소셜 네트워크 분석 기말 보고서

박규훤, 소다영, 윤성연

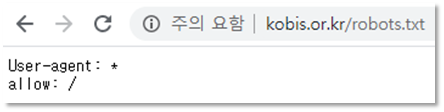
2020 7 1

## 주제

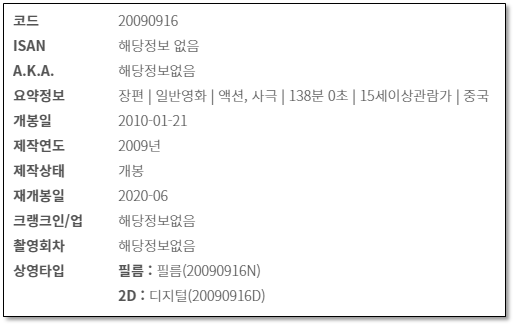
* 특정 감독이 어떤 배우와 많은 영화를 찍었고, 그 사람들이 어떤 특징을 가지는지 분석
* 감독과 배우, 배우와 배우 사이의 캐미를 확인
* 비슷한 상황의 감독이 선택한 배우들의 성향, 및 배우들의 조합 확인

## 데이터

### 출처

* KOBIS (영화관 입장권 통합 전산망)
* <http://www.kobis.or.kr/kobis/business/mast/mvie/searchMovieList.do>
* 
* 로봇 배제 표준

### 소개

* KOBIS의 전국 영화 박스오피스, 영화 별 상세 정보, 관계자(감독, 주연 배우) 정보
* 
* 박스오피스
* 
* 영화 상세 정보
* 
* 관계자 정보

### 크롤링

* Python을 이용하여 크롤링 진행 (crawling.ipynb 파일 별첨)
* 빈 값 및 영화 모음 등에 대해 간단한 전처리 진행

# 빈 값 및 영화 모음 등에 대한 전처리  
  
Sys.setlocale('LC\_ALL', 'C')  
movie <- read.csv('movie\_list.csv', encoding = 'UTF-8')  
actor <- read.csv('staff\_list.csv', encoding = 'UTF-8')  
Sys.setlocale('LC\_ALL', 'korean')  
  
movie[str\_detect(movie$genre,'.+분.+초$'),]$genre <- ''  
movie[str\_detect(movie$country,'.+분.+초$'),]$country <- ''  
movie <- movie[!str\_detect(movie$genre,'기타'),]  
movie <- movie[movie$genre != '',]  
actor <- actor[!is.na(actor$person\_id),]  
  
write.csv(movie, 'pre\_movie.csv', row.names = F, fileEncoding = 'UTF-8')  
write.csv(actor, 'pre\_actor.csv', row.names = F, fileEncoding = 'UTF-8')

### 데이터 예시

movie <- read.csv('pre\_movie.csv', encoding = 'UTF-8', stringsAsFactors = F)  
staff <- read.csv('pre\_actor.csv', encoding = 'UTF-8', stringsAsFactors = F)

## title title\_en code status  
## 1 겁쟁이 페달: 리로드 Yowamushi Pedal: Re:ROAD 20151005 개봉  
## 2 겁쟁이 페달: 리라이드 Yowamushi Pedal: Re:RIDE 20151004 개봉  
## 3 겁쟁이 페달: 더 무비 Yowamushi Pedal: The movie 20159965 개봉  
## 4 겁쟁이 페달: 스페어 바이크 Yowamushi Pedal: SPARE BIKE 20163464 개봉  
## 5 킬러의 보디가드 무삭제 특별판 The Hitman's Bodyguard 20189987 개봉  
## 6 날씨의 아이 Weathering With You 20190280 개봉  
## country genre open  
## 1 일본 애니메이션 2016-01-14  
## 2 일본 애니메이션 2016-01-14  
## 3 일본 애니메이션 2016-01-14  
## 4 일본 애니메이션 2016-11-17  
## 5 미국 액션,코미디 2018-09-12  
## 6 일본 애니메이션,판타지,멜로/로맨스 2019-10-30

## movie\_id person\_category person\_name person\_id  
## 1 20151005 director 나베시마 오사무 10006891  
## 2 20151004 director 나베시마 오사무 10006891  
## 3 20159965 director 나가누마 노리히로 10057861  
## 4 20163464 director 나베시마 오사무 10006891  
## 5 20189987 director 패트릭 휴즈 20136404  
## 6 20189987 actor 라이언 레이놀즈 10014018

### 개요

* 기간: 2010. 01. 01 ~ 2019. 12. 31
* 영화 데이터
  + 75개 국가, 20개 장르
  + 11880 x 7
* 관계자 데이터
  + 11738개 영화, 16706명 관계자
  + 35699 x 4

