 다른 Category 의 영상들보다 인기 영상 목록에 오래 등재되면서 발생한 결과로, Music 이 많은 빈도를 차지하는 영국에서 나타나는 눈여겨볼 만한 현상이다.



한국에서의 News & Politics 의 경우, views 와 likes 간의 산점도를 그린 결과, 두 개의 군집이 나타났다. 조회수당 더 높은 좋아요를 받는 영상들과 그렇지 않은 영상들로 나뉘었다. 실제 어떤 데이터들인지 확인하기 위해 likes 의 로그값이 3 보다 큰 영상들을 추출하는 방법을 택했다. 정교하지

않은 방법이기에, 추출된 영상들 간의 공통된 특징을 파악하기는 쉽지 않았다. 같은 News & Politics 인 영상이더라도, 소재와 시청집단이 다양하기 때문에 더 세분화될 것이라고 생각한다. 따라서 추후 연구에서 해당 군집을 더 정교하게 나눌 수 있는 방법을 택해야 할 것이다.



그 다음 Category 별 데이터 분포를 살펴봤다. 한국에서의 Nonprofits & Activism 영상의 분포를 파악하기 위해 News & Politics 영상과 비교했다. News & Politics 보다 좋아요수와 싫어요수를 적게 받았다. 이는 Nonprofits & Activism 영상의 대부분은 수익 창출이 아니기에, 자극적인 소재 혹은 공격적인 마케팅을 하지 않기에 나타나는 현상으로 파악된다.

또 한국에서의 Sports 와 Gaming 간의 views 와 comments 분포를 비교했다. Gaming 이 Sports 보다 눈에 띄게 조회수당 많은 댓글수를 보였다. 이는 해당 Category 들에 관심을 갖는 대상이 누구인지를 고려할 때 예상가능한 결과이다. Sports 를 즐기는 집단보다 Gaming 을 즐기는 집단이 컴퓨터, SNS, 모바일을 통한 소통에 더 익숙할 것이고 따라서 댓글 기능을 보다 더 어려움 없이 사용할 것이다.

상대적으로 다른 나라보다 높은 Category 비율을 보이던 인도에서의 Education 도 다른 Category 로부터 떨어진 군집이 나타났다. Education 의 경우, 해당 영상을 찾아보는 대상이 주로

영상의 주제에 관심있는 학생들일 것으로 사료된다. 다른 Category 에 비해 접근도가 낮을 수 밖에 없으며, 이에 따라 views 는 자연스레 떨어질 것이다. 하지만 comments 는 다른 군집에 비해 낮지 않은데 이는 인도의 높은 교육열이 반영된 결과로 볼 수 있다.

일본에서 Pets & Animals 도 조회수당 높은 댓글수를 보였다. People & Blogs 와 비교했을 때 views 에 비해 comments 가 높고 더 좁은 범위에서 군집이 형성되어 있었다. 이는 Pets & Animals 가 People & Blogs 보다 컨텐츠의 종류에 한계가 있기 때문에 생겨난 현상이라고 파악했다. 주로 반려동물을 영상의 소재로 삼는 만큼 다양한 사람들의 생활 모습을 보여주는 것보다는 조회수가 다양하지 않을 것이며, 따라서 댓글수도 이와 유사하게 분포할 것이다.

4. 결론

2017 년 11 월부터 2018 년 6 월까지의 짧은 시기의 데이터이지만, feature 별로 여러 특징들을 파악하는 데에 충분했다. 국가별로 인기 영상의 Category 빈도와 업로드 시간대가 다른 것을 통해, 문화권마다 영상의 분포가 다양함을 알 수 있었고, 주요 수치 데이터들과 Category 별 상관관계를 파악함으로써 영상마다 사람들에게 관심 받는 양상에 차이가 있음을 파악할 수 있었다.

