

어벤져스 오락 100점 100점  
할크 **꿀잼** 액션 서울  
할인 아이언맨 헐리웃  
영화 킹스맨 히어로 마블 감독

**Dum** **R** **어벤져스**  
에이치 오버 뮤트



# 텍스트에서 여론과 감정을 발견하기

Using R

김형준

Data Analyst : (주) 퀀트랩



# 퀀트랩 소개

- 2011년 설립
- 데이터 분석, 직무역량평가, 전문성 개발 전문 컨설팅 기업

## members



### 유재명

서울대학교 산업공학과  
서울대학교 인지과학 박사(수료)  
서울디지털대학교 상담심리학과 교수



### 황창주

서울대학교 심리학과  
서울대학교 심리학 박사(수료)  
서울대학교 심리학과 강사



### 김형준

서울대학교 인류학과 / 심리학과  
서울대학교 인지과학 석사  
前 삼성그룹 신입사원 인적성 검사 개발 연구원

## clients

- LG생활건강
- LG U+
- NC소프트
- SK플래닛
- 중소기업진흥공단
- 이지웰페어

# 워크숍 관련 온라인 사이트

<http://course.mindscale.kr/course/text-analysis>

mindscale

---

## 텍스트에서 여론과 감정을 발견하기 : R을 이용한 텍스트 데이터 분석

---

 소개

 강의

 자료

#	제목	길이
1	<a href="#">강의 소개</a>	8:13
2	<a href="#">강의 준비 하기</a>	6:22
3	<a href="#">R의 간단한 기초</a>	23:11

# 영화 자료를 이용한 다양한 분석

## Topics

- 주제(topic) 분류 - Text & Self-Rating
- 개인별 영화 추천 - Text & Self-Rating
- 감정 분석 - Text & Self-Rating