## 전처리

### 장르 분리

* 앞서 보인 영화 데이터와 같이, 한 영화가 다양한 장르에 포함되어 있음
* 따라서 단일 장르로 구성된 여러개의 영화 데이터로 재구성 과정이 필요

movie <- movie %>% mutate(genre = strsplit(genre, ',')) %>% unnest(genre)

### 관계자 데이터 재구성

* 영화별 함께 출연 또는 작업한 관계자 사이의 연결을 데이터 프레임으로 재구성
* 감독과 주연배우 분리
* actor <- staff %>% filter(person\_category == 'actor')  
  director <- staff %>% filter(person\_category == 'director')
* 감독 - 배우 데이터
  + 영화 별 각 감독과 각 주연 배우 사이의 연결을 표현
* director\_data <- merge(director[c('movie\_id', 'person\_id')], actor[c('movie\_id', 'person\_id')], by='movie\_id')  
  director\_data['person\_id.x'] <- paste0('D', director\_data[['person\_id.x']])  
  director\_data['person\_id.y'] <- paste0('A', director\_data[['person\_id.y']])
* ## movie\_id person\_id.x person\_id.y  
  ## 1 19398001 D10033202 A10068907  
  ## 2 19538011 D10050769 A10073138  
  ## 3 19538011 D10050769 A20126569  
  ## 4 19618009 D10032825 A10067252  
  ## 5 19618009 D10032825 A10050206  
  ## 6 19648012 D10019986 A10067700
* 배우 - 배우 데이터
  + 영화 별 각 주연배우 사이의 연결을 표현
* movie\_temp <- unique(actor[1])[[1]]  
  actor\_comb <- data.frame()  
  for (j in 1:length(movie\_temp)) {  
   tmp <- actor[actor$movie\_id==movie\_temp[j],4]  
   if (length(tmp) > 1) {  
   actor\_comb <- rbind(actor\_comb, data.frame(movie\_id=movie\_temp[j], t(combn(tmp,2))))  
   }  
  }
* ## movie\_id X1 X2  
  ## 1 20189987 10014018 10034095  
  ## 2 20190280 10029888 10006733  
  ## 3 20190280 10029888 10050055  
  ## 4 20190280 10029888 20191807  
  ## 5 20190280 10029888 20205108  
  ## 6 20190280 10029888 20307513

## 네트워크 생성

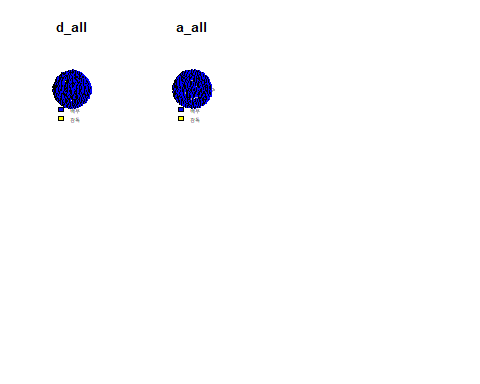
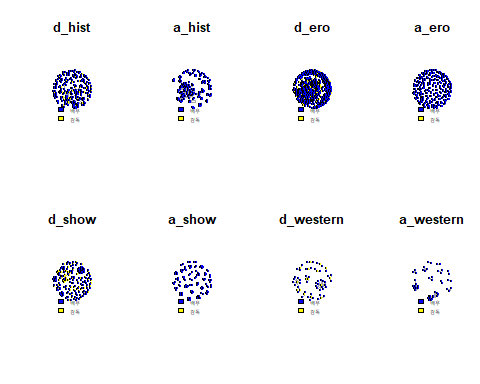
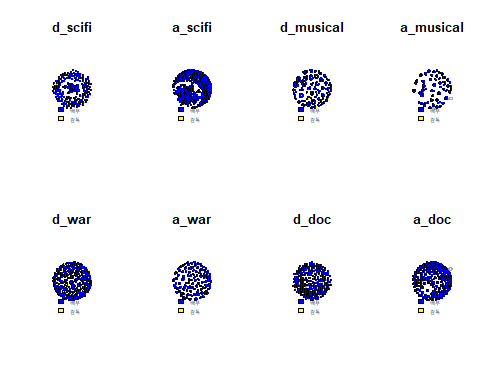
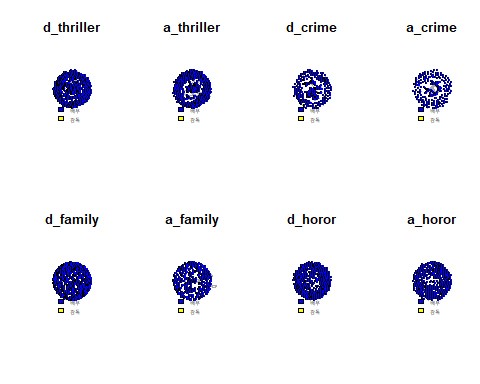
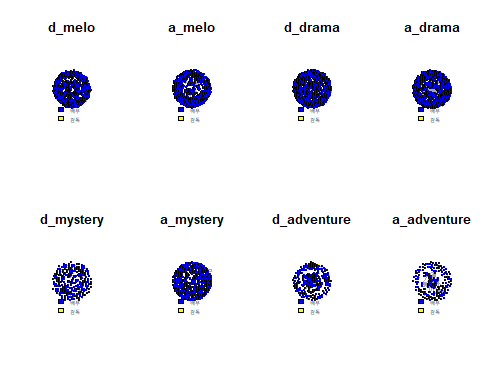
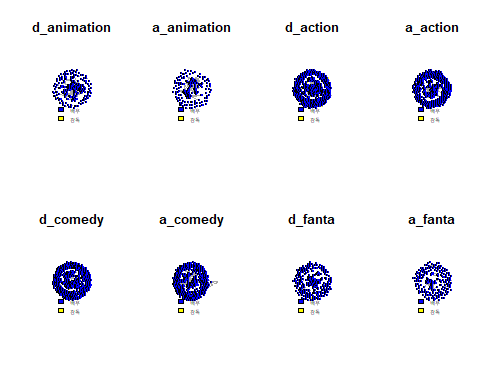
* 애니메이션, 액션, 코미디 등 20개의 장르와 통합 데이터에 대해, 각각 감독 - 배우 네트워크(이하 Director 네트워크, D네트워크)와 배우 - 배우 네트워크(이하 Actor 네트워크, A 네트워크) 생성