본 레포트에서는 Youtube 데이터의 다양한 feature 들을 탐색적으로 분석했다. 하나의 종합적인 결론을 내리는 게 아니라 데이터를 시각화해, feature 별 특성을 파악하는 방식이었다. 해당 Youtube 데이터를 특정 목적을 위해 활용한다면 각 feature 들을 더 깊게 분석하는 과정이 필요하다. 예를 들어, 앞서 한국에서 News & Politics 에서 군집이 왜 나타나는지를 정치인들이 분석한다면, 현재 시민들의 정치적인 요구와 관심사항이 무엇인지를 더 자세히 파악할 수 있다. 이처럼 Youtube 는 단순한 여가활동이 아닌, 문화적 트렌드와 사회적 현상을 해석하는 중요 수단이며, 다른 데이터들과 결합했을 때 더 가치 있는 정보를 제공할 수 있다.

R code

Merge.R - 전처리 데이터 불러옴

```
colClasses_vec <- c("character", "character", "factor", "character", "character", "numeric",
"numeric", "numeric", "numeric", "logical", "logical", "logical", "factor", "factor" )
df <- read.csv("data/df.csv", encoding = 'UTF-8', colClasses = colClasses_vec)
df$trending_date <- as.Date(df$trending_date, "20%y-%m-%d")
df$publish_time <- as.POSIXct(df$publish_time, tz = "GMT", "20%y-%m-%d %H:%M:%S")
df$tags <- with(df, strsplit(x = tags, split = "[\\|,#]+"))

category_vector <- c('Film & Animation', 'Music', 'Pets & Animals', 'Sports', 'Travel & Events',
'Autos & Vehicles', 'Gaming', 'People & Blogs', 'Comedy', 'Entertainment', 'News & Politics', 'Howto
& Style', 'Education', 'Science & Technology', 'Nonprofits & Activism', 'Movies', 'Shows',
'Trailers')
publish_vector <- c('Off_Time', 'Work_Time', 'Sleep_Time')
df$category_id <- factor(df$category_id, labels = category_vector)
df$publish_at <- factor(df$publish_at, levels = publish_vector)
df$country <- factor(df$country)

colorCategoryBrewer <-
 c("#FF00CC", "#CC0099", "#9900CC", "#FF0000",
   "#FF9933", "#FFFF00", "#99CC00", "#00CC99",
   "#003333", "#33FF66", "#3399FF", "#333399",
   "#666699", "#330099", "#996600", "#333366",
   "#000099", "#000066")

re_order_factor <- c("Entertainment", "People & Blogs", "Music", "News & Politics", "Comedy",
"Sports", "Film & Animation", "Howto & Style", "Gaming", "Science & Technology", "Education", "Autos
& Vehicles", "Nonprofits & Activism", "Shows", "Pets & Animals", "Movies", "Travel & Events",
"Trailers")
df$category_id <- factor(df$category_id, levels = re_order_factor)
df$category_id_color <- factor(df$category_id, labels = colorCategoryBrewer)
```