## Method

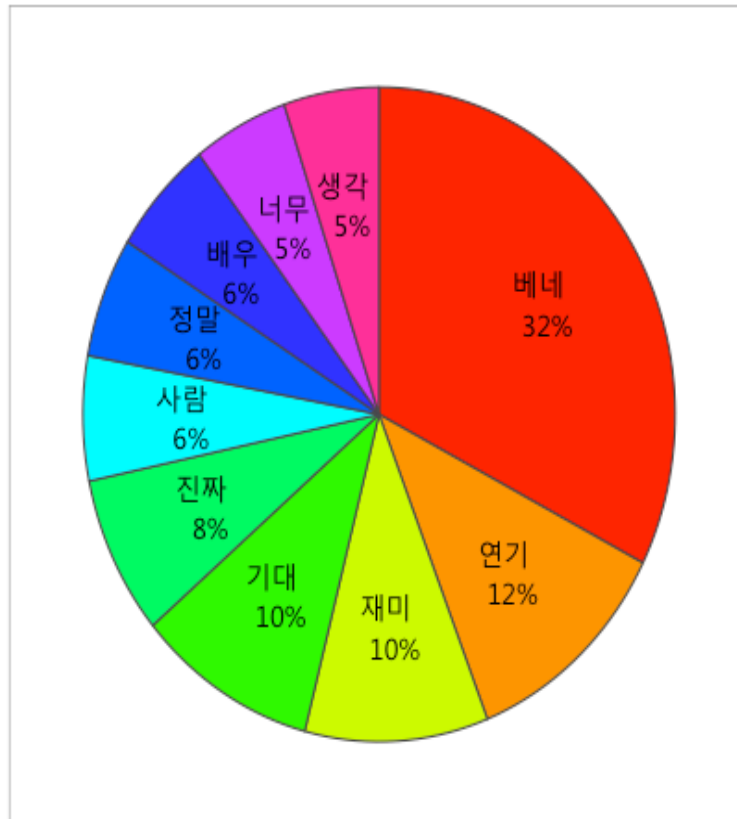
- Visualization
- Prediction

## Model

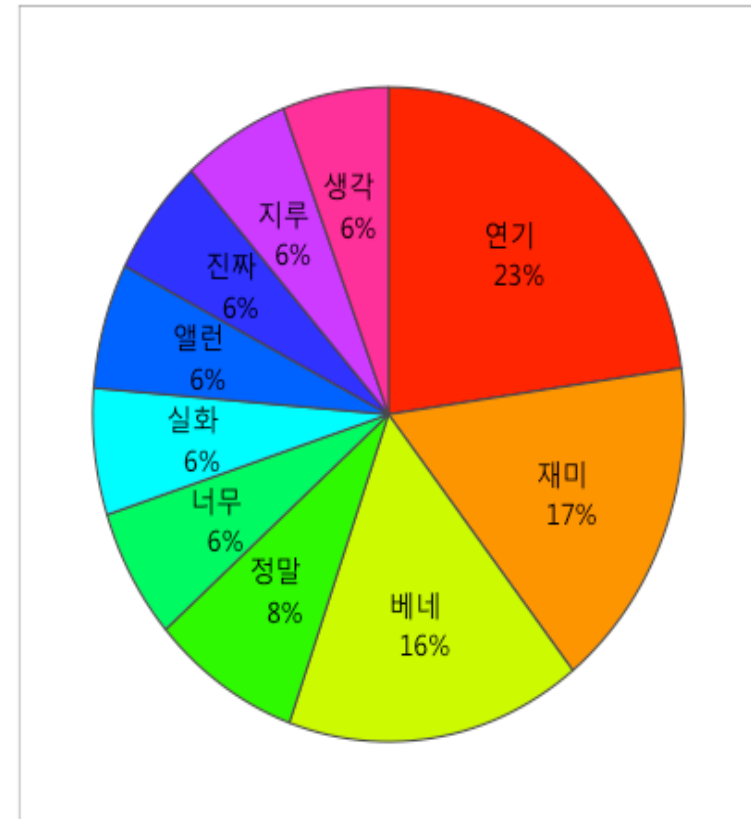
- Lasso LM / LSA / LDA / Deep Learning

# 분석 예시 - Text

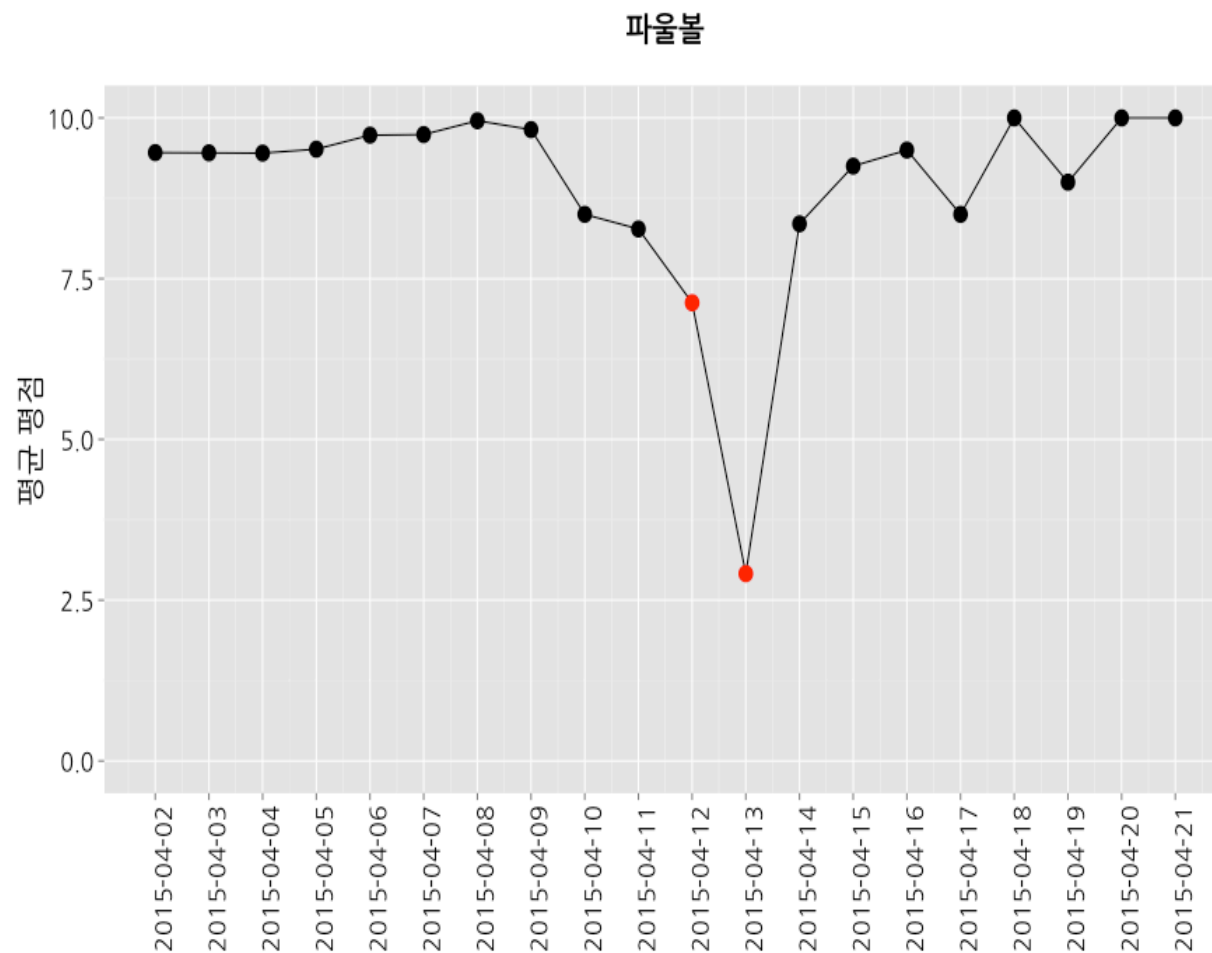
이미테이션 게임 개봉 전



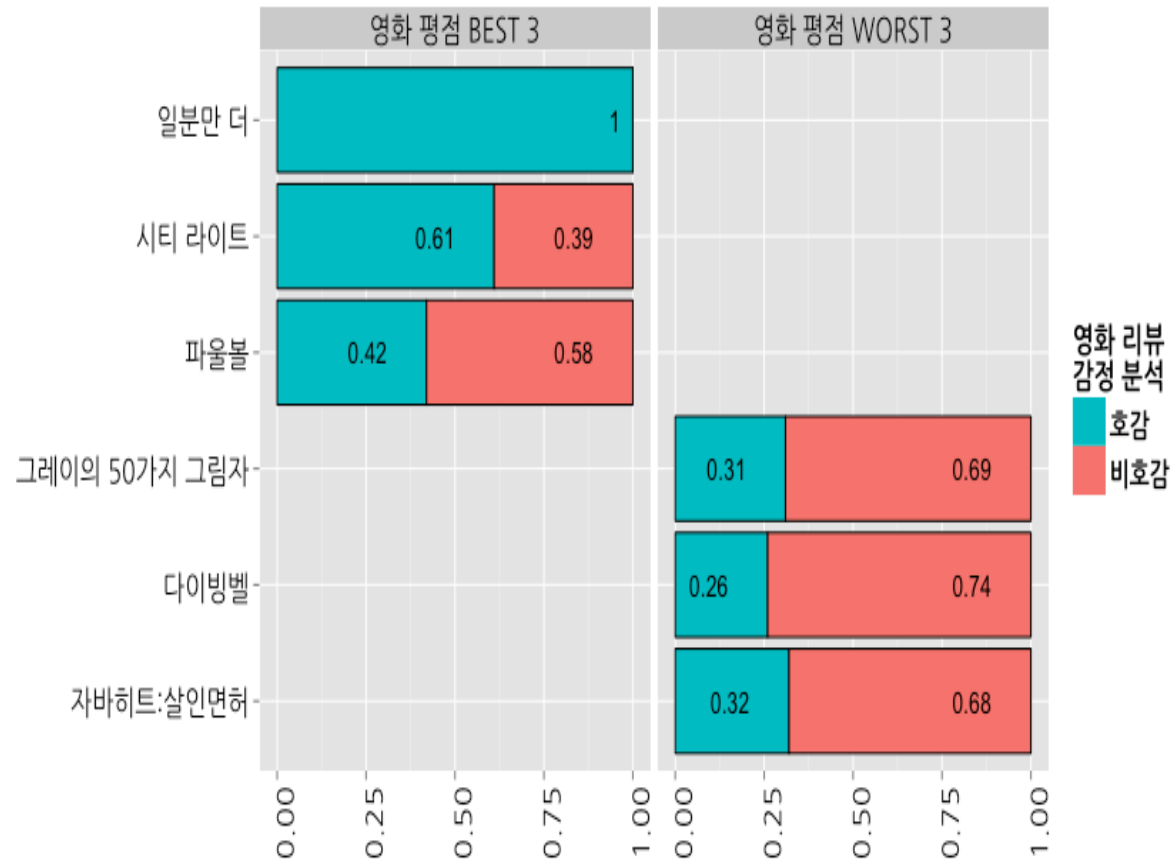
이미테이션 게임 개봉 후



# 분석 예시 - Self-Rating



# 분석 예시 - Self-Rating & Text



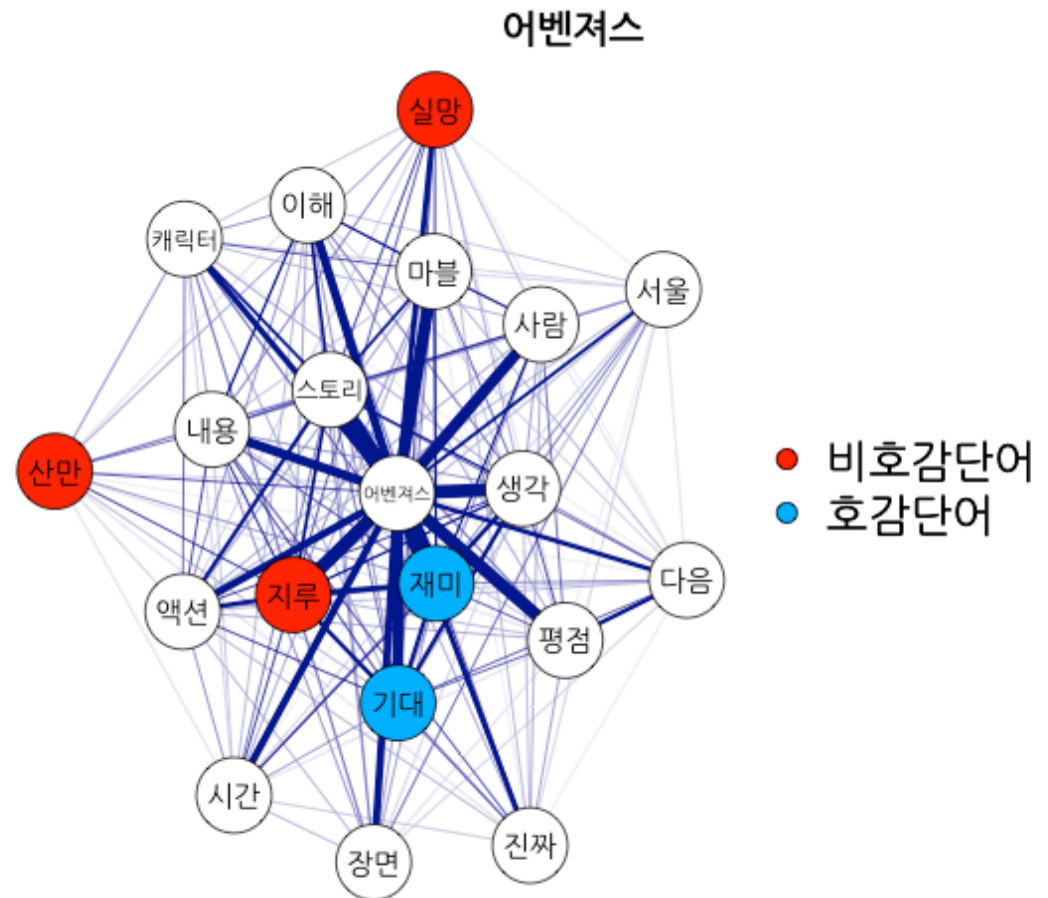


# 목차

1. R 기초 세팅
2. 패키지 인스톨
3. 어벤져스 웹크롤
4. 감정사전 불러오기
5. 키워드 파싱 및 추출
6. Co-occurrence Matrix
7. 시각화

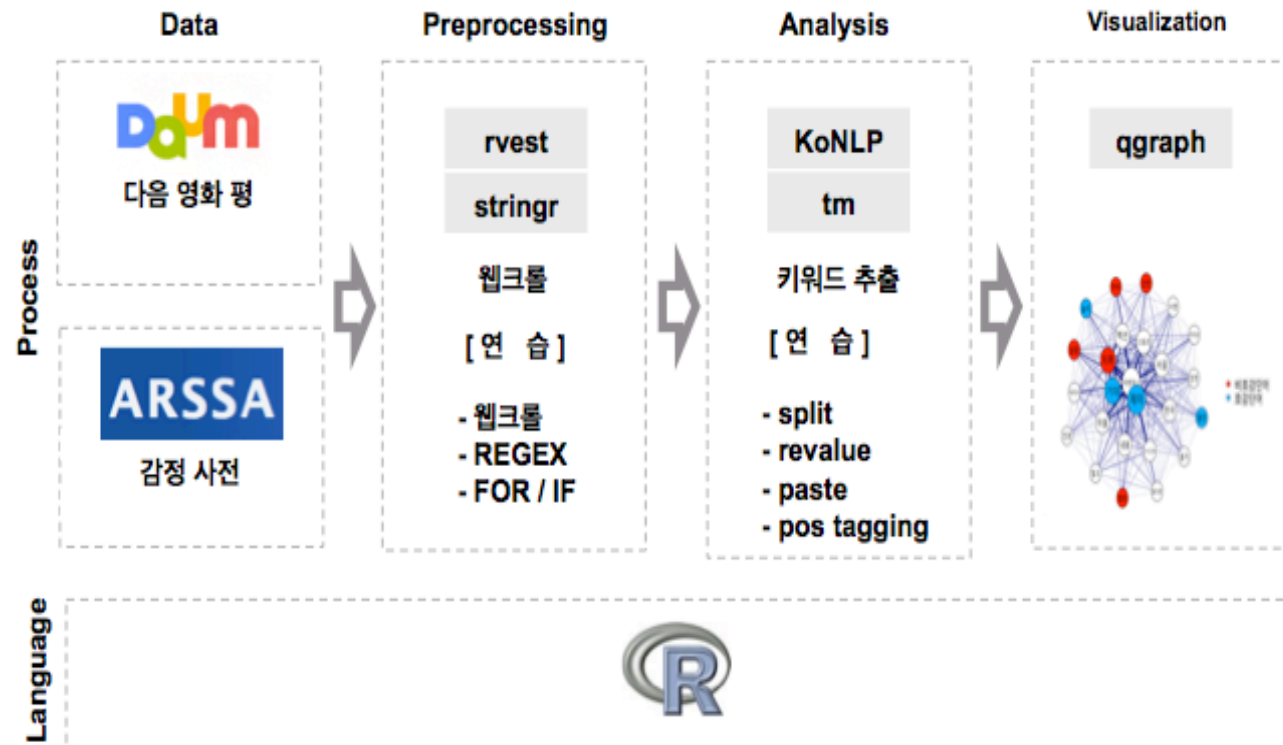
# 목차

1. R 기초 세팅
2. 패키지 인스톨
3. 어벤져스 웹크롤
4. 감정사전 불러오기
5. 키워드 파싱 및 추출
6. Co-occurrence Matrix
7. 시각화



# Work Flow

분석 과정



# 1. R 기초 세팅

# 1. R 기초 세팅

## 학습 목표

- R에서 작업 디렉토리 설정하기
- MAC의 경우 그래픽 설정하기

# 1. R 기초 세팅

- MAC

```
## 사용자 경로  
user_path = "/Users/kimhyungjun/repo/daum_movie"  
par(family="AppleGothic") ## 그림 출력시 한글폰트
```

- Windows

```
user_path = "C:/Users/kimhyungjun/repo/daum_movie"
```

- MAC & Windows 공통

```
setwd(user_path)
```

## 2. 패키지 인스톨

## 2. 패키지 인스톨

### 학습 목표

- 패키지 인스톨
- 패키지 불러오기



## 2. 패키지 인스톨

영화 웹크롤 및 파싱

- [rvest](#)
- [stringr](#)

감정 사전

- [한국어 감정 사전 1](#)
- [한국어 감정 사전 2](#)
- [참고 논문 모음 1](#)
- [참고 논문 모음 2](#)

## 2. 패키지 인스톨

텍스트 분석

- [dplyr](#)
- [tm](#)

네트워크 그래프

- [qgraph](#)

## 2. 패키지 인스톨

```
install.packages(c("rvest", "stringr", "dplyr", "tm", "qgraph", "KoNLP"),  
                 repos="http://cran.nexr.com")
```

## 패키지 불러오기

```
library("httr")  
library("rvest")  
library("stringr")  
library("plyr")  
library("tm")  
library("qgraph")  
library("KoNLP")
```

### 3. 어벤져스 웹크롤

# 3. 어벤져스 웹크롤

## 학습 목표

- 웹사이트 배경지식
- 단일 페이지 웹크롤
- 복수 페이지 웹크롤

### 3. 웹사이트 배경지식

- 웹브라우저 열기(e.g. Firefox, Chrome, Safari, Internet Explorer)
- 웹사이트 입력(e.g. <http://movie.daum.net>)
- 사용자는 client로 페이지, 이미지, 텍스트를 웹 서버로 요청함
- 웹 서버 사용자에게 반응을 보냄
- 사용자와 웹 서버는 프로토콜(e.g. HTTP)로 커뮤니케이션

# HTTP

## HTTP?

- HyperText Transfer Protocol

```
library("httr")
GET("http://google.com/") ## Request -> Response
```

```
## Response [http://www.google.co.kr/?gfe_rd=cr&ei=gP5SVfymLurM8gf-soCABw]
##   Date: 2015-05-13 16:34
##   Status: 200
##   Content-Type: text/html; charset=EUC-KR
##   Size: 18.8 kB
## <!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" 1...
## window.google.vel.lu&&window.google.vel.lu(a),d.src=a,google.li=g+1});go...
## function _gjh(){!_gjuc()&&window.google&&google.x&&google.x({id:"GJH"},f...
## if (!iesg){document.f&&document.f.q.focus();document.gbqf&&document.gbqf...
## }
## })();</script><div id="mngb">    <div id=gbar><noabr><b class=gb1>검색</b> ...
## });})();</script> </div> </span><br clear="all" id="lgpd"><div id="lga">...
```

## 다음 영화

<http://movie.daum.net>

리뷰

Review

네티즌 리뷰

미디어 리뷰

네티즌 평점

전문가 20자평

리뷰 골라보기

어벤져스 : 에이지 오브 울트론

차이나라운

위험한 상견례 2

연애의 맛

다이노 타임

노아의 방주 : 남겨진 녀석들

언프렌디드: 친구삭제

기생수 파트2

장수상회

스틸 엘리츠

엑시덴탈 러브

분노의 질주: 더 세븐

비비안 마이어를 찾아서

짱구는 못말려 극장판: 장대한승리! 로봇이빠의 역습

말할 수 없는 비밀

위플래쉬

네티즌 평점

전체목록

어벤져스 : 에이지 오브 울트론 (2015)

The Avengers: Age of Ultron

평점

★★★★☆ 6.3

평점주기

감독

조스 웨딘

출연

로버트 다우니 주니어, 크리스 헴스워스, 마크 러팔로

상세보기

리뷰보기

예매하기

현재 상영작 평점보기

개봉 예정작 평점보기

리뷰

평점

열혈회원 되는법

관람후 (1747)

관람전 (238)

전체 평점보기

평점선택

로그인 후에 등록됩니다.

