데이터 예측 모델과 기계학습의 응용 교과목 보고서

리더스 어플 데이터의 활용방안 및 추가 제언

독서 시간과 작가를 중심으로

한양대학교
HANYANG UNIVERSITY

언론정보대학 정보사회미디어학과 / 이창준 교수
2조(김민수, 김성은, 김연우, 정수빈, 주영)

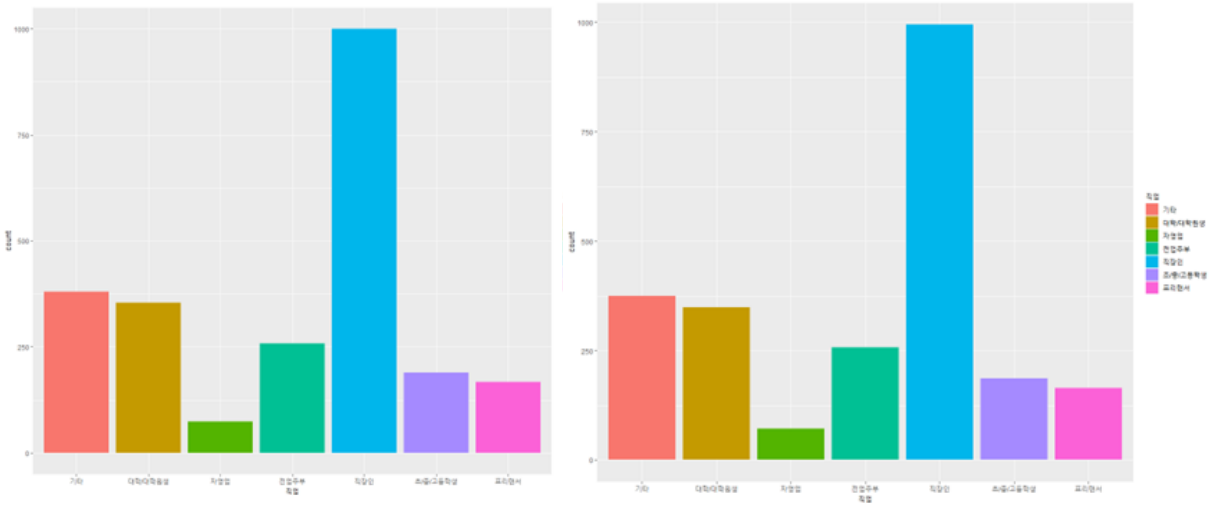
1. 서론

- 리더스 이용자가 독서를 시작한 시간

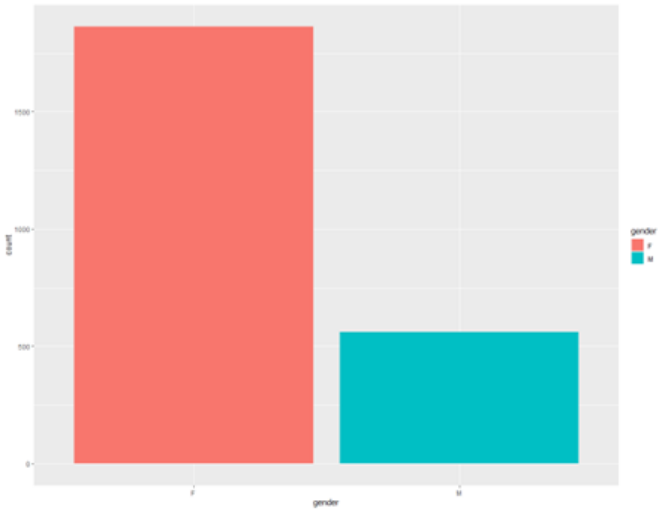
  리더스 어플의 이용 현황(유저 정보) - 직업 카테고리와 성별을 기준으로

  (*리더스 소비자(유저)를 설명할 가장 높은 대표성을 지닌 카테고리라 파악)

  - 선행연구 결과



▲ 전체 유저 직업 분포/완독 유저 직업 분포



▲ 전체 유저 성별 분포

전체 유저의 직업 분포와 완독 유저의 직업 분포를 봤을 때 유의미한 차이가 도출되지 않았다. 이는 리더스를 활용하는 소비자의 리더스 어플 이용 행태가 긍정적임을 시사한다고 볼 수 있다. 비슷한 분포 양상은 곧, 리더스 소비자 대부분이 독서(완독)를 했다는 것을 의미하기 때문이다.

기본적으로, 리더스를 활용하지 않는다면, 완독 유저의 분포가 전체 유저의 분포와 차이가 있어야하지만 그러지 아니했다. 따라서, 마케팅 전략을 위한 데이터 분석, 추출 및 활용을 하기 전, 소비자의 리더스 이용 현황(행태)를 살펴본 결과, 리더스 자체의 높은 활용률이 긍정적인 형태를 띔을 확인할 수 있었다.