EDA_donut.R - 국가별 Category donut chart 작성

```
library(ggplot2)
library(cowplot)
library(gridExtra)

# pie graph
tb <- data.frame(table(df$category_id))
names(tb) <- c("Category", "Freq")
tb$fraction <- tb$Freq / sum(tb$Freq)
tb$ymax <- cumsum(tb$fraction)
```

```
tb$ymin <- c(0, head(tb$ymax, n=-1))

donut <- ggplot(tb, aes(ymax=ymax, ymin=ymin, xmax=4, xmin=3,fill=Category)) + geom_rect()
donut <- donut + coord_polar(theta="y") + xlim(c(2, 4)) + theme_void() + scale_fill_manual(values =
colorCategoryBrewer)
donut <- donut + theme(title = element_text(size = 24, face = "bold.italic"),legend.position =
"right", plot.title = element_text(hjust = 0.5))
donut <- donut + theme(legend.text = element_text(size = 8, face = "plain"))
donut <- donut + theme(legend.title = element_text(size = 12, face = "bold"))
donut <- donut + theme(plot.margin = unit(c(0,0,0,0), "mm")) + theme(legend.box.margin = margin(-5,
-3, -5, -6, "mm"))
donut

donut_legend <- get_legend(donut)
ggdraw(donut_legend)
donut <- donut + ggtitle("Total") + theme(legend.position = "none", title = element_text(size = 12,
face = "bold"), plot.title = element_text(hjust = 0.2, vjust = -1.0), plot.margin = unit(c(-2.5,-
2.5,-2.5,-2.5), "mm"))

drawcategoryDonut <- function(df, country_name) {
 tb <- data.frame(table(subset(df, country == country_name)$category_id))
 names(tb) <- c("Category", "Freq")
 tb$fraction <- tb$Freq / sum(tb$Freq)
 tb$ymax <- cumsum(tb$fraction)
 tb$ymin <- c(0, head(tb$ymax, n=-1))
 donut <- ggplot(tb, aes(ymax=ymax, ymin=ymin, xmax=4, xmin=3,fill=Category)) + geom_rect() +
ggtitle(country_name)
 donut <- donut + coord_polar(theta="y") + xlim(c(2, 4)) + theme_void() + scale_fill_manual(values =
colorCategoryBrewer)
 donut <- donut + theme(title = element_text(size = 10, face = "bold"), plot.title =
element_text(hjust = 0.2))
 donut <- donut + theme(legend.title = element_blank())
 donut <- donut + theme(legend.position = "none") + theme(plot.margin = unit(c(-2.5,-2.5,-2.5,-2.5),
"mm"))
 donut + theme(legend.title = element_text(size = 20, face = "plain"), legend.position = "left",
plot.margin = unit(c(0.2,0.2,0.2,0.2), "mm"), title = element_text(size = 20, face = "bold"),
plot.title = element_text(hjust = 0.3))
 return(donut)
}

donut_CA <- drawcategoryDonut(df, "Canada")
donut_DE <- drawcategoryDonut(df, "Germany")
donut_FR <- drawcategoryDonut(df, "France")
donut_GB <- drawcategoryDonut(df, "UK")
donut_IN <- drawcategoryDonut(df, "India")
donut_JP <- drawcategoryDonut(df, "Japan")
donut_KR <- drawcategoryDonut(df, "Korea")
donut_MX <- drawcategoryDonut(df, "Mexico")
donut_RU <- drawcategoryDonut(df, "Russia")
```

```
donut_US <- drawcategoryDonut(df, "USA")

heights_v <- c( 2, 1, 2, 1, 2)
widths_v <-  c( 2, 2, 2, 2, 1)
lay <- rbind(c( 3, 4, 5, 6, 2),
             c(NA,NA, 1, 1, 2),
             c( 7, 8, 1, 1, 2),
             c(NA,NA, 1, 1, 2),
             c( 9,10,11,12, 2))

panel <- list(donut, donut_CA, donut_DE, donut_FR, donut_GB, donut_IN, donut_JP, donut_KR, donut_MX,
donut_RU, donut_US)

grid.arrange(donut, donut_legend, donut_CA, donut_DE, donut_FR, donut_GB, donut_IN, donut_JP,
donut_KR, donut_MX, donut_RU, donut_US, widths = widths_v, heights = heights_v, layout_matrix = lay,
top = textGrob("<Category Distribution per each Country>\n", gp=gpar(fontsize=24, fontface =
"bold.italic")))
```