graph\_names <- c('d\_animation', 'a\_animation',  
 'd\_action', 'a\_action',  
 'd\_comedy', 'a\_comedy',  
 'd\_fanta', 'a\_fanta',  
 'd\_melo', 'a\_melo',  
 'd\_drama', 'a\_drama',  
 'd\_mystery', 'a\_mystery',  
 'd\_adventure', 'a\_adventure',  
 'd\_thriller', 'a\_thriller',  
 'd\_crime', 'a\_crime',  
 'd\_family', 'a\_family',  
 'd\_horor', 'a\_horor',  
 'd\_scifi', 'a\_scifi',  
 'd\_musical', 'a\_musical',  
 'd\_war', 'a\_war',  
 'd\_doc', 'a\_doc',  
 'd\_hist', 'a\_hist',  
 'd\_ero', 'a\_ero',  
 'd\_show', 'a\_show',  
 'd\_western', 'a\_western')  
  
categories <- c("애니메이션",  
 "액션",  
 "코미디",  
 "판타지",  
 "멜로/로맨스",  
 "드라마",  
 "미스터리",  
 "어드벤처",  
 "스릴러",  
 "범죄",  
 "가족",  
 "공포(호러)",  
 "SF",  
 "뮤지컬",  
 "전쟁",  
 "다큐멘터리",  
 "사극",  
 "성인물(에로)",  
 "공연",  
 "서부극(웨스턴)")  
  
# 네트워크 생성  
for (i in 1:length(categories)) {  
 staff\_temp <- director\_data[director\_data$movie\_id %in% movie[movie$genre == categories[i], ][['code']],]  
 graph\_temp <- graph\_from\_data\_frame(staff\_temp[2:3], directed=F)  
 assign(graph\_names[i\*2-1], graph\_temp)  
   
 actor\_temp <- actor\_comb[actor\_comb$movie\_id %in% movie[movie$genre == categories[i],][['code']],]  
 graph\_temp2 <- graph\_from\_data\_frame(actor\_temp[2:3], directed=F)  
 assign(graph\_names[i\*2], graph\_temp2)  
}  
  
# 전체 네트워크 생성  
staff\_temp <- director\_data  
d\_all <- graph\_from\_data\_frame(staff\_temp[2:3], directed=F)  
  
actor\_temp <- actor\_comb  
a\_all <- graph\_from\_data\_frame(actor\_temp[2:3], directed=F)  
  
graph\_names <- c(graph\_names, 'd\_all', 'a\_all')