또한, 데이터 분석 결과, 직장인>대학/대학원생>전업주부>초중고등학생>프리랜서>자영업 순으로 리더스를 활용함을 살펴볼 수 있다. 이어, 여성 소비자가 남성 소비자의 약 3배 이상 활용하는 것을 선행연구를 통해 확인했다.

- 사전조작화

다양한 카테고리를 일정하게 범주화 하고자 했다. Table 로 살펴본 리더스 내의 많은 카테고리를 다음과 같이 8개의 카테고리로 공통적으로 분류해 후에 데이터를 활용할 때 동일하게 적용했다.

100개 미만으로 나타나는 카테고리는 유의미한 관계에 영향을 미치지 않을 듯 해 제외했고, 분야가 비슷한 것들은 각각의 카테고리로 묶어 새로운 카테고리 분류 요소를 추가했다. 완성한 조작화된 카테고리는 다음과 같다.

가정/취미/레저	가정/요리/뷰티, 가정/원예/인테리어, 건강/스포츠, 건강/취미/레저. 공예/취미/수집,여행, 요리, 좋은부모
인문/사회	자기계발, 인문학, 인문/사회, 법률, 사회과학, 역사, 경제경영
과학	과학, 기술공학, 자연과학, 의학
예술	만화, 소설/시/희곡, 예술/대중문화
어린이	어린이, 유아, 전집/중고전집
종교	종교/명상/점술, 종교/역학
교육	교육자료, 대학교재/전문서적, 수험서/자격증, 외국어, 컴퓨터/모바일, 중고등참고서, 중학교참고서, 청소년, 초등참고서, 초등학교참고서, 교육자료, 대학교재/전문서적

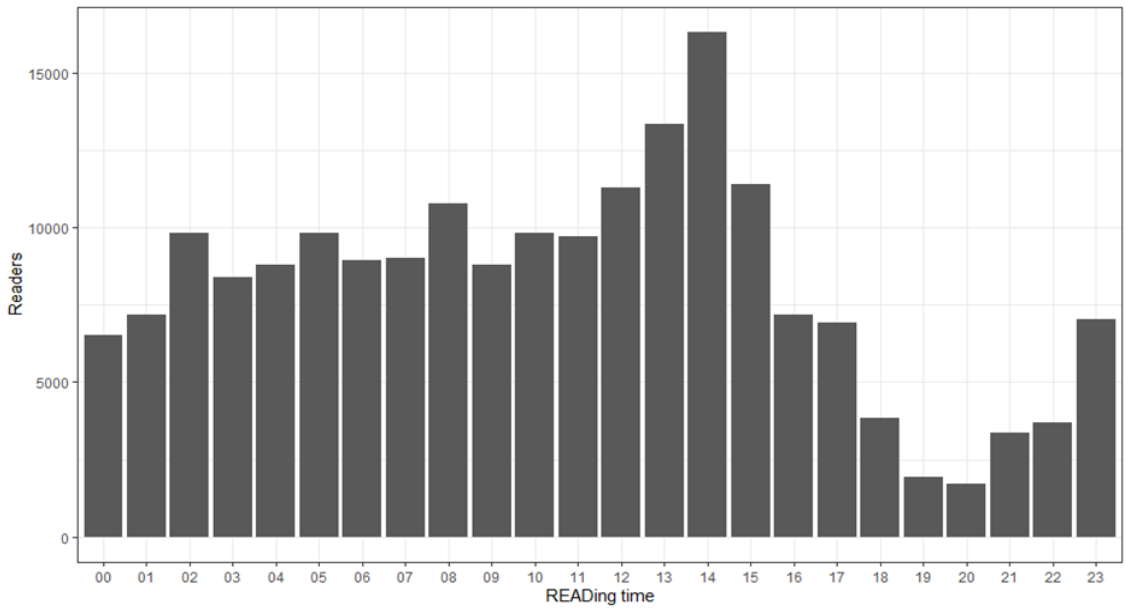
2. 데이터 시각화

우리는 도입 부분을 통해서 '시간대'에 초점을 두기로 했다. 여러 가지 가설을 세우던 중, '대부분의 사람들이 일과가 끝난 저녁 시간대에 독서를 시작할 것이다'라는 가설을 핵심 가설로 채택하였다. 리더스로 소비자 마케팅을 전략하기 위해서는 리더스를 이용하는 특정 시간대의 소비자 분포를 살펴봐야 할 필요가 있다고 파악했기 때문이다.