EDA_publish.R – 국가별 업로드 시간 분포 pie chart 작성

```
colorVectors_publish_at <- brewer.pal(3, "Blues")

# pie graph
tb <- data.frame(table(df$publish_at))
names(tb) <- c("Publish_Time", "Freq")
tb$fraction <- tb$Freq / sum(tb$Freq)
tb$ymax <- cumsum(tb$fraction)
tb$ymin <- c(0, head(tb$ymax, n=-1))
tb$per <- with(tb, paste0(round(fraction * 100, 1), "%"))

pie <- ggplot(tb, aes(x = "", y = Freq, fill = Publish_Time)) + geom_bar(stat = "identity")
pie <- pie + coord_polar("y") + theme_void()
pie <- pie + scale_fill_brewer(name = "Publish At", palette = "Blues", labels = c("Off Time", "Work
Time", "Sleep Time")) + theme(legend.title = element_text(size = 18, face = "plain"))
pie <- pie + theme(title = element_blank(), legend.position = "right", plot.title =
element_text(hjust = 0.5)) + theme(text = element_text(size = 12))
pie <- pie + theme(plot.margin = unit(c(2.5,2.5,2.5,2.5), "mm")) + theme(legend.box.margin =
margin(-5, 0, -5, -10, "mm"))
pie <- pie + geom_text(aes(label = per),color = c("#3182BD", "white", "white"), position =
position_stack(vjust = 0.5),size = 4) # + theme(text = element_text(size = 18))
pie

pie_legend <- get_legend(pie)
ggdraw(pie_legend)
pie <- pie + ggtitle("Total") + theme(legend.position = "none", title = element_text(size = 12, face
```

```
= "bold"), plot.title = element_text(hjust = 0.2, vjust = -1.0), plot.margin = unit(c(-2.5,-2.5,-
2.5,-2.5), "mm"))


drawPublishPie <- function(df, country_name) {
 tb <- data.frame(table(subset(df, country == country_name)$publish_at))
 names(tb) <- c("Publish_Time", "Freq")
 tb$fraction <- tb$Freq / sum(tb$Freq)
 tb$ymax <- cumsum(tb$fraction)
 tb$ymin <- c(0, head(tb$ymax, n=-1))
 tb$per <- with(tb, paste0(round(fraction * 100, 1), "%"))

 pie<- ggplot(tb, aes(x = "", y = Freq, fill = Publish_Time)) + geom_bar(stat = "identity")
 pie <- pie + coord_polar("y") + theme_void() + ggtitle(country_name)
 pie <- pie + scale_fill_brewer(name = "Publish At", palette = "Blues", labels = c("Off Time", "Work
Time", "Sleep Time")) + theme(legend.title = element_text(size = 18, face = "plain"))
 pie <- pie + theme(title = element_text(size = 10, face="bold"), legend.position = "none",
plot.title = element_text(hjust = 0.5))
 pie <- pie + theme(plot.margin = unit(c(2.5,2.5,2.5,2.5), "mm"))
 pie + geom_text(aes(label = per),color = c("#3182BD", "white", "white"), position =
position_stack(vjust = 0.5),size = 6) # + theme(text = element_text(size = 18))
 return(pie)
}

pie_CA <- drawPublishPie(df, "Canada")
pie_DE <- drawPublishPie(df, "Germany")
pie_FR <- drawPublishPie(df, "France")
pie_GB <- drawPublishPie(df, "UK")
pie_IN <- drawPublishPie(df, "India")
pie_JP <- drawPublishPie(df, "Japan")
pie_KR <- drawPublishPie(df, "Korea")
pie_MX <- drawPublishPie(df, "Mexico")
pie_RU <- drawPublishPie(df, "Russia")
pie_US <- drawPublishPie(df, "USA")

grid.arrange(pie, pie_legend, pie_CA, pie_DE, pie_FR, pie_GB, pie_IN, pie_JP, pie_KR, pie_MX,
pie_RU, pie_US, widths = widths_v, heights = heights_v, layout_matrix = lay, top =
textGrob("<Publish Time Distribution per each Country>\n", gp=gpar(fontsize=24, fontface =
"bold.italic")))
```