## 시각화

* 생성된 42개의 네트워크에 대해 시각화

par(mfrow=c(2,4))  
for (s in graph\_names) {  
 g <- get(s)  
  
 gubun <- str\_sub(V(g)$name, 1, 1)  
 colors <- ifelse(gubun=='D', 'yellow', 'blue')  
 sizes <- ifelse(gubun=='D', 2, 1)  
 plot(g, main=s,  
 #layout=layout.kamada.kawai,   
 vertex.size=sizes, vertex.color=colors, edge.arrow.size=0.01, vertex.label=NA)  
 legend("bottomleft",c("배우","감독"),cex=0.7,fill=c('blue', 'yellow'),col=c('blue', 'yellow'),bty="n")  
}



## 분석

* 모든 네트워크에 대해 노드 수(vcount), 연결 수(ecount), 밀도(edge\_density), 추이성(transitivity), 3명이 서로 모두 연결된 그룹 수(triad300), 중심성(centr\_degree) 등의 통계치와 컴포넌트를 계산하여 정리
* D네트워크는 A네트워크와 다르게 2부 네트워크이므로 D네트워크와 A네트워크를 나누어 분석

graph\_stat <- data.frame()  
graph\_comp <- list()  
for (s in graph\_names) {  
 g <- get(s)  
 graph\_stat <- rbind(graph\_stat, data.frame(graph=s,   
 vcount=vcount(g),   
 ecount=ecount(g),  
 density\_a=edge\_density(g, loops=F),   
 density\_d=ecount(g)/(sum(str\_detect(names(V(g)), 'D'))\*sum(str\_detect(names(V(g)), 'A'))),  
 dyad\_mut=dyad\_census(g)$mut,   
 dyad\_null=dyad\_census(g)$null,   
 triad\_003=triad\_census(g)[1],   
 triad\_201=triad\_census(g)[11],   
 transitivity=transitivity(g),  
 centr=centr\_degree(g)$centralization))  
 graph\_comp[s] <- components(g)  
}