따라서 24시간 중 특정 시간 대에 범주하는 상위 3개의 시간대와 하위 3개의 시간대의 소비자 이용률을 살핀 후, 해당 시간대에 사람들이 주로 읽는 분야, 즉 책의 카테고리를 파악하고자 했다. 이를 파악하면, 리더스에 들어오는 적정한 시간대의 소비자들에게 추천해줄 수 있는 분야를 공략할 수 있을 것이라 생각했기 때문이다.

해당 가설을 증명하고자 데이터 시각화를 진행하였다. 성별, 직업까지 고려하기에는 너무 많은 변수가 개입됨에 따라, 한정적인 결과를 초래할 수 있다는 문제점이 있어 책의 카테고리로만 분류를 했다. 책 카테고리는 table를 우선적으로 살펴 보았다. 이후 100미만의 카테고리는 유의미한 관계에 영향이 없을 것이라 판단해 삭제했다. 또한, 다양한 분야를 8가지의 카테고리로 분류해 쉽게 소비자의 정보를 파악할 수 있도록 재설정 했다는 점에서 의의가 있다.

- 리더스 이용자가 독서를 시작한 시간

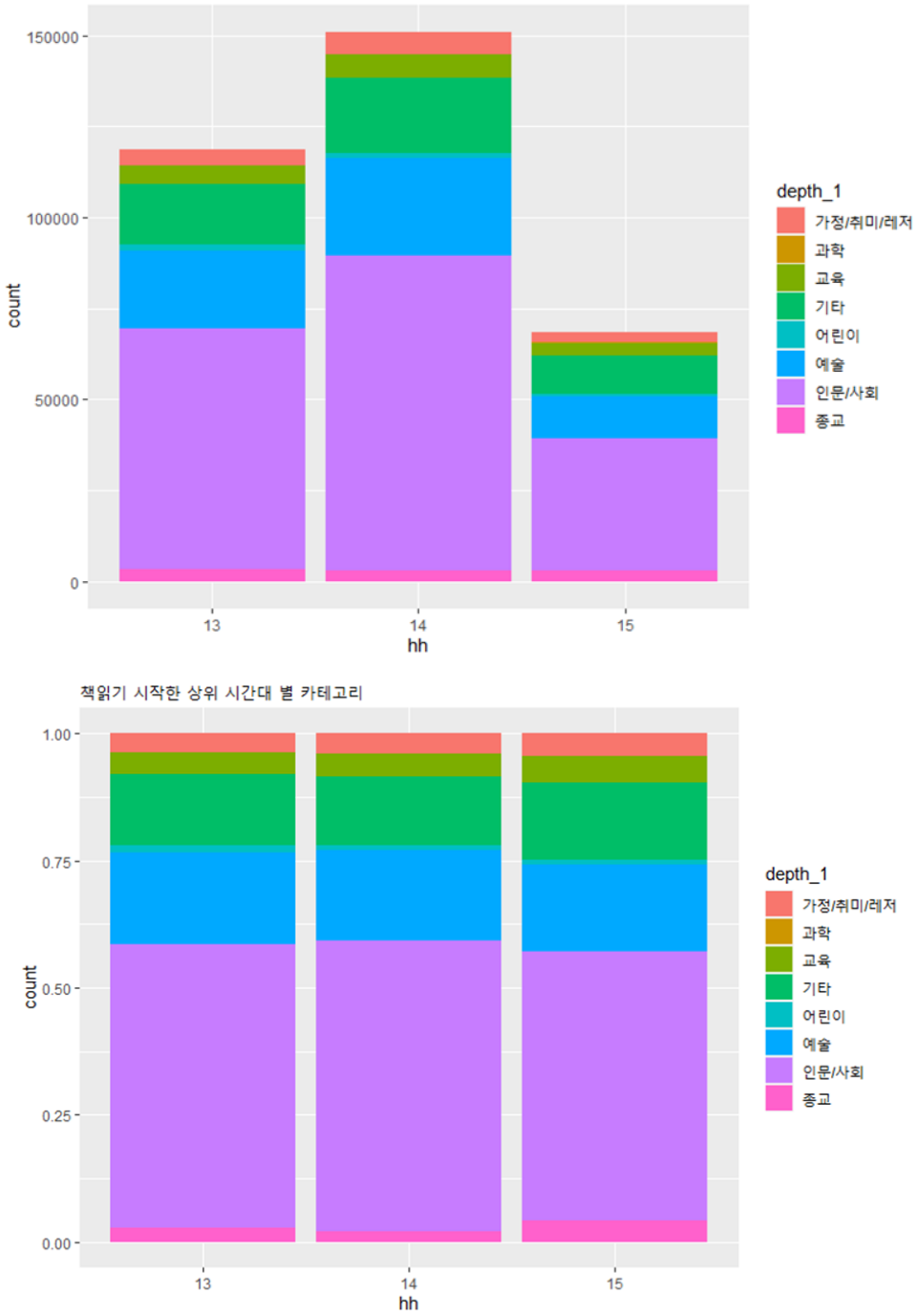


▲독서를 시작한 시간을 나타낸 그래프

저녁시간에 책을 읽기 시작할 것이라 예상한 것과 달리 낮 시간대의 이용률이 높았고, 오히려 늦은 저녁 시간대의 이용률이 저조한 것을 확인할 수 있었다. 오후 2시에 가장 많은 이용률을 보였으며, 20시에 가장 저조한 이용률을 보였다. 13시부터 15시까지의 이용률이 가장 높은 상위 이용률이고, 19시부터 21시까지의 이용률이 가장 낮은 하위 이용률이라고 설정하였다.

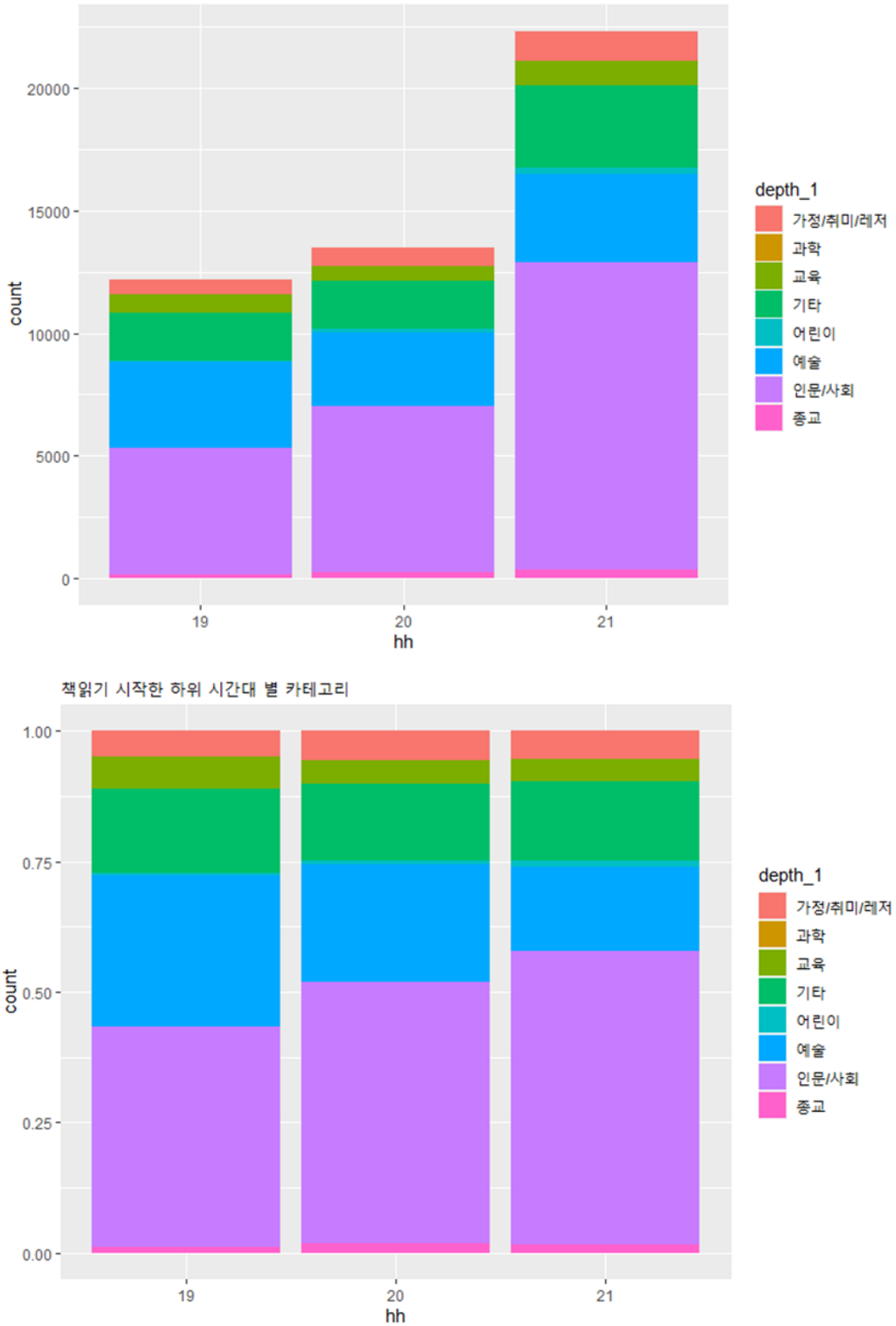
그 후, 상위 이용률 시간대 별 도서 카테고리와 하위 이용률 시간대 별 도서 카테고리의 변화가 있는지 파악하기 위한 시각화 작업을 진행하였다. 카테고리의 경우, 앞서 분류한 카테고리를 사용하였다.