등록

0 / 150자 | 평점 운영원칙

어벤져스 : 에이지 오브 울트론

★★★★☆ 1.0

인생 최악의 영화..내용이 없고 남는게 없는 영화일거라고 생각은 했지만, 재미까지 없을줄 몰랐다. 정말 시간 낭비 돈 낭비..

YY

2015.05.12

추천

어벤져스 : 에이지 오브 울트론

★★★★☆ 2.0

애들 만화 영화 수준! 한국 수준하고는!

네이념

2015.05.12

추천

어벤져스 : 에이지 오브 울트론

★★★★☆ 4.0

원 내용인지 글구 서울이 서울같이 안나온 글구 찰문제는 재미가없다

f9f99

2015.05.12

추천



어벤져스 : 에이지 오브 울트론 x
HyungJun

---

movie.daum.net/moviedetail/moviedetailNetizenPoint.do?moviel=...
☆ ⓘ ≡

---

Links Microsoft Websites neuro parkinson 2013 Autumn Course >> 기타 북마크
로고인

---

## 영화 만화 음악 연예 유아영

**영화**

전체 ▾

---

영화를 영화정보 다운로드 리뷰 매거진 스타존 포토존 이벤트 예매

### 고화질로 감상하는 HD 예고편

**어벤져스 : 에이지 오브 울트론 (2015)** 상영중

The Avengers: Age of Ultron

액션, 어드벤처, SF | 미국 | 141 분 | 2015-04-23

네티즌 ★★

메인	줄거리	출연	포토	동영상	리뷰	평점	명대사	관련영화
----	-----	----	----	-----	----	----	-----	------

이

관람후 (1785) | 관람전 (238)
[업로드된 된!](#)

네티즌 (1785명) 전체보기

★★★★☆ **6.2**

업로드원 (121명) 모아보기

★★★★☆ **6.9**

전문가 (7명) 상세보기

★★★☆☆ **6.9**

로그인 후에 등록됩니다.

평점선택 ▾ 등록

/ 150자 | 평점 운영정책
[내가 쓴 평점](#)

★★★★☆ **2.0** 기대가 너무 컸었나봐요..... ☹️

hee 2015.05.15

0 추천

★★★★☆ **0.0** 개쓰레기영화 영화 끝나기 기다리면서 디잔돌알았네~ 쓰레기영화가 상영시간은 또 길나게걸어요 ☹️

블랙역타인 2015.05.15

0 추천

★★★★☆ **2.0** 그래픽 복역영화수준.. ☹️

사랑파기름 2015.05.15

0 추천

★★★★☆ **9.0** 히어로물은 언제나 관심이 가. 인간 능력 밖의 놀라운 세계를 영상으로 만나다 는건 지금을 사는 우리가 누릴 수 있는 선물중에 하나 아닐까? 그러므로 평점은 좀 높게 줬어. 빠른 좀 잊고 빠져들어보는것도 좋겠지 ☺️

황곡기는진다 2015.05.15

0 추천

```

<div id="movieNetizenPointList" class="commentList">
  <ul>

    <li>
      <span class="starWrap">
        <span class="star_mid">
          <span class="star_mid pink">
            <span class="pt1">
              <span class="pt2" style="width:20.0%"> 네티즌별점 </span>
            </span>
            <em>2.0</em>
          </span>
        </span>
      </span>

      <span class="comment article">

                <a
                  href="http://movie.daum.net/moviedetail/moviedetailNetizenPointComment.do?moviel=73750&ratingId=1812275&type=after" title="댓글달기">기대가 너무 컸었나봐요.....
                   </a>

                </span>

      <span class="authorWrap">
        <span class="fs11 author "> <a href="/review/netizen_point/movieNetizenPoint.do?type=after&page=1&op=userId&searchValue=EFSHG-zlkY0" title="hee님이 작성한 평점 모두보기" class="id">hee</a>
      </span>

        <span class="date">
          <span class="datext">2015.05.15</span>
          <a class="reportBtn" href="#" onclick="new ULpopUp('http://spam119.daum.net/form?platformCode=MOVIE&contentType=COMMENT&siteld=MovieDetail&commentId=1812275','popupReport',461,280,0,0,1);">
            contents/movie/2008_home/common/bell_on.gif" width="10" height="10" class="oops" alt="신고하기" title="신고하기">
          </a>
        </span>
      </span>
      <span>
        <span class="reco"><button onkeypress="this.onclick;" onclick=" ULcheckLogin(); return false;"
        <span class="count" id="point1812275">0</span> <span class="bt">추천</span></button></span>
      </span>
    </li>
  </ul>

```

# 요소점검

어벤저스 - Daum 영화

movie.daum.net/moviedetail/moviedetailNetizenPoint.do?movieId=54081&searchType=all&type=after&page=1


관람후 (1900) | 관람전 (151)


네티즌 (1900명) 전체보기 일화회원 (215명) 모아보기 전문가 (7명) 상세보기

★★★★☆ 8.0 ★★★★★ 7.7 ★★★★★ 7.7


평점선택 로그인 후에 등록됩니다. 등록

0 / 150자 | 평점 운영원칙 내가 쓴 평점

★★★★☆ 8.0 유머와 슈퍼 히어로 캐릭터가 잘 버무려진 영화  돌돌이 2015.05.14 0 추천

★★★★☆ 9.0 가히 마블 최고의 블록버스터  김태환 0 추천

클릭하면 이전 페이지로 가고 누르고 있으면 방문 기록이 나타납니다.

★★★★☆ 9.0 마블 시리즈 참나요 토르도 볼만했구요 헐크가 잘 찍혔다  아낙수나 2015.05.05 0 추천

Elements Network Sources Timeline Profiles Resources Audits Console

```

<a name="form"></a>
<div id="movieNetizenPointForm">...</div>
<div id="movieNetizenPointList" class="commentList">
  <ul>
    <li>
      <span class="starWrap">...</span>
      <span class="comment article">
        <a href="http://movie.daum.net/moviedetail/moviedetailNetizenPointComment.do?movieId=54081&ratingId=1811834&type=after" title="댓글달기">
          "유머와 슈퍼 히어로 캐릭터가 잘 버무려진 영화"
          
        </a>
      </span>
      <span class="authorWrap">...</span>
      <span class="rec">...</span>
    </li>
  </ul>
</div>

```

html body #Daum\_doc #Daum\_bd #t\_m div #movieNetizenPoint div#movieNetizenPointList.commentList ul li span.comment.article a

# 어벤져스 (1 페이지 웹크롤)

```
urls_view <- "http://movie.daum.net/moviedetail/moviedetailNetizenPoint.do?movieId=73750&searchType=all&type=after&page="
r <- GET(urls_view)
htxt <- html(r)
```

```
library("rvest")
```

```
movie_text <- html_nodes(htxt, ".comment")
movie_text
```

[[1]]

```
<a href="http://movie.daum.net/moviedetail/moviedetailNetizenPointComment.do"
</span>
```

```
<a href="http://movie.daum.net/moviedetail/moviedetailNetizenPointComment.do"
</span>
```

[[2]]

```
<a href="http://movie.daum.net/moviedetail/moviedetailNetizenPointComment.do"
```

왼쪽 주먹으로 콩콩콩 때리면서 기절해~기절해~기절해~ 하면서 때리는 장면....ㅋㅋㅋ 

[[3]]

```
<a href="http://movie.daum.net/moviedetail/moviedetailNetizenPointComment.do"
```

헐크가 더 싸... 역시.. 

28/119

```
movie_text <- html_nodes(movie_text, "a")
movie_text
```

[[1]] [정말 지루한 영화~ 비추](#)

[[2]] [어벤져스2 에서 가장 웃겼던 장면은 아이언맨이 헐크버스터를 입고 헐크랑 싸울때 헐크 뉘혀놓고 왼쪽 주먹으로 콩콩콩 때리면서 기절해~기절해~기절해~ 하면서 때리는 장면....ㅋㅋㅋ](#)

[[3]] [ㅋㅋ 난 아이언맨이 가장 강할줄 알았는데.. 완전 반전!! 헐크가 더 세... 역시..](#)

[[4]] [정말 수준 낮고 내용없고 극장의 횡포를 보여준 영화다 다른 영화를 고를 기회를 주지 않는이런 대기업은 망해야한다](#)

[[5]] [사람들 하품하고 옆자리 초딩도 지루하다고..ㅋㅋㅋ 잡다하게 영웅들이 몽땅들 출연해 잔챙이 로봇들과 계속 싸워요 그와중에 느린 활쏘기ㅋ.. 엔딩역시 지리멸렬 무슨 자긍심인지 캐릭터 스스로들 셀프 감동ㅋ 스케일만 크면 좋은 영화인가ㅋ 내용도 하나도 없다ㅋ](#)

[[6]] [진짜 극장 독점 그만좀해라..... 다른영화는 시간대 오전이나 밤이라 볼수가 없네 쓰벌... 관객이 만아서 스크린 독점한게 아니라 다른거 못보게 스크린 다 차지하니까 울며겨자먹기로 이것만 볼수밖에.....; 나도 히어로물 왕팬인데 이건아니지.....; 베트맨이나 스파이더맨 발도 못따라가](#)

[[7]] [그냥 가벼운 소재의 오락물이라고 봄](#)

[[8]] [어벤져스를 잘 모르면, 엄청 재미없을 듯~ 알아도 그저 그런](#)

[[9]] [수현은 극중 비중은 꽤 중요한 역이나 등장씬이 너무 적음](#)

[[10]] [기대가 너무 컸었나봐요.....](#)

[[11]] [개쓰레기영화 영화끝나기 기다리면서 디진줄알왕네~ 쓰레기영화가 상영시간은 또 겁나게길어요](#)

```
movie_text <- html_text(movie_text)
movie_text
```