## graph vcount ecount density\_a density\_d dyad\_mut dyad\_null  
## 1 d\_animation 1687 2861 0.0020117555 0.0047400510 2861 1419280  
## 2 a\_animation 1153 5313 0.0079999639 Inf 5313 658815  
## 3 d\_action 3834 5073 0.0006904050 0.0016191151 5073 7342788  
## 4 a\_action 2519 7775 0.0024515824 Inf 7775 3163646  
## 5 d\_comedy 3552 4123 0.0006537620 0.0015899428 4123 6302453  
## 6 a\_comedy 2465 6283 0.0020688997 Inf 6283 3030597  
## 7 d\_fanta 1493 1697 0.0015236430 0.0039714300 1697 1112081  
## 8 a\_fanta 1085 3574 0.0060775078 Inf 3574 584496  
## 9 d\_melo 3840 4254 0.0005771360 0.0013697148 4254 7366626  
## 10 a\_melo 2381 4180 0.0014752646 Inf 4180 2829210  
## 11 d\_drama 7880 10323 0.0003325360 0.0007750946 10323 31032937  
## 12 a\_drama 5021 12614 0.0010008958 Inf 12614 12590096  
## 13 d\_mystery 1161 1034 0.0015355371 0.0038571439 1034 672346  
## 14 a\_mystery 819 1295 0.0038660063 Inf 1295 333676  
## 15 d\_adventure 1714 2206 0.0015026828 0.0038086162 2206 1465835  
## 16 a\_adventure 1222 4567 0.0061217295 Inf 4567 741464  
## 17 d\_thriller 3601 3925 0.0006055417 0.0013905895 3925 6477875  
## 18 a\_thriller 2336 4739 0.0017376287 Inf 4739 2722541  
## 19 d\_crime 1533 1534 0.0013063346 0.0031517353 1534 1172744  
## 20 a\_crime 1052 2079 0.0037606770 Inf 2079 550747  
## 21 d\_family 953 1058 0.0023323075 0.0064004840 1058 452570  
## 22 a\_family 717 1776 0.0069189594 Inf 1776 254910  
## 23 d\_horor 1951 1899 0.0009983046 0.0024074910 1899 1900326  
## 24 a\_horor 1283 1958 0.0023808279 Inf 1958 820445  
## 25 d\_scifi 1329 1466 0.0016612726 0.0042322482 1466 880990  
## 26 a\_scifi 926 2636 0.0061549238 Inf 2636 425639  
## 27 d\_musical 295 370 0.0085322264 0.0242210003 370 42995  
## 28 a\_musical 223 691 0.0279158082 Inf 691 24062  
## 29 d\_war 488 402 0.0033830410 0.0082512315 402 118426  
## 30 a\_war 327 488 0.0091555506 Inf 488 52813  
## 31 d\_doc 1354 1237 0.0013504647 0.0033529578 1237 914744  
## 32 a\_doc 881 2863 0.0073857187 Inf 2863 384777  
## 33 d\_hist 258 268 0.0080837330 0.0236916549 268 32885  
## 34 a\_hist 194 560 0.0299129320 Inf 560 18161  
## 35 d\_ero 806 903 0.0027834718 0.0066475265 903 323512  
## 36 a\_ero 267 185 0.0052096533 Inf 185 35326  
## 37 d\_show 213 197 0.0087253078 0.0187084520 197 22381  
## 38 a\_show 124 197 0.0258326777 Inf 197 7429  
## 39 d\_western 65 57 0.0274038462 0.0727040816 57 2023  
## 40 a\_western 48 103 0.0913120567 Inf 103 1025  
## 41 d\_all 16059 25746 0.0001996778 0.0004470151 25746 128911965  
## 42 a\_all 9775 35367 0.0007403535 Inf 35367 47735058  
## triad\_003 triad\_201 transitivity centr  
## 1 794788563 14141 0.0000000 0.029423594  
## 2 250478644 40610 0.4678914 0.161270869  
## 3 9368925348 19133 0.0000000 0.013136884  
## 4 2644632629 52753 0.4189558 0.035276773  
## 5 7449376514 11810 0.0000000 0.008357784  
## 6 2479416944 21595 0.5427404 0.022281750  
## 7 551302414 5481 0.0000000 0.042712282  
## 8 209186865 9480 0.7568857 0.055730610  
## 9 9416272430 11650 0.0000000 0.017656779  
## 10 2237859925 8235 0.6552374 0.016591962  
## 11 81445432670 40821 0.0000000 0.007282644  
## 12 21025953633 73681 0.3624888 0.012744124  
## 13 258987315 1876 0.0000000 0.009671359  
## 14 90204199 913 0.8034446 0.020583871  
## 15 834360627 8240 0.0000000 0.035274900  
## 16 298911238 21750 0.5989157 0.050389327  
## 17 7762627662 10228 0.0000000 0.005783347  
## 18 2111389309 17016 0.4367987 0.014536461  
## 19 597080735 3303 0.0000000 0.011748496  
## 20 191487658 4083 0.5927588 0.020026193  
## 21 142861334 3016 0.0000000 0.024978617  
## 22 60040183 2286 0.7949960 0.050343610  
## 23 1232258448 5567 0.0000000 0.016950413  
## 24 348783456 1268 0.8289261 0.013999827  
## 25 388645751 4685 0.0000000 0.038248366  
## 26 129936067 10246 0.6702816 0.078169401  
## 27 4144851 1342 0.0000000 0.100311311  
## 28 1687487 614 0.8921483 0.107219327  
## 29 19064101 632 0.0000000 0.019204228  
## 30 5625293 211 0.8781755 0.036856719  
## 31 411201019 3923 0.0000000 0.028213467  
## 32 111151851 87 0.9980311 0.036932463  
## 33 2768517 746 0.0000000 0.081410430  
## 34 1105280 1363 0.6556342 0.141071524  
## 35 86329848 4995 0.0000000 0.051874913  
## 36 3089765 102 0.3333333 0.017346738  
## 37 1548802 512 0.0000000 0.080897334  
## 38 286770 24 0.9627907 0.031077891  
## 39 40458 141 0.0000000 0.206971154  
## 40 13038 63 0.8888889 0.185283688  
## 41 689759395791 141815 0.0000000 0.004159520  
## 42 155325891826 403937 0.3247177 0.019210537

### 노드 및 연결 수

* D네트워크와 A네트워크 모두 드라마 장르에서 최대 노드수와 최대 연결수를 보였고, 서부극(웨스턴) 장르에서 최소 노드수와 최소 연결수를 보임
  + 드라마 장르에 포함된 영화가 4198개로가 가장 많고, 한 영화의 다수의 주연 배우가 출연
  + 가운데에 응집성이 큰 노드 집합이 있으며, 외곽을 구성하는 노드들 또한 조밀하게 형성되어 있음
  + 서부극 장르의 영화는 24개로 가장 적고, 눈으로 셀 수 있을 정도의 노드 수를 구성하고 있으며, 노트간의 특징은 크게 찾아보기 어려움

### 밀도

* D네트워크는 이부 네트워크이기 때문에 ecount를 가능한 최대 엣지 수로 나누는 방식으로 밀도 계산 (density\_d)
* D네트워크와 A네트워크 모두 규모가 가장 작은 서부극 장르가 가장 밀도가 높은 것으로 나타남
  + 규모가 작아 형성 가능한 엣지 수가 적기 때문에 밀도가 높게 나온 것으로 추정
* 이와 반대로, 가장 규모가 큰 드라마 장르의 밀도가 가장 낮은 것으로 나타남