- 리더스 이용자가 독서를 시작한 시간(카테고리 빈도/비율)



▲독서를 시작한 상위 3개 시간대 카테고리 시각화 그래프(13-15시)

상위 이용률의 그래프에서는 13시와 14시에 가장 높은 이용률을 보이며, 상위 3번째 시간대인 15시에 약간의 차이가 나는 것을 확인할 수 있다. 또한, 동일 비율로 측정한 결과 인문사회/예술 등 비슷한 비율분포를 보이고 있는 것을 확인할 수 있다.



▲독서를 시작한 하위 3개 시간대 카테고리 시각화 그래프(19-21시)

하위 이용률의 그래프는 19시에 가장 저조한 이용률을 보이며, 19시와 20시에 대부분의 소비자가 리더스에서 책을 활용하지 않음을 확인할 수 있다. 기존에 추측했던 것과 달리 살펴본 그래프 값으로, 해당 시간대에 이용률이 상당히 저조함을 알 수 있다. 동일 비율로 측정한 결과, 상위 이용률의 카테고리보다 차이 있는 시간대별 카테고리 이용률을 확인할 수 있다. 시간대가 뒤로 갈수록, 인문/사회의 영역이 증가하는 것을 보이며, 예술 분야의 이용률이 저조해지는 것을 한눈에 파악할 수 있다.

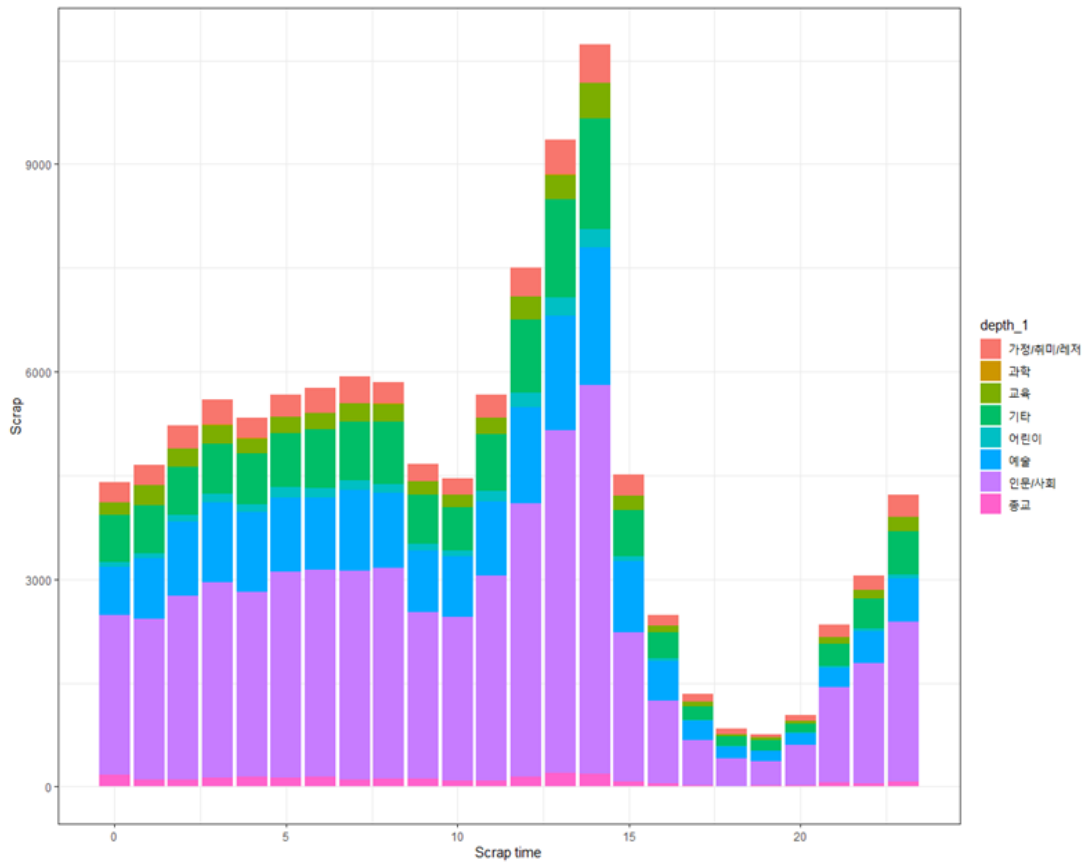
3. 연관 분석을 통한 추천 시스템

독서 서비스 어플 리더스의 특징 중 하나는 책의 한 페이지를 찍어 인스타그램 피드처럼 기록할 수 있다는 점이다. 우리 팀은 이 기능을 보며, '이용자들이 낮에는 자기계발이나 자격증 도서, 저녁에는 에세이나 수필 같은 감성적인 도서를 스크랩 할 것이다'라는 가설을 세웠다. 만약 우리 팀의 가설대로 낮과 밤 시간대 별로 스크랩하는 책이 다르다면 시간대에 따라서 책을 추천하고자 주제를 선정하였다.