EDA_disable.R – comments, ratings 차단 여부 그래프 작성

```
drawPieDisable <- function(df, country_name = NULL, category_name = NULL, disable_which =
c("comments_disabled", "ratings_disabled")) {
 if (is.null(country_name)) {
   temp_df <- df
```

```
}
 else {
   temp_df <- subset(df, country == country_name)
}

 if (!is.null(category_name)) {
   temp_df <- subset(temp_df, category_id == category_name)
}

 temp_df <- temp_df[c("country", "category_id", disable_which)]
 names(temp_df) <- c("Country", "Category", "Comment", "Rating")

 tb_1 <- data.frame(table(temp_df$Comment))
 names(tb_1) <- c("Bool", "Comment")
 tb_2 <- data.frame(table(temp_df$Rating))
 names(tb_2) <- c("Bool", "Rating")
 tb <- merge(tb_1, tb_2, by = "Bool")

 comment_disable_per <- paste0(round(tb$Comment[2] / sum(tb$Comment) * 100,2), "%")
 rating_disable_per <- paste0(round(tb$Rating[2] / sum(tb$Rating) * 100,2), "%")

 comment_gg <- ggplot(tb, aes(x = "", y = Comment, fill = Bool)) + geom_bar(stat = "identity")
 comment_gg <- comment_gg + coord_polar("y") + theme_void() + scale_fill_brewer(name = "Comment",
palette = "RdPu") + theme(legend.position = "bottom")
 comment_gg <- comment_gg + geom_text(aes(y = cumsum(Comment) - 0.5*Comment, label =
c(NA,comment_disable_per)), size = 4)

 rating_gg <- ggplot(tb, aes(x = "", y = Rating, fill = Bool)) + geom_bar(stat = "identity")
 rating_gg <- rating_gg + coord_polar("y") + theme_void() + scale_fill_brewer(name = "Rating",
palette = "Purples") + theme(legend.position = "bottom")
 rating_gg <- rating_gg + geom_text(aes(y = cumsum(Comment) - 0.5*Comment, label =
c(NA,rating_disable_per)), size = 4)

 if (is.null(category_name)) {
   category_name = "Total"
}

 temp_gg <- grid.arrange(comment_gg, rating_gg,
                         top = textGrob(sprintf("%s", category_name, size = 16)),
                         ncol = 2)

 return(temp_gg)
}
disable_which = c("comments_disabled", "ratings_disabled")

pie_disable_news <- drawPieDisable(df, category_name = c("News & Politics"))
pie_disable_news
pie_disable_music <- drawPieDisable(df, category_name = c("Music"))
pie_disable_music
```

```
pie_disable_people <- drawPieDisable(df, category_name = c("People & Blogs"))
pie_disable_people

pie_disable_news_kr <- drawPieDisable(df, country_name = "Korea",category_name = c("News &
Politics"))
pie_disable_news_kr

pie_disable <- drawPieDisable(df)
pie_disable

pie(tb$Comment)
pie(tb$Rating)
```

EDA_category_line.R – 날짜별 Category 변동 추이 그래프 작성

```
colorCategoryBrewer <-
 c("#FF00CC", "#CC0099", "#9900CC", "#FF0000",
   "#FF9933", "#FFFF00", "#99CC00", "#00CC99",
   "#003333", "#33FF66", "#3399FF", "#333399",
   "#666699", "#330099", "#996600", "#333366",
   "#000099", "#000066")

drawCategoryLine <- function(df, country_name, category_name) {
 temp_df <- subset(df, country == country_name)
 temp_df <- temp_df[c('trending_date', 'category_id', 'country')]
 temp_df$count <- c(1)

 ix <- match(category_name, re_order_factor)
 line_color <- colorCategoryBrewer[ix]
 temp_df <- with(subset(temp_df, category_id == category_name), aggregate(count, list(trending_date,
category_id, country), sum))
 names(temp_df) <- c("date", "Category", "Country","Freq")

 temp_gg <- ggplot(temp_df, aes(x = date, y = Freq)) + geom_line(aes(color = Category, linetype =
Country), size = 0.8) + xlab("") + ylab("")
 temp_gg <- temp_gg + scale_x_date(date_labels = "%y-%m-%d", date_breaks = "1 month",
limit=c(as.Date("2017-11-14"),as.Date("2018-06-14")))
 temp_gg <- temp_gg + theme_minimal() + theme(axis.text.x=element_text(angle = 30, size = 13, face =
"bold"), axis.text.y=element_text(size = 17) ,legend.position = "bottom")
 temp_gg <- temp_gg + scale_color_manual(values = line_color)
 return(temp_gg)
}

line_CA <- drawCategoryLine(df, "Canada", "News & Politics")
line_CA
```

```
line_US <- drawCategoryLine(df, "USA", c("Music", "News & Politics", "Sports"))
line_US + ggtitle("<USA: 'Music', 'News & Politics' and 'Film & Animation'>") + theme(plot.title =
element_text(size = 20, face = "bold.italic", hjust = 0.5))

line_KR <- drawCategoryLine(df, "Korea", c("Music", "Film & Animation"))
line_KR + ggtitle("<Korea: 'Music' and 'Film & Animation'>") + theme(plot.title = element_text(size
= 20, face = "bold.italic", hjust = 0.5))

line_KRandUS <- drawCategoryLine(df, c("Korea", "USA"), c("News & Politics"))
line_KRandUS

line_KRandUS + ggtitle("<Korea and USA: 'News & Politics'>") + theme(plot.title = element_text(size
= 20, face = "bold.italic", hjust = 0.5))
line_KR + ggtitle("<Korea: 'News & Politics' and 'Comedy'>") + theme(plot.title = element_text(size
= 20, face = "bold.italic", hjust = 0.5))
```