### 추이성

* D네트워크는 추이성을 구할 수 없는 구조(이부 네트워크)이므로 A네트워크에 대해서만 분석 진행
* 추이성이 가장 높게 나온 A네트워크는 다큐멘터리 장르
  + 삼각형 구조가 대부분을 차지하는 것을 통해 거의 전부의 배우가 서로 함께 작품을 진행한 경험이 있는 것으로 볼 수 있음
* 장르 특성상 추이성이 낮을 수밖에 없는 에로(성인물) 장르를 제외하고, 드라마 장르의 추이성이 가장 낮음
  + 노드 수가 가장 많아 삼각형 구조 형성 자체에 어려움이 있음

### 컴포넌트

comp\_stat <- data.frame()  
for (s in graph\_names) {  
 g <- get(s)  
 comp <- graph\_comp[s]  
 top <- table(comp)[order(table(comp), decreasing = T)][1]  
 comp\_stat <- rbind(comp\_stat, data.frame(graph=s, top=top, top\_rate=top/vcount(g)))  
}

## graph top top\_rate  
## 233 d\_doc 41 0.03028065  
## 181 a\_horor 58 0.04520655  
## 44 a\_doc 40 0.04540295  
## 141 a\_war 18 0.05504587  
## 7 d\_war 28 0.05737705  
## 45 a\_show 8 0.06451613  
## 221 d\_horor 143 0.07329575  
## 91 a\_ero 24 0.08988764  
## 20 d\_mystery 114 0.09819121  
## 9 a\_mystery 85 0.10378510  
## 261 d\_musical 50 0.16949153  
## 42 a\_musical 45 0.20179372  
## 46 d\_western 14 0.21538462  
## 31 a\_melo 537 0.22553549  
## 112 d\_show 51 0.23943662  
## 6 a\_western 13 0.27083333  
## 19 a\_family 221 0.30822873  
## 25 d\_family 318 0.33368311  
## 21 a\_fanta 428 0.39447005  
## 26 d\_scifi 590 0.44394281  
## 16 a\_thriller 1060 0.45376712  
## 3 d\_fanta 687 0.46014735  
## 110 a\_scifi 439 0.47408207  
## 18 a\_crime 578 0.54942966  
## 17 d\_crime 859 0.56033920  
## 43 d\_hist 145 0.56201550  
## 13 a\_comedy 1394 0.56551724  
## 2 d\_comedy 2074 0.58389640  
## 22 d\_melo 2283 0.59453125  
## 15 d\_thriller 2168 0.60205498  
## 111 a\_hist 118 0.60824742  
## 12 a\_action 1604 0.63676062  
## 27 d\_ero 519 0.64392060  
## 23 a\_drama 3278 0.65285800  
## 113 a\_all 6522 0.66721228  
## 4 d\_action 2582 0.67344810  
## 14 a\_adventure 832 0.68085106  
## 24 d\_adventure 1188 0.69311552  
## 41 d\_drama 5581 0.70824873  
## 11 a\_animation 845 0.73287077  
## 47 d\_all 12238 0.76206489  
## 1 d\_animation 1320 0.78245406

* 네트워크 내 가장 큰 컴포넌트에 속해 있는 노드가 전체 노드 중에서 차지하는 비율 계산
* D네트워크와 A네트워크 모두 애니메이션 장르의 최대 컴포넌트 비율이 가장 높음 (78.2% / 73.2%)
  + 애니메이션 장르에 속한 감독과 배우가 가장 넓게 연결되어 있음
  + 해당 장르에서 특정 배우 또는 감독과 연락이 닿고 싶으면 70% 이상의 확률로 여러 다리를 건너 연락이 닿을 수 있음
* D네트워크에서는 다큐멘터리 장르(3.0%), A네트워크에서는 공포(4.5%)와 다큐멘터리 장르(4.5%)의 최대 컴포넌트 비율이 가장 작음
  + 다큐멘터리 장르의 경우 추이성이 가장 높았음에도 최대 컴포넌트 비율이 가장 작다는 의외의 결과가 나옴
  + 그래프를 자세히 살펴보면 그래프 전체가 대부분 4~5 노드의 작은 그룹의 조합으로 구성됨