시간대 별로 이용자들의 스크랩 책 카테고리 변화를 구하기 위해 book_cat(6), cat(7), scrap(8) 데이터를 결합하여 user_id, book_category, datetime이 포함된 하나의 데이터 프레임을 완성하였다. 책 카테고리는 위의 사용된 8개의 카테고리를 그대로 사용하였다.

스크랩 시간이 제일 많은 시간 대와 제일 적은 시간대, 그리고 이에 대조되는 하나의 시간대 총 3개의 시간대의 사람들이 읽는 책의 카테고리를 파악하고자 하였다. 이를 파악하면, 리더스 어플로 스크랩을 한 유저들에게 본인의 카테고리 패턴과 맞는 추천 시스템을 제공할 수 있을 것이라 생각하기 때문이다.

● 리더스 이용자의 스크랩 빈도(카테고리)



▲ 스크랩 빈도 비교(시간, 카테고리)

'잠들기 직전 시간대의 스크랩 빈도가 가장 높을 것이다'라는 우리 팀의 예측과는 달리 12-14시가 스크랩이 가장 활발하였고 18-20시가 스크랩이 가장 저조한 것을 그래프 분석을 통해 확인하였다. 리더스 이용자 직업 1위가 직장인인 점을 고려한다면, 직장인의 점심시간인 12-14 사이에 가장 스크랩을 많이 하고 퇴근 시간이 18-20시 사이에는 스크랩을 거의 하지 않는다는 결론을 내릴 수 있다. 카테고리는 인문/사회 분야의 책이 각 시간대의 50% 이상을 차지하였고, 다른 카테고리의 책도 대체적으로 전체 빈도에 비례하여 증감하였다. 지금부터는 12-14시, 18-20시, 0-2시로 나누어 시간대 별로 스크랩한 책의 카테고리에서 연관 규칙을 찾고자 한다.