EDA_histo.R – Category 별 히스토그램 작성

```
drawHistogram <- function(df, country_name=NULL, numeric_name, category_name=NULL, position_name =
"stack") {
 if (is.null(country_name)) {
   temp_df <- df
}
 else {
   temp_df <- subset(df, country == country_name)
}

 if (is.null(category_name)) {
   temp_df <- temp_df[c(numeric_name)]+1
   temp_df <- data.frame(apply(temp_df, 1, log10))
   names(temp_df) <- c("value")
   temp_gg <- ggplot(temp_df, aes(x = value)) + geom_histogram(fill = "gray", color = "darkgray",
alpha = 0.8, bins = 50) + theme_minimal()
   temp_gg
}
 else {
   temp_df <- temp_df[c(numeric_name, "category_id")]
   temp_df <- subset(temp_df, category_id == category_name)
   names(temp_df) <- c("value", "Category")
   temp_df$value <- log10(temp_df$value+1)
   ix <- match(category_name, re_order_factor)
   line_color <- colorCategoryBrewer[ix]
   temp_df$Category <- factor(temp_df$Category)

   temp_gg <- ggplot(temp_df, aes(x = value, color = Category, fill = Category)) +
```

```
geom_histogram(alpha = 0.8, bins = 50, position = position_name) + theme_minimal()
   temp_gg <- temp_gg + scale_color_manual(values = line_color) + scale_fill_manual(values =
line_color) + theme(legend.position = c(0.9,0.85), legend.background = element_rect(fill = "white",
color = "black"),legend.direction = "vertical")
}

 if (numeric_name == "comment_count") {
   x_name = "comments"
   x_lim = c(0,6)
}
 else {
   x_name = numeric_name
   if (numeric_name == "views") {
     x_lim = c(3,8)
  }
   else if (numeric_name == "likes") {
     x_lim = c(0,7)
  }
   else if (numeric_name == "dislikes") {
     x_lim = c(0,6)
  }
}

 temp_gg <- temp_gg + xlab(x_name) + ylab("") + coord_cartesian(xlim = x_lim)
 temp_gg <- temp_gg + theme(axis.title.x = element_text(face = "bold", angle = 0, size = 16, hjust =
1.0))

 return(temp_gg)
}


hist_CA <- drawHistogram(df, country_name = "Canada", numeric_name = "views")
hist_CA

hist_CA_category <- drawHistogram(df, country_name = "Canada", numeric_name = "views", category_name
= "Sports")
hist_CA_category

hist_USA <- drawHistogram(df, country_name = "USA", numeric_name = "views")
hist_USA

hist_KR <- drawHistogram(df, country_name = "Korea", numeric_name = "dislikes")
hist_KR+ ggtitle("<Korea: 'dislikes'>") + theme(plot.title = element_text(size = 20, face =
"bold.italic", hjust = 0.5))

select <- "dislikes"

hist_KR_category <- drawHistogram(df, country_name = "Korea", numeric_name = select, category_name =
c("Entertainment", "People & Blogs", "Music", "News & Politics", "Comedy", "Sports"), position_name
```

```
= "identity")
hist_KR_category + ggtitle(sprintf("<Korea: '%s'>", select)) + theme(plot.title = element_text(size
= 20, face = "bold.italic", hjust = 0.5))

hist <- drawHistogram(df, numeric_name = "views", category_name = c("Entertainment", "People &
Blogs", "News & Politics", "Comedy", "Film & Animation"))
hist <- drawHistogram(df, numeric_name = select, category_name = c("Entertainment", "People &
Blogs", "Music", "News & Politics", "Comedy", "Sports"), position_name = "identity")
hist+ ggtitle(sprintf("<Total: '%s'>", select)) + theme(plot.title = element_text(size = 20, face =
"bold.italic", hjust = 0.5))
```