- [1] "정말 지루한 영화~ 비추 "
- [2] "어벤져스2 에서 가장 웃겼던 장면은 아이언맨이 헐크버스터를 입고 헐크랑 싸울때 헐크 높혀놓고\r\n왼쪽 주먹으로 쿵쿵쿵 때리면서 기절해~기절해~기절해~ 하면서 때리는 장면....ㅋㅋㅋ "
- [3] "ㅋㅋ 난 아이언맨이 가장 강할줄 알았는데.. 완전 반전!!\r\n헐크가 더 세...\r\n역시.. "
- [4] "정말 수준 낮고 내용없고 극장의 횡포를 보여준 영화다 다른 영화를 고를 기회를 주지 않는이런 대기업은 망해야한다 "
- [5] "사람들 하품하고 옆자리 초딩도 지루하다고..ㅋㅋㅋ 잡다하게 영웅들이 몽땅들 출연해 잔챙이 로봇들과 계속 싸워요 그와중에 느린 활쏘기ㅋ.. 엔딩역시 지리멸렬 무슨 자긍심인지 캐릭터 스스로들 셀프 감동ㅋ 스케일만 크면 좋은 영화인가ㅋ 내용도 하나도 없다ㅋ "
- [6] "진짜 극장 독점 그만좀해라..... 다른영화는 시간대 오전이나 밤이라 볼수가 없네 쓰벌...\r\n관객이 만아서 스크린 독점한게 아니라 다른거 못보게 스크린 다 차지하니까 울며겨자먹기로\r\n이것만 볼수밖에;;;;;;;;;;;;;; 나 도 히어로물 왕팬인데 이건아니지;;;;;;;;;;;;;\r\n베트맨이나 스파이더맨 발도 못따라가 " [7] "그냥 가벼운 소재의 오락물이라고 봄 "
- [8] "어벤져스를 잘 모르면, 엄청 재미없을 듯~ 알아도 그저 그런 "
- [9] "수현은 극중 비중은 꽤 중요한 역이나 등장씬이 너무 적음 "
- [10] "기대가 너무 컸었나봐요..... "
- [11] "개쓰레기영화 영화끝나기 기다리면서 디진줄알왕네~\n쓰레기영화가 상영시간은 또 겁나게길어요 "

# 어벤져스 (1 페이지 웹크롤)

```
urls_view <- "http://movie.daum.net/moviedetail/moviedetailNetizenPoint.do?movieId=73750&search"
r <- GET(urls_view)
htxt <- html(r)
movie_text <- html_nodes(htxt, ".comment")
movie_text <- html_nodes(movie_text, "a")
movie_text <- html_text(movie_text)
length(html_nodes(htxt, ".comment"))
```

```
## [1] 15
```

**R 연습 - FOR & IF, paste**



# FOR & IF, break (연습 1)

```
for (i in 1:5)
{
    print(i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
```

```
for (i in 1:100)
{
    if(i==3) break
    print(i)
}
```

```
## [1] 1
## [1] 2
```

## c, paste (연습 2)

```
ex1 <- c("어벤져스 재밌다")  
ex2 <- c("줄리다")  
ex_sum <- c(ex1, ex2)  
ex_sum
```

```
## [1] "어벤져스 재밌다" "줄리다"
```

## c, paste (연습 2)

```
paste("page=", 1)
```

```
## [1] "page= 1"
```

```
paste("page=", 1, sep="")
```

```
## [1] "page=1"
```

```
page_num = 1  
paste("page=", page_num, sep="")
```

```
## [1] "page=1"
```

```
page_num = 2  
paste("page=", page_num, sep="")
```

```
## [1] "page=2"
```

## c, paste (연습 2)

```
ex <- c("어벤져스", ex2)  
paste(ex, collapse="")
```

```
## [1] "어벤져스졸리다"
```

```
paste(ex, collapse=" ")
```

```
## [1] "어벤져스 졸리다"
```

```
paste(ex, collapse=" + ")
```

```
## [1] "어벤져스 + 졸리다"
```

```
urls_view <- "http://movie.daum.net/moviedetail/moviedetailNetizenPoint.do?movieId=73750&search"
r <- GET(urls_view)
htxt <- html(r)
length(html_nodes(htxt, ".comment"))
```

```
## [1] 0
```

# 어벤져스 전체 페이지 웹크롤

```
movie_text_sum <- c()

for (page_num in 1:1000)
{
  urls_view <-
  paste("http://movie.daum.net/moviedetail/moviedetailNetizenPoint.do?movieId=73750&searchType=all&type=after&page=",
  page_num, sep="")
  r <- GET(urls_view)
  htxt <- html(r)

  movie_text <- html_nodes(htxt, ".comment")
  movie_text <- html_nodes(movie_text, "a")
  movie_text <- html_text(movie_text)

  if(length(movie_text)==0) break;

  movie_text_sum <- c(movie_text_sum, movie_text)
  print(paste(page_num, "-th page", sep=""))
}
```

## 4. 감정사전 불러오기

## 4. 감정사전 불러오기

### 학습 목표

- 파일 불러오기(read.csv)
- 긍정 사전과 부정 사전으로 분할(subset)



## 4. 감정사전 불러오기

```
emotion_dict <- read.csv("emotion_dict.csv",  
                        header = T,  
                        fileEncoding = "UTF-8",  
                        stringsAsFactors = F)
```

```
pos_word <- subset(emotion_dict, pos_neg=="pos")[,"words"]  
neg_word <- subset(emotion_dict, pos_neg=="neg")[,"words"]  
#emotion_dict[11:15,]; emotion_dict[1301:1305,];
```

감정사전

## 5. 키워드 파싱 및 추출

# 5. 키워드 파싱 및 추출

## 학습 목표

- R에서 자연어 처리 문제
- 키워드 추출 방법 I (KoNLP - ExtracNoun)
- 키워드 추출 방법 II (KoNLP - POStagging) - APPENDIX II
- 키워드 추출 방법 III (앞 두 글자 자르기) - APPENDIX III

# R에서 자연어 처리 (KoNLP)

기대했던 것보다 좀 지루했음... 와이프는 재미있다고...

```
library("KoNLP")  
extractNoun("기대했던 것보다 좀 지루했음... 와이프는 재미있다고...")
```

```
## [1] "것"      "지루"    "와이프"
```

```
split_12("기대했던 것보다 좀 지루했음... 와이프는 재미있다고...")
```

```
## [1] "기대" "것보" "좀"    "지루" "와이" "재미"
```

```
extractNounVerbAdj("기대했던 것보다 좀 지루했음... 와이프는 재미있다고...")
```

```
## [[1]]  
## [1] ""      "기대"  "하"    "것"    "지루"  "와이프" "재미있" ""
```

**R 연습 - extractNoun, nchar, revalue**

## extractNoun (연습 3)

```
ex <- "어 헐크 대박이네 ㅋ 잼슴"  
ex <- extractNoun(ex)  
ex
```

```
## [1] "어"    "헐크"  "대박"  "ㅋ"     "잼슴"
```

# nchar로 1글자 제거 (연습 4)

```
ex
```

```
## [1] "어" "헐크" "대박" "ㅋ" "잼슴"
```

```
nchar(ex)
```

```
## [1] 1 2 2 1 2
```

```
ex <- ex[nchar(ex) > 1]  
ex
```

```
## [1] "헐크" "대박" "잼슴"
```

# revalue로 맞춤법 교정 (연습 5)

```
ex
```

```
## [1] "헐크" "대박" "잼슴"
```

```
library("plyr")
```

```
revalue(ex, c("잼슴" = "재미"))
```

```
## [1] "헐크" "대박" "재미"
```

```
revalue(ex, c("대박" = "완전", "잼슴" = "재미"))
```

```
## [1] "헐크" "완전" "재미"
```



# 키워드 추출 (어벤져스)

- Step (1) extractNoun

```
movie_text_sum[1]
```

```
## [1] "괜찮음.. 시원한 액션.. 간적으로 캡틴아메리카 너무 좋아함. 헐크도 좋고.. ♥♥♥"
```

```
key_vec <- extractNoun(movie_text_sum[1])  
key_vec
```

```
## [1] "시원"      "한"        "액션"      "적"  
## [5] "캡틴아메리카" "헐크도"    "♥♥♥"
```

# 키워드 추출 (어벤져스)

- Step (2) 한 글자 제거

```
key_vec <- key_vec[nchar(key_vec) > 1]  
key_vec
```

```
## [1] "시원"      "액션"      "캡틴아메리카" "헐크도"  
## [5] "❤️❤️❤️"
```

# 키워드 추출 (어벤져스)

- Step (3) 맞춤법 교정

```
movie_name <- "어벤져스"
key_vec <- revalue(key_vec, c("재밋" = "재미",
                              "재밋" = "재미",
                              "잼있" = "재미",
                              "영화" = movie_name),
                  warn_missing = F)

key_vec
```

```
## [1] "시원"          "액션"          "캡틴아메리카" "헐크도"
## [5] "♥♥♥♥"
```

# 키워드 추출 (어벤져스)

- Step (4) 다시 한 문장으로 합치기

```
key_vec <- paste(key_vec, collapse=" ")  
key_vec
```

```
## [1] "시원 액션 캡틴아메리카 헐크도 ❤️❤️❤️"
```

# 키워드 추출 (어벤져스)

```
key_vec_sum <- c();
movie_name = "어벤져스"

for (i in 1:length(movie_text_sum))
{
key_vec <- extractNoun(movie_text_sum[i])

key_vec <- revalue(key_vec, c("재밋" = "재미",
                             "재밋" = "재미",
                             "잼있" = "재미",
                             "영화" = movie_name),
                    warn_missing = F)

key_vec <- key_vec[nchar(key_vec) > 1]
key_vec <- c(key_vec, ' ') ## 윈도우 tm 버그 때문
key_vec_sum[i] <- paste(key_vec, collapse=' ') ## 두 칸 (윈도우 tm 버그 때문)
}
```

```
## Warning: It's not kind of right sentence : '재미겁나.없음.스토리자체가어거지로맞츠는데, '
## Warning: It's not kind of right sentence : '꾸벅꾸벅.....억지로봤네요.....'
## Warning: It's not kind of right sentence : '저는개인적으로1편보단재밋게봤습니다솔직히첨부분은지루한점은있었는데중반부터재밋더라고요역시CG가디
## Warning: It's not kind of right sentence : '재미없다는사람들대체이때까지어떤영화를본거지....핵꿀잼이던데...'
## Warning: It's not kind of right sentence : '그냥자다가다시일어나서보고ㅠ별로모르겠음ㅜㅜ'
```

## **6. Co-occurrence Matrix**

## 6. Co-occurrence Matrix

### 학습 목표

- Term x Document Matrix
- Co-occurrence Matrix

# Term x Document Matrix

- 행(row)은 Term(단어들), 열(col)은 Document(개인들)로 이루어진 Matrix
- 단어에 대하여 Weight
- 문서 내 단어에 대하여 Weight
- 모형에 따라 다양한 방식으로 처리

```
library("tm")
```

```
key_corpus <- Corpus(DataframeSource(as.data.frame(key_vec_sum)))  
key_corpus
```

```
## <<VCorpus (documents: 1737, metadata (corpus/indexed): 0/0)>>
```



# Term x Document Matrix

```
key_tdm <- TermDocumentMatrix(key_corpus)
key_tdm
```

```
## <<TermDocumentMatrix (terms: 3442, documents: 1737)>>
## Non-/sparse entries: 5632/5973122
## Sparsity           : 100%
## Maximal term length: 152
## Weighting          : term frequency (tf)
```