● 시간대별 스크랩 이용자의 카테고리 목록 생성

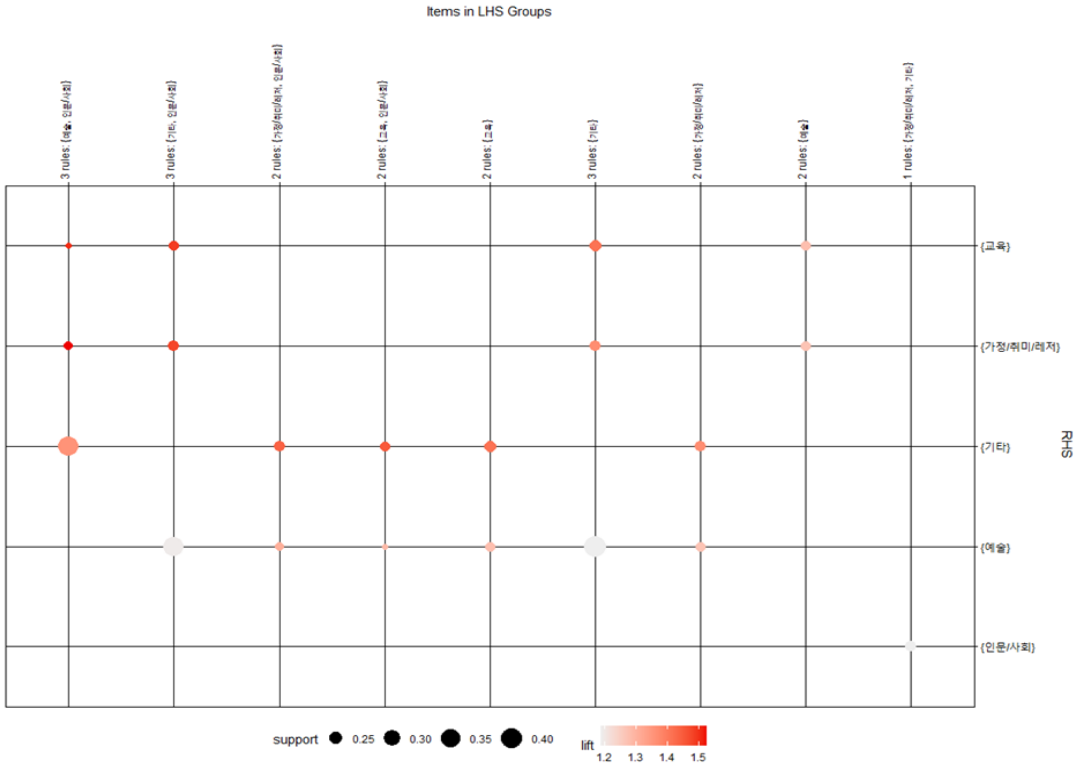
	user_id	category
1	29446	인문/사회,인문/사회
2	73870	기타,예술,예술,기타,인문/사회,기타,인문/사회,기타,인문/사회,기타,인문/사회,기타,인문/사회,기타,인문/사회,기타,인문/사회,기타,인문/사회,기타,인문...
3	414454	인문/사회
4	532918	예술,인문/사회,인문/사회,인문/사회,교육,인문/사회,예술,인문/사회,예술,인문/사회,예술,인문/사회,예술,인문/사회,예술,인문/사회,예술,인문/사회,예술...
5	1117834	인문/사회,기타,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,예술,인문/사회,예술,기타,예술,기타,예술,인문/사회,인문/사회,인문/사...
6	1121536	인문/사회,교육,예술,인문/사회,인문/사회,인문/사회,기타,가정/취미/레저,인문/사회
7	2228434	예술,인문/사회,기타,인문/사회,기타,예술,예술,예술,인문/사회,인문/사회,예술,인문/사회,예술,예술,인문/사회,인문/사회
8	2280262	기타,인문/사회,기타,인문/사회,기타,인문/사회,인문/사회,인문/사회,인문/사회
9	2761522	인문/사회,기타,예술
10	3268696	인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인...
11	3394564	인문/사회,예술
12	3487114	인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회
13	3668512	인문/사회,인문/사회,인문/사회,인문/사회,예술,기타,인문/사회,기타,인문/사회,인문/사회,기타,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인...
14	3742552	예술,인문/사회
15	3786976	예술,인문/사회
16	3979480	인문/사회,예술,인문/사회,예술,예술
17	5519512	인문/사회,가정/취미/레저,기타,인문/사회
18	5926732	어린이,예술,어린이,예술
19	6015580	인문/사회,예술,예술,예술,예술,예술,예술,기타,예술,예술,예술,예술,인문/사회,예술
20	6948484	인문/사회
21	7103968	인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,인문/사회,예술,인문/사회,인문/사회,인문/사회
22	7733308	인문/사회,예술

▲ 스크랩 이용자의 카테고리 목록(0시)

위 데이터 프레임은 0시에 스크랩한 이용자의 목록을 나타낸 것이다. user_id가 29446번인 이용자는 인문/사회 카테고리 책을 두 권 읽은 것이고, 2761522번의 이용자는 인문/사회 책과 예술 책 2권을 읽은 것이다. 위처럼 시간대별로 이용자의 스크랩한 책의 카테고리를 정렬한 뒤 시간대를 크게 3개로 분류하여 Apriori 알고리즘을 적용하여 스크랩 패턴을 살펴보고자 하였다.

1. *12-14시(스크랩 빈도가 가장 높음)*
2. *18-20시(스크랩 빈도가 가장 낮음)*
3. *0-2시 (스크랩 빈도가 가장 높았던 12-14시와 대조하기 위함)*