### 장르별 가장 많은 관계자와 작업한 사람

* 장르별로 가장 많은 배우와 작업한 배우, 가장 많은 배우와 작업한 감독, 가장 많은 감독과 작업한 배우를 알아봄

result <- data.frame()  
topn <- 3  
for (j in 1:(length(graph\_names)/2)) {  
 g <- get(graph\_names[j\*2-1])  
 counts <- degree(g)[order(degree(g), decreasing = T)]  
 ordered\_names <- names(counts)  
 # D는 가장 많은 배우와 함께 일한 감독, A는 가장 많은 감독과 일한 배우  
 top\_d <- ordered\_names[str\_detect(ordered\_names, 'D')][1:topn]   
 top\_a <- ordered\_names[str\_detect(ordered\_names, 'A')][1:topn]  
 id\_d <- str\_sub(top\_d, 2)  
 id\_a <- str\_sub(top\_a, 2)  
 counts\_d <- counts[top\_d]  
 counts\_a <- counts[top\_a]  
 names\_d <- unique(staff[staff[['person\_id']] %in% id\_d,]$person\_name)  
 names\_a <- unique(staff[staff[['person\_id']] %in% id\_a,]$person\_name)  
   
 g2 <- get(graph\_names[j\*2])  
 counts2 <- degree(g2)[order(degree(g2), decreasing = T)]  
 id2 <- names(counts2)  
 top2 <- id2[1:topn]  
 names2 <- unique(staff[staff[['person\_id']] %in% top2,]$person\_name)  
   
 for (i in 1:topn) {  
 result <- rbind(result, data.frame('장르'=str\_sub(graph\_names[j\*2], 3),  
 '순위'=i,  
 '감독 with 배우'=paste0('(',counts\_d[i], ') ', names\_d[i]),   
 '배우 with 감독'=paste0('(',counts\_a[i], ') ', names\_a[i]),   
 '배우 with 배우'=paste0('(',counts2[top2[i]], ') ', names2[i])))  
 }   
}