# Term x Document Matrix

```
rownames(key_tdm)[rownames(key_tdm)=="어벤져스"]
```

```
## [1] "어벤져스"
```

```
rownames(key_tdm)[rownames(key_tdm)=="헐크"]
```

```
## character(0)
```

```
rownames(key_tdm)[rownames(key_tdm)=="등등"]
```

```
## character(0)
```

```
rownames(key_tdm)[rownames(key_tdm)=="-.—"]
```

```
## [1] "-.—"
```

# Term x Document Matrix

```
key_tdm <- TermDocumentMatrix(key_corpus,  
                                control = list(  
                                  removeNumbers = TRUE,  
                                  removePunctuation = TRUE))
```

# Term x Document Matrix

```
rownames(key_tdm)[rownames(key_tdm)=="어벤져스"]
```

```
## [1] "어벤져스"
```

```
rownames(key_tdm)[rownames(key_tdm)=="헐크"]
```

```
## character(0)
```

```
rownames(key_tdm)[rownames(key_tdm)=="등등"]
```

```
## character(0)
```

```
rownames(key_tdm)[rownames(key_tdm)=="-.—"]
```

```
## character(0)
```

# Term x Document Matrix

?TermDocumentMatrix

?TermFreq

# Term x Document Matrix

```
key_tdm <- TermDocumentMatrix(key_corpus,  
                                control = list(  
                                  removeNumbers = TRUE,  
                                  removePunctuation = TRUE,  
                                  wordLengths = c(2, Inf)))
```

# Term x Document Matrix

```
rownames(key_tdm)[rownames(key_tdm)=="어벤져스"]
```

```
## [1] "어벤져스"
```

```
rownames(key_tdm)[rownames(key_tdm)=="헐크"]
```

```
## [1] "헐크"
```

```
rownames(key_tdm)[rownames(key_tdm)=="등등"]
```

```
## [1] "등등"
```

```
rownames(key_tdm)[rownames(key_tdm)=="-.—"]
```

```
## character(0)
```

# 불필요 단어제거

```
stopwords()
```

```
## [1] "i"      "me"      "my"      "myself"  "we"
## [6] "our"    "ours"    "ourselves" "you"     "your"
## [11] "yours"  "yourself" "yourselves" "he"      "him"
## [16] "his"    "himself" "she"       "her"     "hers"
## [21] "herself" "it"      "its"       "itself"  "they"
## [26] "them"   "their"   "theirs"    "themselves" "what"
## [31] "which"  "who"     "whom"     "this"     "that"
## [36] "these"  "those"   "am"       "is"       "are"
## [41] "was"    "were"    "be"       "been"     "being"
## [46] "have"   "has"     "had"      "having"   "do"
## [51] "does"   "did"     "doing"    "would"    "should"
## [56] "could"  "ought"   "i'm"      "you're"   "he's"
## [61] "she's"  "it's"    "we're"    "they're"  "i've"
## [66] "you've" "we've"   "they've"  "i'd"      "you'd"
## [71] "he'd"   "she'd"   "we'd"     "they'd"   "i'll"
## [76] "you'll" "he'll"   "she'll"   "we'll"    "they'll"
## [81] "isn't"  "aren't"  "wasn't"   "weren't"  "hasn't"
```



# Term x Document Matrix

- 해석이 힘든 단어들을 Term x Document Matrix 생성 시 제거

```
delete_dic <- c("그냥", "등등", "중간")
```

# Term x Document Matrix

```
key_tdm <- TermDocumentMatrix(key_corpus,  
                               control = list(  
                                 removeNumbers = TRUE,  
                                 removePunctuation = TRUE,  
                                 wordLengths = c(2, Inf),  
                                 stopwords = delete_dic))
```

# Term x Document Matrix

```
rownames(key_tdm)[rownames(key_tdm)=="어벤져스"]
```

```
## [1] "어벤져스"
```

```
rownames(key_tdm)[rownames(key_tdm)=="헐크"]
```

```
## [1] "헐크"
```

```
rownames(key_tdm)[rownames(key_tdm)=="등등"]
```

```
## character(0)
```

```
rownames(key_tdm)[rownames(key_tdm)=="-.—"]
```

```
## character(0)
```

# Term x Document Matrix

```
key_tdm_m <- as.matrix(key_tdm)
rownames(key_tdm_m) <- str_trim(rownames(key_tdm_m)) ## for windows
dim(key_tdm)
```

```
## [1] 4940 1737
```

# Term x Document Matrix

- 행(row)은 Term(단어들), 열(col)은 Document(개인들)로 이루어진 Matrix

```
ex <- matrix(c(1,1,1,0,  
              1,0,1,0,  
              0,1,0,1),  
            nrow=4)  
rownames(ex) <- c("아이폰", "갤럭시", "좋다", "나쁘다")  
colnames(ex) <- c("사람1", "사람2", "사람3")
```

# Co-occurrence Matrix

- 특정 단어와 다른 단어가 동시에 영화평 내에서 발생한 것을 Counts
- 예시)

ex %\*% t(ex)

##	아이폰	갤럭시	좋다	나쁘다
## 아이폰	2	1	2	0
## 갤럭시	1	2	1	1
## 좋다	2	1	2	0
## 나쁘다	0	1	0	1

# Co-occurrence Matrix

```
rowSums(key_tdm_m)[1:5]
```

```
##          and an개인적으로는          avengers          bad          bbb  
##          2          1          1          1          1
```

```
order(rowSums(key_tdm_m), decreasing = T)[1:5]
```

```
## [1] 2797 3584 4013 528 2292
```

```
key_tdm_m <- key_tdm_m[order(rowSums(key_tdm_m), decreasing = T),]
```

# Co-occurrence Matrix

```
key_tdm_m <- key_tdm_m[1:20, ]  
co_matrix <- key_tdm_m %*% t(key_tdm_m)  
co_matrix[1:5,1:5]
```

```
##           Terms  
## Terms      어벤져스 재미 지루 기대 스토리  
## 어벤져스    988  104   75   66    99  
## 재미        104  282   35   29    24  
## 지루         75   35  212   28    25  
## 기대         66   29   28  206    11  
## 스토리        99   24   25   11   192
```



# Term x Document Matrix와 감정 사전

```
groups_list = list()
groups_list$비호감단어 = which(colnames(co_matrix) %in% neg_word)
groups_list$호감단어 = which(colnames(co_matrix) %in% pos_word)
groups_list
```

```
## $비호감단어
## [1]  3 18 20
##
## $호감단어
## [1]  2  4
```

## 7. 시각화

# 7. 시각화

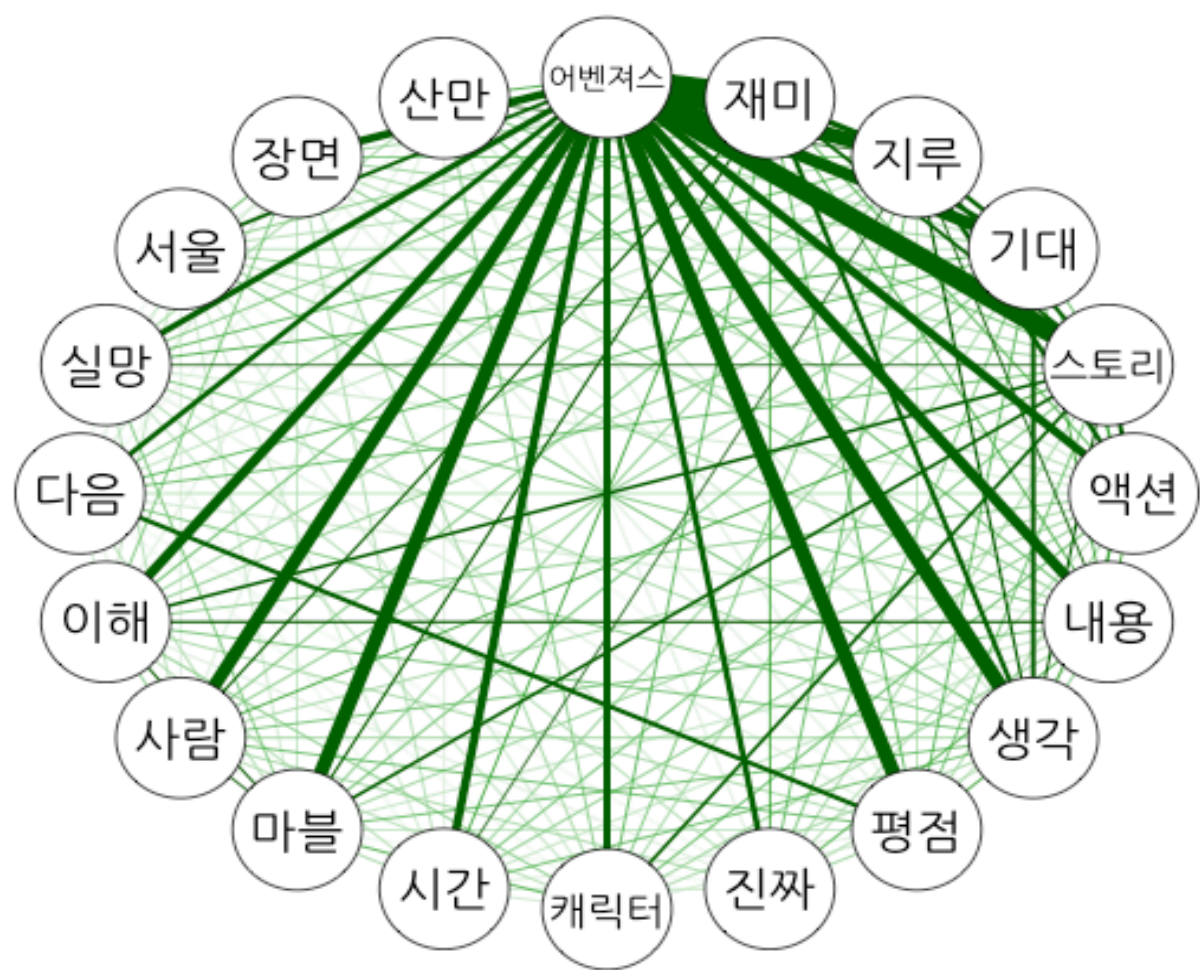
## 학습 목표

- Graph 그리기(qgraph)