EDA_scatter.R – Category 별 산점도 그래프 작성

```
drawScatter <- function(df, country_name=NULL, numeric_name1, numeric_name2, category_name=NULL,
elipse_on = TRUE, se_on = TRUE) {
 if (is.null(country_name)) {
   temp_df <- df
}
 else {
   temp_df <- subset(df, country == country_name)
}

 if (is.null(category_name)) {
   temp_df <- temp_df[c(numeric_name1, numeric_name2)]
   names(temp_df) <- c("value1", "value2")
   temp_df$value1 <- log10(temp_df$value1+1)
   temp_df$value2 <- log10(temp_df$value2+1)
   temp_gg <- ggplot(temp_df, aes(x = value1, y = value2)) + geom_point(color = "black", alpha =
0.1) + theme_minimal()
   temp_gg <- temp_gg + theme(legend.position = c(0.1,0.9), legend.background = element_rect(fill =
"white", color = "black"),legend.direction = "horizontal")
}
 else {
   temp_df <- temp_df[c(numeric_name1, numeric_name2, "category_id")]
   temp_df <- subset(temp_df, category_id == category_name)
   names(temp_df) <- c("value1", "value2", "Category")
   temp_df$value1 <- log10(temp_df$value1+1)
   temp_df$value2 <- log10(temp_df$value2+1)
   ix <- match(category_name, re_order_factor)
   line_color <- colorCategoryBrewer[ix]
   temp_df$Category <- factor(temp_df$Category)

   if (length(category_name) == 1) {
     temp_gg <- ggplot(temp_df, aes(x = value1, y = value2, color = Category, fill = Category)) +
geom_point(alpha = 0.1) + stat_smooth(method = loess, se = se_on) + theme_minimal()
```

```
  }
   else {
     temp_gg <- ggplot(temp_df, aes(x = value1, y = value2, color = Category, fill = Category)) +
geom_point(alpha = 0.2) + theme_minimal()
     if (elipse_on) {
       temp_gg <- temp_gg + stat_ellipse(geom = "polygon", type = "norm", alpha = 0.2, color = NA)
    }
  }

   temp_gg <- temp_gg + scale_color_manual(values = line_color) + scale_fill_manual(values =
line_color) + theme(legend.position = c(0.1,0.9), legend.background = element_rect(fill = "white",
color = "black"),legend.direction = "vertical")
}

 if (numeric_name1 == "comment_count") {
   x_name = "comments"
} else {
   x_name = numeric_name1
}
 if (numeric_name2 == "comment_count") {
   y_name = "comments"
} else {
   y_name = numeric_name2
}

 temp_gg <- temp_gg + xlab(x_name) + ylab(y_name)

 return(temp_gg)
}

scatter <- drawScatter(df, numeric_name1 = "views", numeric_name2 =
"likes",category_name=c("Music"), se_on = FALSE)
scatter + ggtitle("<Total: 'views' and 'likes' at 'Music'>") + theme(plot.title = element_text(size
= 20, face = "bold.italic", hjust = 0.5))

scatter_CA <- drawScatter(df, country_name = "Canada", "views", "likes", c("News & Politics",
"Comedy"))
scatter_CA

scatter_KR <- drawScatter(df, country_name = "Korea", "views", "likes", c("Music"))
scatter_KR + ggtitle("<Korea: 'views' and 'likes'>") + theme(plot.title = element_text(size = 20,
face = "bold.italic", hjust = 0.5))


scatter_GB <- drawScatter(df, country_name = "UK", "views", "likes", "Music")
scatter_GB + ggtitle("<UK: 'views' and 'likes' at 'Music'>") + theme(plot.title = element_text(size
= 20, face = "bold.italic", hjust = 0.5))

scatter_RU <- drawScatter(df, country_name = "Russia", "views", "comment_count")
```

```
scatter_RU + ggtitle("<Russia: 'views' and 'comments'>") + theme(plot.title = element_text(size =
20, face = "bold.italic", hjust = 0.5))

scatter_IN <- drawScatter(df, country_name = "India", "views", "comment_count", c("Entertainment",
"News & Politics", "Education"))
scatter_IN + ggtitle("<India: 'views' and 'comments'>") + theme(plot.title = element_text(size = 20,
face = "bold.italic", hjust = 0.5))

scatter_FR <- drawScatter(df, country_name = "France", "dislikes", "likes", c("Entertainment",
"People & Blogs", "Music", "News & Politics", "Comedy"))
scatter_FR + ggtitle("<France: 'dislikes' and 'likes'>") + theme(plot.title = element_text(size =
20, face = "bold.italic", hjust = 0.5))

scatter_JP <- drawScatter(df, country_name = "Japan", "views", "comment_count", c("People & Blogs",
"Pets & Animals"))
scatter_JP + ggtitle("<Japan: 'views' and 'comments'>") + theme(plot.title = element_text(size = 20,
face = "bold.italic", hjust = 0.5))
```