## 장르 순위 감독.with.배우 배우.with.감독  
## 1 animation 1 (51) 최신규 (53) 엄상현  
## 2 animation 2 (39) 김현호 (30) 남도형  
## 3 animation 3 (34) 유야마 쿠니히코 (27) 이지현  
## 4 action 1 (53) 안소니 루소 (22) 사무엘 L. 잭슨  
## 5 action 2 (53) 조 루소 (21) 브루스 윌리스  
## 6 action 3 (37) 피터 잭슨 (20) 스티븐 시걸  
## 7 comedy 1 (32) 백승환 (11) 라이언 레이놀즈  
## 8 comedy 2 (25) 강효진 (10) 백승환  
## 9 comedy 3 (22) 강형철 (10) 성동일  
## 10 fanta 1 (66) 안소니 루소 (10) 이안 맥켈런  
## 11 fanta 2 (30) 조 루소 (9) 케이트 블란쳇  
## 12 fanta 3 (30) 피터 잭슨 (8) 휴고 위빙  
## 13 melo 1 (42) 백승환 (70) 박주빈  
## 14 melo 2 (32) 조성규 (40) 이은미  
## 15 melo 3 (30) 카에데 (33) 이채담  
## 16 drama 1 (60) 홍상수 (18) 유해진  
## 17 drama 2 (46) 이상우 (16) 박혁권  
## 18 drama 3 (42) 카에데 (15) 이채담  
## 19 mystery 1 (13) 구모 (6) 박성웅  
## 20 mystery 2 (10) 라스 폰 트리에 (5) 이상희  
## 21 mystery 3 (10) 팀 버튼 (4) 리암 니슨  
## 22 adventure 1 (63) 안소니 루소 (13) 사무엘 L. 잭슨  
## 23 adventure 2 (34) 조 루소 (13) 크리스 헴스워스  
## 24 adventure 3 (34) 피터 잭슨 (9) 엄상현  
## 25 thriller 1 (23) 리들리 스콧 (17) 니콜라스 케이지  
## 26 thriller 2 (19) 연상호 (13) 에단 호크  
## 27 thriller 3 (16) 김휘 (11) 브루스 윌리스  
## 28 crime 1 (20) 이해영 (9) 마동석  
## 29 crime 2 (15) 두기봉 (8) 니콜라스 케이지  
## 30 crime 3 (13) 존 힐코트 (7) 브루스 윌리스  
## 31 family 1 (26) 딜런 브라운 (10) 니콜라스 케이지  
## 32 family 2 (18) 김현호 (9) 엄상현  
## 33 family 3 (17) 김성호 (6) 남도형  
## 34 horor 1 (35) 정범식 (10) 배수빈  
## 35 horor 2 (27) 민규동 (10) 김지원  
## 36 horor 3 (24) 김곡 (10) 아담 윈가드  
## 37 scifi 1 (53) 조스 웨던 (13) 로버트 다우니 주니어  
## 38 scifi 2 (53) 안소니 루소 (12) 크리스 에반스  
## 39 scifi 3 (19) 조 루소 (11) 스칼렛 요한슨  
## 40 musical 1 (32) 백승환 (10) 백승환  
## 41 musical 2 (16) 롭 마샬 (5) 조달환  
## 42 musical 3 (16) 올 파커 (5) 강찬희  
## 43 war 1 (11) 장훈 (4) 견자단  
## 44 war 2 (10) 이재한 (3) 장쯔이  
## 45 war 3 (10) 이상우 (3) 잭 오코넬  
## 46 doc 1 (40) 임흥순 (16) 노무현  
## 47 doc 2 (19) 김재환 (8) 경순  
## 48 doc 3 (16) 이승현 (5) 최하동하  
## 49 hist 1 (23) 김주호 (6) 유해진  
## 50 hist 2 (12) 윤종빈 (5) 류승룡  
## 51 hist 3 (8) 김한민 (5) 고창석  
## 52 ero 1 (44) 카타오카 슈지 (11) 하타노 유이  
## 53 ero 2 (41) 테이 쟈가 (11) 마루 준코  
## 54 ero 3 (40) 칸바라 나오토 (11) 미즈타니 케이  
## 55 show 1 (19) 석진욱 (10) 안나 네트렙코  
## 56 show 2 (9) 정성복 (6) 플라시도 도밍고  
## 57 show 3 (8) 손석 (6) 르네 플레밍  
## 58 western 1 (15) 쿠엔틴 타란티노 (3) 맷 데이먼  
## 59 western 2 (8) 안톤 후쿠아 (2) 게리 올드만  
## 60 western 3 (4) 맷 피드몬트 (2) 덴젤 워싱턴  
## 61 all 1 (66) 조 루소 (70) 이은미  
## 62 all 2 (60) 홍상수 (53) 엄상현  
## 63 all 3 (53) 피터 잭슨 (44) 이채담  
## 배우.with.배우  
## 1 (195) 엄상현  
## 2 (121) 신용우  
## 3 (115) 이지현  
## 4 (95) 사무엘 L. 잭슨  
## 5 (92) 크리스 헴스워스  
## 6 (78) 마크 러팔로  
## 7 (60) 성동일  
## 8 (60) 이용선  
## 9 (52) 김나영  
## 10 (67) 이안 맥켈런  
## 11 (66) 케이트 블란쳇  
## 12 (63) 휴고 위빙  
## 13 (43) 김주혁  
## 14 (41) 조달환  
## 15 (36) 이채담  
## 16 (69) 유준상  
## 17 (55) 유해진  
## 18 (55) 윤여정  
## 19 (20) 서영희  
## 20 (14) 윌렘 데포  
## 21 (14) 박성웅  
## 22 (69) 사무엘 L. 잭슨  
## 23 (68) 크리스 헴스워스  
## 24 (68) 케이트 블란쳇  
## 25 (38) 제이슨 스타뎀  
## 26 (36) 브루스 윌리스  
## 27 (28) 누미 라파스  
## 28 (25) 우디 해럴슨  
## 29 (24) 류준열  
## 30 (22) 브루스 윌리스  
## 31 (41) 엄상현  
## 32 (33) 홍진욱  
## 33 (26) 남도형  
## 34 (21) 배수빈  
## 35 (20) 최윤영  
## 36 (19) 김지원  
## 37 (78) 로버트 다우니 주니어  
## 38 (75) 크리스 에반스  
## 39 (75) 스칼렛 요한슨  
## 40 (30) 백승환  
## 41 (29) 메릴 스트립  
## 42 (24) 줄리 월터스  
## 43 (15) 류승룡  
## 44 (14) 장쯔이  
## 45 (9) 잭 오코넬  
## 46 (39) 이옥선  
## 47 (39) 김복동  
## 48 (39) 이용수  
## 49 (33) 조진웅  
## 50 (26) 고창석  
## 51 (22) 강현구  
## 52 (6) 마루 준코  
## 53 (6) 야마구치 마리  
## 54 (4) 미즈타니 케이  
## 55 (7) 이성인  
## 56 (7) 소경호  
## 57 (7) 현채민  
## 58 (13) 사무엘 L. 잭슨  
## 59 (13) 커트 러셀  
## 60 (8) 덴젤 워싱턴  
## 61 (195) 사무엘 L. 잭슨  
## 62 (136) 엄상현  
## 63 (121) 신용우

## 개선할 점

* 영화의 흥행을 의미할 만한 지표가 부족하여 네트워크를 형성할 때 단순히 함께 작업했다는 사실만으로 분석 진행
* 영화의 제작년도, 제작국가에 대한 데이터의 활용이 부족했음
* 통계치로부터의 깊은 의미 유도 부족
  + 배급사에 따른 배우의 케미 등 깊이 있는 분석을 하지 못한 부분에 아쉬움이 있음