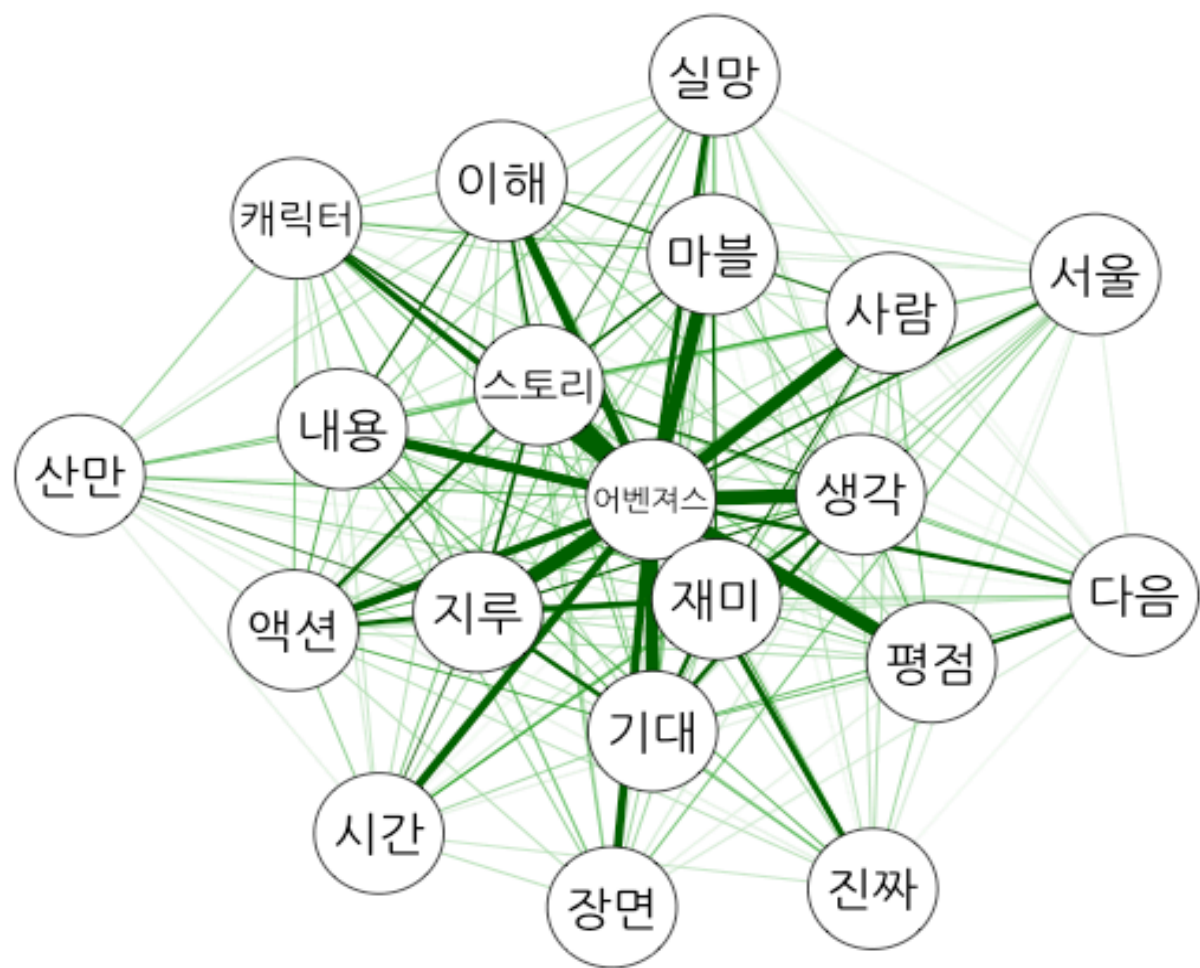
# Graph

```
library("qgraph")
```

```
qgraph(co_matrix, labels = colnames(co_matrix), diag=F)
```

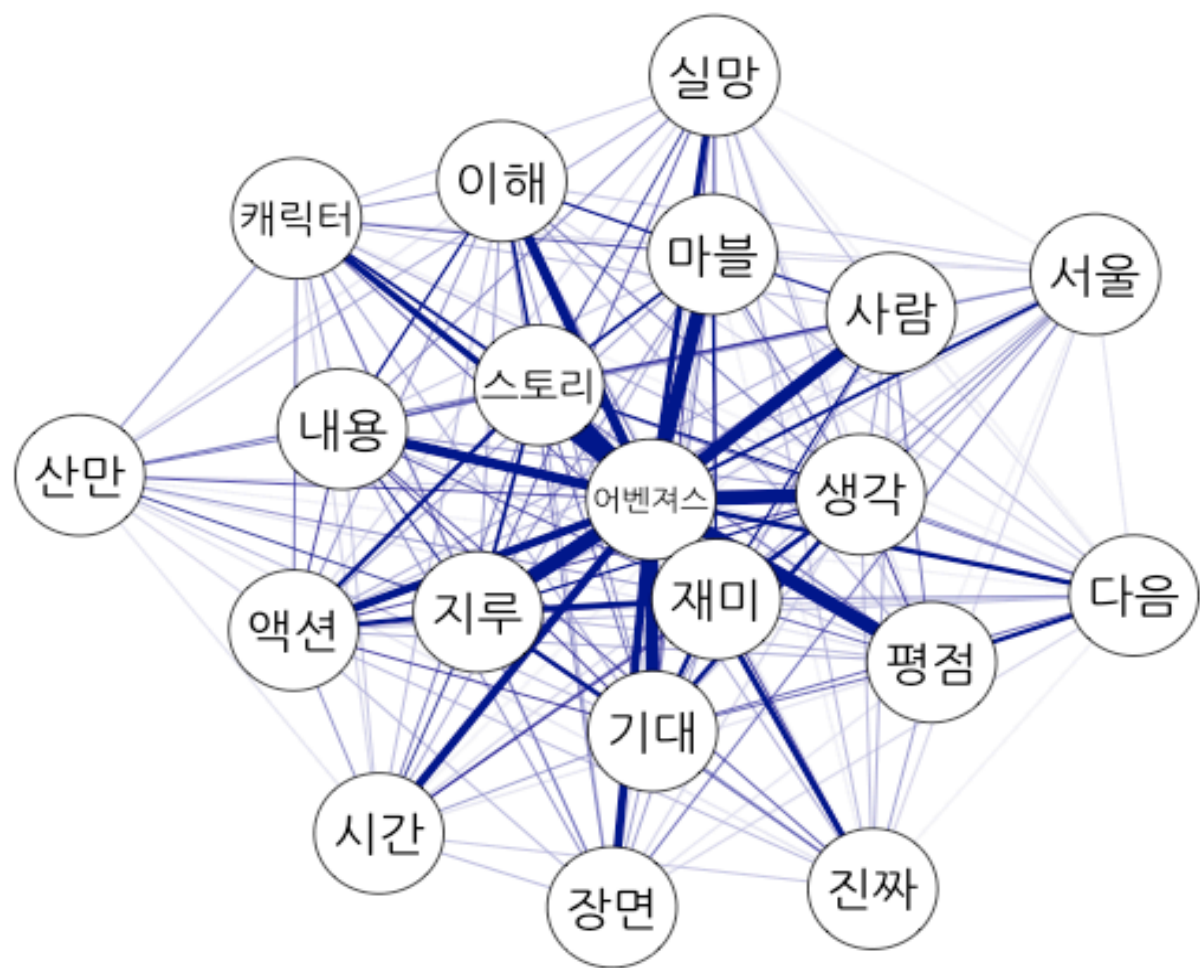


```
qgraph(co_matrix, labels = colnames(co_matrix), diag=F,  
       layout="spring")
```

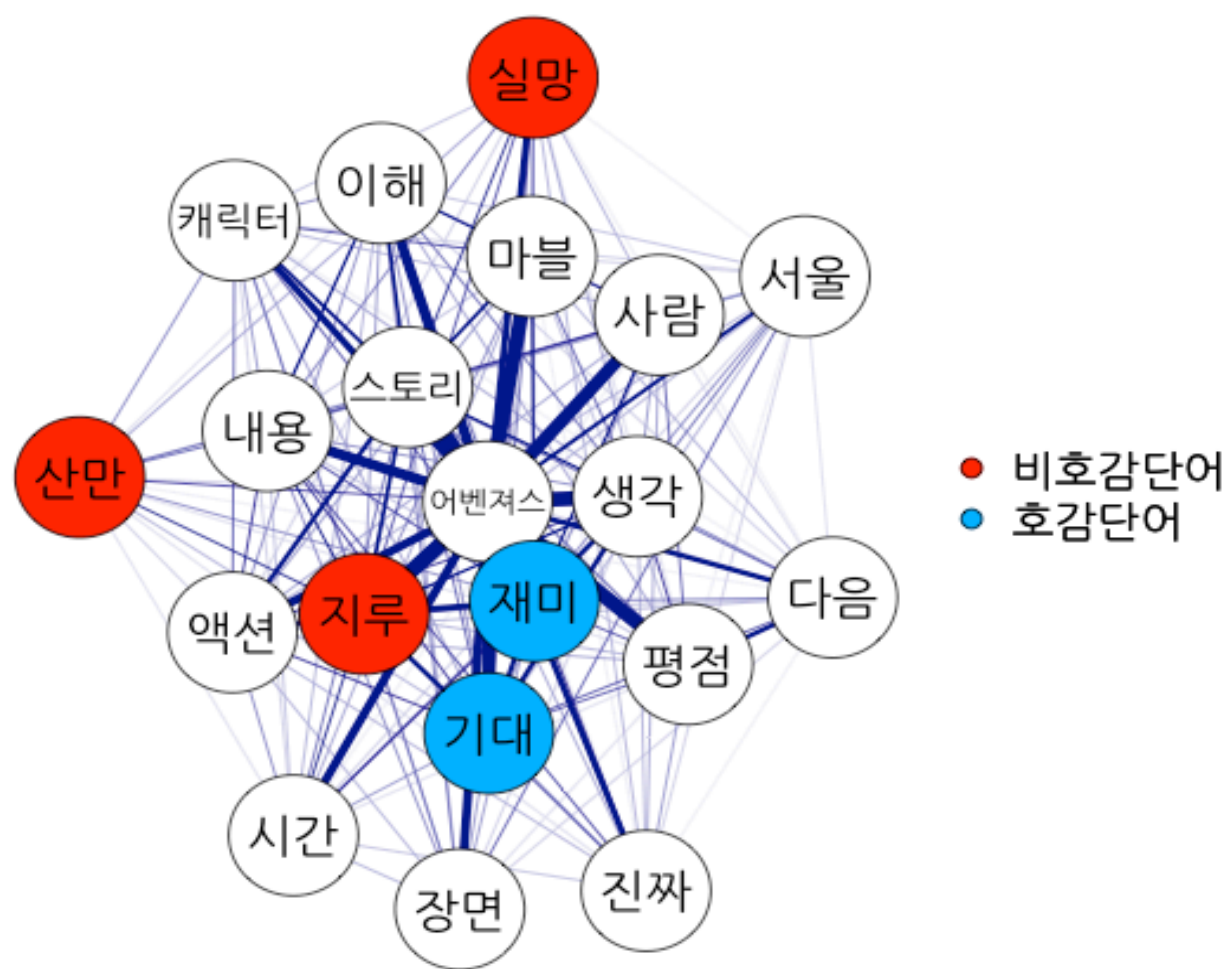


```
qgraph(co_matrix, labels = colnames(co_matrix), diag=F,  
       layout="spring",  
       edge.color = "darkblue")
```