EDA_scatter2.R – views, likes, dislikes 사이의 산점도 그래프 작성

```
drawScatter2 <- function(df, country_name=NULL, numeric_name1, numeric_name2, numeric_name3,
category_name=NULL, elipse_on = TRUE) {
 if (is.null(country_name)) {
   temp_df <- df
}
 else {
   temp_df <- subset(df, country == country_name)
}

 if (!is.null(category_name)) {
   temp_df <- subset(temp_df, category_id == category_name)
}

 temp_df <- temp_df[c(numeric_name1, numeric_name2, numeric_name3)]
 names(temp_df) <- c("x", "y1", "y2")
 temp_df$x <- scale(log10(temp_df$x+1))
 temp_df$y1 <- scale(log10(temp_df$y1+1))
 temp_df$y2 <- scale(log10(temp_df$y2+1))

 temp_df <- reshape(data = temp_df,
                    varying = 2:3,
                    v.names = "y",
                    direction = "long",
                    timevar = "value",
                    times = c(numeric_name2, numeric_name3))
 temp_df$value <- factor(temp_df$value)
```

```
 temp_gg <- ggplot(temp_df, aes(x = x, y = y, color = value, fill = value)) + geom_point(alpha =
0.3) + theme_minimal()
 if (elipse_on) {
   temp_gg <- temp_gg + stat_ellipse(geom = "polygon", type = "norm", alpha = 0.2, color = NA)
}

 x_lim = c(-3.0,5.0)
 y_lim = c(-3.0,5.0)

 temp_gg <- temp_gg + xlab(numeric_name1) + ylab("") + coord_cartesian(xlim = x_lim, ylim = y_lim)
 temp_gg <- temp_gg + theme(legend.position = c(0.15,0.9), legend.background = element_rect(fill =
"white", color = "black"),legend.direction = "horizontal")

 return(temp_gg)
}

select <- "People & Blogs"

scatter_KR2 <- drawScatter2(df, country_name = "Korea", "views", "likes", "dislikes", select,
elipse_on = FALSE)
scatter_KR2 + ggtitle(sprintf("<Korea: '%s'>", select)) + theme(plot.title = element_text(size = 20,
face = "bold.italic", hjust = 0.5))
```

EDA_word_cloud.R – tag 를 이용한 Word Cloud 작성

```
library(wordcloud)

word_df <- subset(df, country == "USA") # change this to Korea

tag_list <- with(word_df, strsplit(x = tags, split = "[\\|,#]+"))
tag_list <- unlist(tag_list)

write.csv(tag_list, "data/tag_US.csv", row.names = FALSE)
''' preprocessed at Python3 '''
tag_df <- read.csv("data/text_US.txt", na.strings = "[none]", colClasses = c("character"))

wordcount <- table(tag_df$x)

word_df <- as.data.frame(wordcount, stringAsFactors = FALSE)
head(word_df)
names(word_df) <- c("word", "Freq")

word_df <- subset(word_df, Freq >= 2)
```

```r
word_df <- subset(word_df, !grepl(pattern = "<", x = word, fixed = FALSE))

pal <- brewer.pal(8, "PuOr")
par(mar = rep(0, 4))
word_cloud <- wordcloud(words = word_df$word,
                        freq = word_df$Freq,
                        min.freq = 10,
                        max.words = 800,
                        random.order = FALSE,
                        rot.per = .1,
                        scale = c(4, 0.3),
                        colors = pal,
                        family = "serif")
```