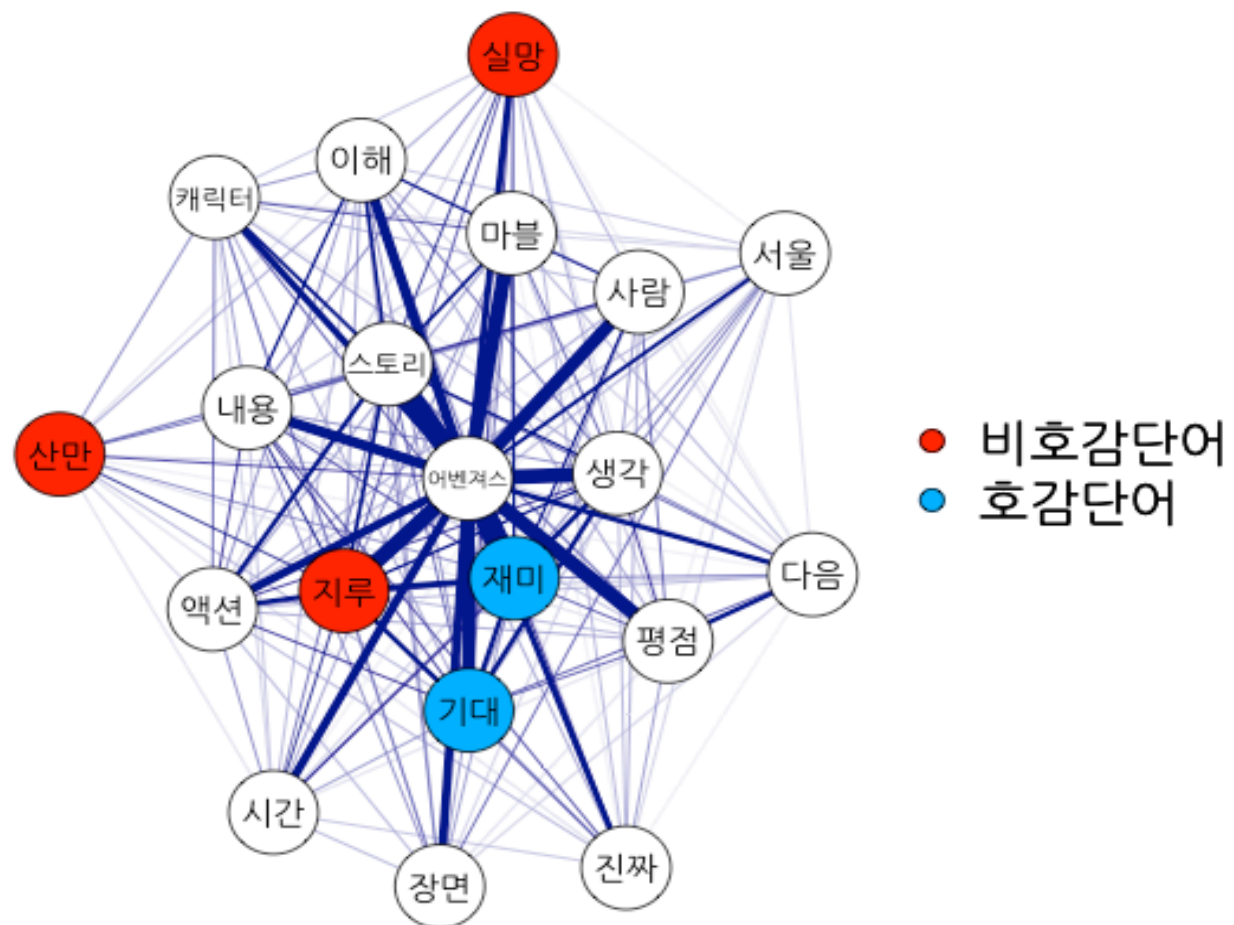


```
qgraph(co_matrix, labels = colnames(co_matrix), diag=F,  
       layout="spring",  
       edge.color = "darkblue",  
       groups = groups_list)
```



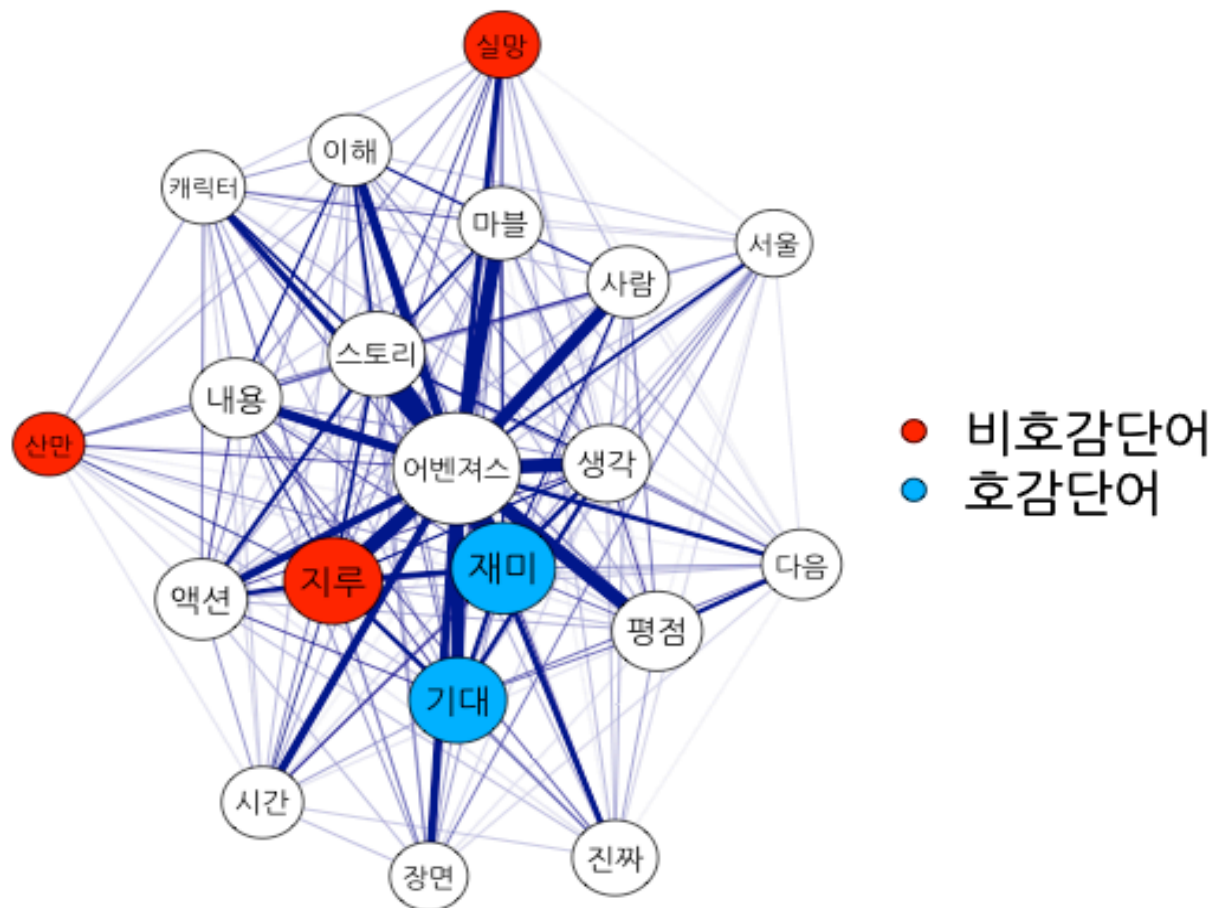
```
qgraph(co_matrix, labels = colnames(co_matrix), diag=F,  
       layout = "spring",  
       edge.color = "darkblue",  
       groups = groups_list,  
       vsize = 5,  
       legend.cex = .7)  
  
title(movie_name, line = 3)
```

## 어벤져스



```
qgraph(co_matrix, labels = colnames(co_matrix), diag=F,  
       layout = "spring",  
       edge.color = "darkblue",  
       groups = groups_list,  
       size = log(diag(co_matrix)),  
       legend.cex = .7)  
  
title(movie_name, line = 3)
```

## 어벤져스



# 의문 단어 찾아보기(상관관계)

```
head(findAssocs(key_tdm, "서울", 0))
```

```
##          서울
## 후진국   0.31
## 배경    0.28
## 이미지가 0.27
## 건물외관 0.25
## 뉴스방송 0.25
## 당근     0.25
```



# 의문 단어 찾아보기(상관관계)

```
head(findAssocs(key_tdm, "다음", 0))
```

```
##           다음
## 검색      0.36
## 개봉많이  0.35
## 네이버보다 0.35
## 물타기알바 0.35
## 본사람들  0.35
## 빵점      0.35
```

## **APPENDIX I - R 연습 (REGEX)**

# REGEX (연습 1)

```
ex <- c("아!! R이 왜 이렇게 재미있냐?", "[1]", "[2]", "[20]", "줄리다...")
str_replace_all(ex, '!', '')
```

```
## [1] "아 R이 왜 이렇게 재미있냐?" "[1]"
## [3] "[2]"                        "[20]"
## [5] "줄리다..."
```

```
str_replace_all(ex, "! | \\. ", "")
```

```
## [1] "아!R이 왜 이렇게 재미있냐?" "[1]"
## [3] "[2]"                        "[20]"
## [5] "줄리다..."
```

```
str_replace_all(ex, "[[:punct:]]", "")
```

```
## [1] "아 R이 왜 이렇게 재미있냐" "1"
## [3] "2"                        "20"
## [5] "줄리다"
```

# REGEX (연습 1)

```
str_replace_all(ex, "1", "")
```

```
## [1] "아!! R이 왜 이렇게 재미있냐?" "[ ]"  
## [3] "[2]" "[20]"  
## [5] "줄리다..."
```

```
str_replace_all(ex, "[1]", "")
```

```
## [1] "아!! R이 왜 이렇게 재미있냐?" "[ ]"  
## [3] "[2]" "[20]"  
## [5] "줄리다..."
```

```
str_replace_all(ex, "\\[1\\]", "")
```

```
## [1] "아!! R이 왜 이렇게 재미있냐?" ""  
## [3] "[2]" "[20]"  
## [5] "줄리다..."
```

# REGEX (연습 1)

?regex

# REGEX (연습 1)

```
str_replace_all(ex, "\\[[0-9]\\]", "")
```

```
## [1] "아!! R이 왜 이렇게 재미있냐?" ""  
## [3] "" "[20]"  
## [5] "줄리다..."
```

```
str_replace_all(ex, "\\[[0-9]+\\]", "")
```

```
## [1] "아!! R이 왜 이렇게 재미있냐?" ""  
## [3] "" ""  
## [5] "줄리다..."
```

# REGEX (연습 1)

```
ex <- str_replace_all(ex, "\\[[0-9]+\\]", "")
```

```
ex==" "
```

```
## [1] FALSE TRUE TRUE TRUE FALSE
```

```
ex!=" "
```

```
## [1] TRUE FALSE FALSE FALSE TRUE
```

```
ex[ex!=" "]
```

```
## [1] "아!! R이 왜 이렇게 재미있냐?" "즐리다..."
```

**R 연습**

**APPENDIX II - POS-TAGGING 이용**



# 문장 자르기(split) (연습 3)

```
ex <- "기대보다 아주 재밌음!!!!!!! !! 꼭 봐요~ > < *"
```

```
result <- str_split(ex, "([ㄱ-ㅎㅏ-ㅣ]|[:punct:]|[0-9A-Za-z]|[:space:])+")  
result
```

```
## [[1]]
```

```
## [1] "기대보다" "아주"      "재밌음"   "꼭"       "봐요"     ""
```

## 앞 2글자 추출 (연습 4)

```
result <- str_sub(result[[1]], 1, 2)  
result
```

```
## [1] "기대" "아주" "재밋" "꼭"   "봐요" ""
```

# 맞춤법 교정 (연습 5)

```
plyr::revalue("재밋", c("재밋" = "재미", "재민" = "재미"), warn_missing=F)
```

```
## [1] "재미"
```

```
plyr::revalue(result, c("재밋" = "재미", "재민" = "재미"), warn_missing=F)
```

```
## [1] "기대" "아주" "재미" "꼭" "봐요" ""
```

# 한 문장으로 합치기 (연습 6)

```
paste(c("하", "R은", "정말", "신나"), sep= ' ')
```

```
## [1] "하" "R은" "정말" "신나"
```

```
paste(c("하", "R은", "정말", "신나"), collapse=" ")
```

```
## [1] "하 R은 정말 신나"
```

```
paste(c("하", "R은", "정말", "신나"), collapse="+")
```

```
## [1] "하+R은+정말+신나"
```

# POS Tagging (연습 7)

SimplePos09(ex)

```
## $기대보다
## [1] "기대/N+보다/J"
##
## $아주
## [1] "아주/M"
##
## $재밌음
## [1] "재밌음/N"
##
## $`!!!!~`
## [1] "!!!!/S"
##
## $`!!!~`
## [1] "!!!/S"
##
## $`!!~`
## [1] "!!/S"
##
## $꼭
## [1] "꼭/M"
##
```

# POS Tagging (연습 7)

태그 메뉴얼(pp.16 ~ 17)

```
result <- paste(SimplePos09(ex))  
result
```

```
## [1] "기대/N+보다/J" "아주/M"      "재밌음/N"      "!!!!!!/S"  
## [5] "!!!!/S"         "!!/S"          "꼭/M"          "보/P+아/E+~/S"  
## [9] ">/S"            "</S"           "*/S"
```

# POS Tagging (연습 7)

- 체언(N)과 용언(P)만 추출
- 체언(N) : 보통명사 + 고유명사 + 의존명사 + 대명사 + 수사
- 용언(P) : 동사 + 형용사 + 보조용언

```
result <- str_extract_all(result, "[가-힣]+/P|[가-힣]+/N")
result
```

```
## [[1]]
## [1] "기대/N"
##
## [[2]]
## character(0)
##
## [[3]]
## [1] "재밌음/N"
##
## [[4]]
## character(0)
##
## [[5]]
## character(0)
```

# 체언과 용언 추출 (연습 8)

```
result <- paste(result, collapse = " ")  
result
```

```
## [1] "기대/N character(0) 재밌음/N character(0) character(0) character(0) character(0) 보/P character(0) character(0) cha
```



# 체언과 용언 추출 (연습 8)

```
result <- str_split(result,"([ㄱ-ㅎㅏ-ㅣ]|[:punct:]|[0-9A-Za-z]|[:space:]))+")
result
```

```
## [[1]]
## [1] "기대" "재밌음" "보" ""
```

```
ex ## 앞 두 글자가 오히려 좋을 수도
```

```
## [1] "기대보다 아주 재밌음!!!!!!!!!! !! 꼭 봐요~ > < *"
```

```
split_12(ex)
```

```
## [1] "기대" "아주" "재밌" "!!" "꼭" "봐요" ">" "<" "*" "
```

# 키워드 추출 (어벤져스)

- Step (1) POS Tagging

```
movie_text_sum[1]
```

```
## [1] "괜찮음.. 시원한 액션.. 간적으로 캡틴아메리카 너무 좋아함. 헐크도 좋고.. ♥♥♥"
```

```
key_vec <- SimplePos09(movie_text_sum[1])  
key_vec <- paste(SimplePos09(movie_text_sum[1]), "")  
key_vec
```

```
## [1] "괜찮/P+음/E "      ".. /S "  
## [3] "시원한/N "          "액션/N "  
## [5] ".. /S "             "개/P+L /E+적/N+으로/J "  
## [7] "캡틴아메리카/N "    "너무/M "  
## [9] "좋/P+아/E+하/P+ㅁ/E " ". /S "  
## [11] "헐크도/N "          "좋/P+고/E "  
## [13] ".. /S "             "♥♥♥/N "
```

# 키워드 추출 (어벤져스)

- Step (2) 체언(N)과 용언(P) 추출

```
key_vec <- str_extract_all(key_vec, "[가-힣]+/P|[가-힣]+/N")
key_vec <- paste(key_vec, collapse=" ")
key_vec
```

```
## [1] "관찰/P character(0) 시원한/N 액션/N character(0) c(\"개/P\", \"적/N\") 캡틴아메리카/N character(0)"
```

# 키워드 추출 (어벤져스)

- Step (3) 파싱(문장 자르기)

```
key_vec <- str_split(key_vec,"([ㄱ-ㅎㅏ-ㅣ]|[:punct:]|[0-9A-Za-z]|[:space:]))+")  
key_vec <- key_vec[[1]]  
key_vec
```

```
## [1] "관찰"      "시원한"    "액션"      "개"  
## [5] "적"        "캡틴아메리카" "좋"        "하"  
## [9] "헐크도"   "좋"        " "
```

## 키워드 추출 (어벤져스)

- Step (4) 맞춤법 교정

```
movie_name <- "어벤져스"
key_vec <- plyr::revalue(key_vec, c("재밋" = "재미",
                                     "재밋" = "재미",
                                     "짬있" = "재미",
                                     "재밋음" = "재미",
                                     "재미있" = "재미",
                                     "지루함" = "지루",
                                     "좋" = " 좋음",
                                     "영화" = movie_name,
                                     "캐릭" = "캐릭터"),
                           warn_missing = F)
key_vec
```

##	[1]	"관찰"	"시원한"	"액션"	"개"
##	[5]	"적"	"캡틴아메리카"	"좋음"	"하"
##	[9]	"헐크도"	"좋음"	" "	

# 키워드 추출 (어벤져스)

- Step (5) 다시 한 문장으로 합치기

```
key_vec <- paste(key_vec, collapse=" ")  
key_vec
```

```
## [1] "괜찮 시원한 액션 개 적 캡틴아메리카 좋음 하 헐크도 좋음 "
```

# 키워드 추출 (어벤져스)

```
key_vec_sum <- c(); movie_name = "어벤져스"
for (i in 1:length(movie_text_sum))
{
  key_vec <- paste(SimplePos09(movie_text_sum[i]))
  key_vec <- str_extract_all(key_vec, "[가-힣]+/P|[가-힣]+/N")
  key_vec <- paste(key_vec, collapse=" ")
  key_vec <- str_split(key_vec, "([ㄱ-ㅎㅌ-ㄴ]|[:punct:])|[0-9A-Za-z]|[:space:])+")
  key_vec <- key_vec[[1]]
  movie_name <- "어벤져스"
  key_vec <- plyr::revalue(key_vec, c("재밋" = "재미",
                                       "재밋" = "재미",
                                       "잼있" = "재미",
                                       "재밋음" = "재미",
                                       "재미있" = "재미",
                                       "지루함" = "지루",
                                       "좋" = "좋음",
                                       "영화" = movie_name,
                                       "캐릭" = "캐릭터"),
                           warn_missing = F)
  key_vec <- c(key_vec, ' ') ## 윈도우 tm 버그 때문
  key_vec_sum[i] <- paste(key_vec, collapse=' ') ## 두 칸 (윈도우 tm 버그 때문)
}
```

## Appendix III - 앞 2글자 자르기



# 키워드 추출 (어벤져스)

- Step 1) 문장 자르기(split)

```
## Warning: cannot open file
## '/Users/kimhyungjun/repo/daum_movieavengers_text.csv': No such file or
## directory
```

```
## Error: cannot open the connection
```

```
key_vec <- str_split(movie_text_sum[1],
                      "([ㄱ-ㅎㅏ-ㅣ]|[:punct:]|[0-9A-Za-z]|[:space:]))+" )
key_vec
```

```
## [[1]]
## [1] "관찰음"      "시원한"      "액션"        "간적으로"
## [5] "캡틴아메리카" "너무"        "좋아함"      "헐크도"
## [9] "좋고"        ""
```

# 키워드 추출 (어벤져스)

- Step 2) 앞 2글자 추출

```
key_vec <- str_sub(key_vec[[1]], 1, 2)  
key_vec
```

```
## [1] "관찰" "시원" "액션" "간적" "캡틴" "너무" "좋아" "헐크" "좋고" ""
```

# 키워드 추출 (어벤져스)

- Step 3) 맞춤법 교정

```
movie_name <- "어벤져스"
key_vec <- plyr::revalue(key_vec, c("재밌" = "재미",
                                     "재밋" = "재미",
                                     "잼있" = "재미",
                                     "영화" = movie_name,
                                     "스토" = "스토리",
                                     "시사" = "시사회",
                                     "어벤" = "어벤져스",
                                     "아이" = "아이언맨",
                                     "히어" = "히어로",
                                     "어벤져스" = "어벤져스",
                                     "캐릭" = "캐릭터",
                                     "울트" = "울트론"),
                             warn_missing = F)

key_vec
```

```
## [1] "관찰" "시원" "액션" "간적" "캡틴" "너무" "좋아" "헐크" "좋고" ""
```

# 키워드 추출 (어벤져스)

- Step 4) 다시 한 문장으로 합치기

```
key_vec <- paste(key_vec, collapse=" ")  
key_vec
```

```
## [1] "괜찮 시원 액션 갠적 캡틴 너무 좋아 헐크 좋고 "
```

# 키워드 추출 (어벤져스)

```
key_vec_sum <- c(); movie_name = "어벤져스"
for (i in 1:length(movie_text_sum))
{
key_vec <- str_split(movie_text_sum[i], "([ㄱ-ㅎㅏ-ㅣ]|[:punct:]|[0-9A-Za-z]|[:space:])+")

key_vec <- str_sub(key_vec[[1]], 1, 2)
movie_name <- "어벤져스"
key_vec <- plyr::revalue(key_vec, c("재밋" = "재미",
                                     "재밋" = "재미",
                                     "잼있" = "재미",
                                     "영화" = movie_name,
                                     "스토" = "스토리",
                                     "시사" = "시사회",
                                     "어벤" = "어벤져스",
                                     "아이" = "아이언맨",
                                     "히어" = "히어로",
                                     "어벤져스" = "어벤져스",
                                     "캐릭" = "캐릭터",
                                     "울트" = "울트론"),
                           warn_missing = F)
key_vec <- c(key_vec, ' ') ## 윈도우 tm 버그 때문
key_vec_sum[i] <- paste(key_vec, collapse=' ') ## 두 칸 (윈도우 tm 버그 때문)
}
```

# 긍정과 부정 - 중복 단어

```
pos_neg_word <- c(pos_word, neg_word)
pos_neg_word[duplicated(pos_neg_word)]
```

```
## [1] "구슬프" "벅차" "사상" "새삼스럽" "서글프" "서운"
## [7] "섭섭하" "수작" "아리" "애끓" "애처롭" "애타"
## [13] "염려" "욕심나" "유행" "이변" "탈피" "탐나"
## [19] "태평"
```

# 긍정과 부정 - 중복 단어

- 하나의 대안

```
negation_words <- c("없", "않", "안")
ex <- "재미없다. 기대안했다 헐크 웃기다 웃음!!"
sapply(str_split(ex, " ")[[1]],
       function(x) if ( str_sub(x,3,3) %in% negation_words )
                     str_sub(x,1,3) else { str_sub(x,1,2) },
       USE.NAMES = F)
```

```
## [1] "재미없" "기대안" "헐크"  "웃기"   "웃음"
```