● 12-14시 스크랩 카테고리 패턴 분석



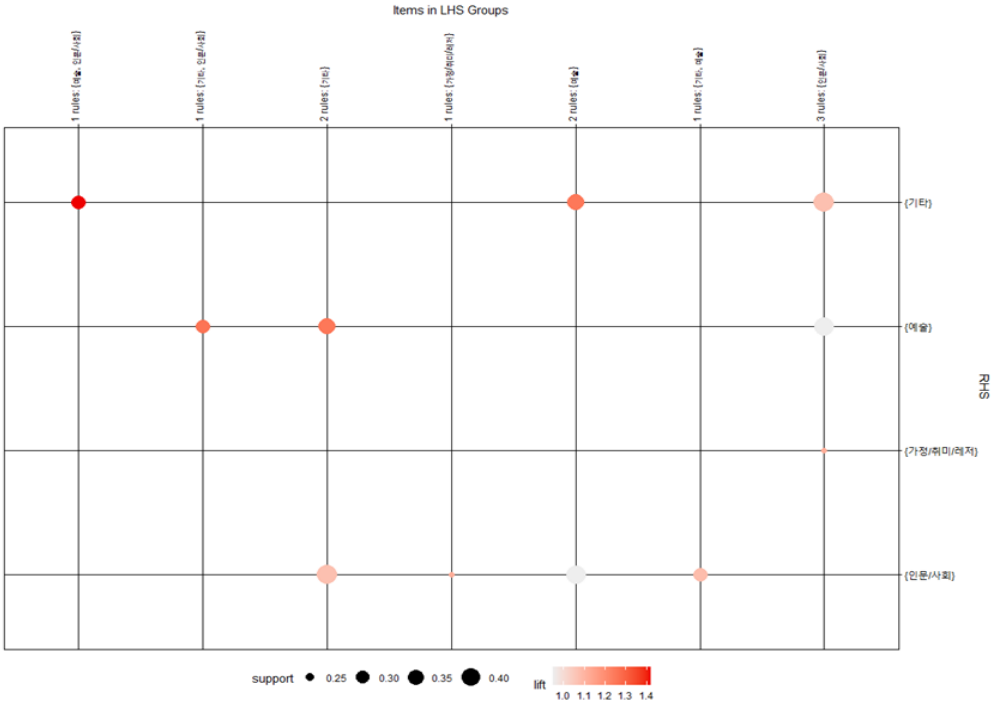

```
> inspect(sort(`basket_rules_12-14`, by="lift"))
     lhs                          rhs                 support   confidence coverage  lift     count
[1]  {예술, 인문/사회}            => {가정/취미/레저} 0.2178899 0.4460094  0.4885321 1.5251772 190
[2]  {예술, 인문/사회}            => {교육}           0.2087156 0.4272300  0.4885321 1.5021960 182
[3]  {기타, 인문/사회}            => {교육}           0.2247706 0.4233261  0.5309633 1.4884693 196
[4]  {기타, 인문/사회}            => {가정/취미/레저} 0.2293578 0.4319654  0.5309633 1.4771524 200
[5]  {교육, 인문/사회}            => {기타}           0.2247706 0.8448276  0.2660550 1.4501765 196
[6]  {가정/취미/레저, 인문/사회}  => {기타}           0.2293578 0.8368201  0.2740826 1.4364313 200
[7]  {교육}                       => {기타}           0.2339450 0.8225806  0.2844037 1.4119888 204
[8]  {기타}                       => {교육}           0.2339450 0.4015748  0.5825688 1.4119888 204
[9]  {가정/취미/레저}             => {기타}           0.2327982 0.7960784  0.2924312 1.3664968 203
[10] {기타}                       => {가정/취미/레저} 0.2327982 0.3996063  0.5825688 1.3664968 203
```

```
> inspect(sort(`basket_rules_12-14`, by="confidence"))
     lhs                          rhs                 support   confidence coverage  lift     count
[1]  {가정/취미/레저, 기타}       => {인문/사회}      0.2293578 0.9852217  0.2327982 1.1899076 200
[2]  {가정/취미/레저, 예술}       => {인문/사회}      0.2178899 0.9644670  0.2259174 1.1648410 190
[3]  {교육, 기타}                 => {인문/사회}      0.2247706 0.9607843  0.2339450 1.1603932 196
[4]  {교육, 예술}                 => {인문/사회}      0.2087156 0.9430052  0.2213303 1.1389204 182
[5]  {가정/취미/레저}             => {인문/사회}      0.2740826 0.9372549  0.2924312 1.1319754 239
[6]  {교육}                       => {인문/사회}      0.2660550 0.9354839  0.2844037 1.1298365 232
[7]  {기타, 예술}                 => {인문/사회}      0.3864679 0.9157609  0.4220183 1.1060159 337
[8]  {기타}                       => {인문/사회}      0.5309633 0.9114173  0.5825688 1.1007700 463
[9]  {교육, 인문/사회}            => {기타}           0.2247706 0.8448276  0.2660550 1.4501765 196
[10] {가정/취미/레저, 인문/사회}  => {기타}           0.2293578 0.8368201  0.2740826 1.4364313 200
```

▲ 스크랩 이용자의 카테고리 Apriori 분석(12-14시)

시간대에 따라 이용자의 카테고리를 정리한 데이터 프레임을 완성한 후 이제 시간대 별 그래프를보고자 한다. 첫 번째로 분석할 시간은 스크랩 시간이 가장 많았던 12-14시이다. 12-14시 사이에 스크랩 한 유저들을 선별하여 카테고리를 정리하고 Apriori 알고리즘을 실행하여 lift와 confidence에 대한 내림차순으로 정렬하였다. 생성된 연관 규칙들을 살펴본 결과 {가정/취미/레저, 기타}를 스크랩한 유저들의 98%가 {인문/사회}에 해당하는 책을 스크랩하였으며, {예술, 인문사회}에서 {가정/취미/레저}로 이어지는 스크랩 유저들의 lift값이 1.5로 가장 높았다.

● 0-2시 스크랩 카테고리 패턴 분석

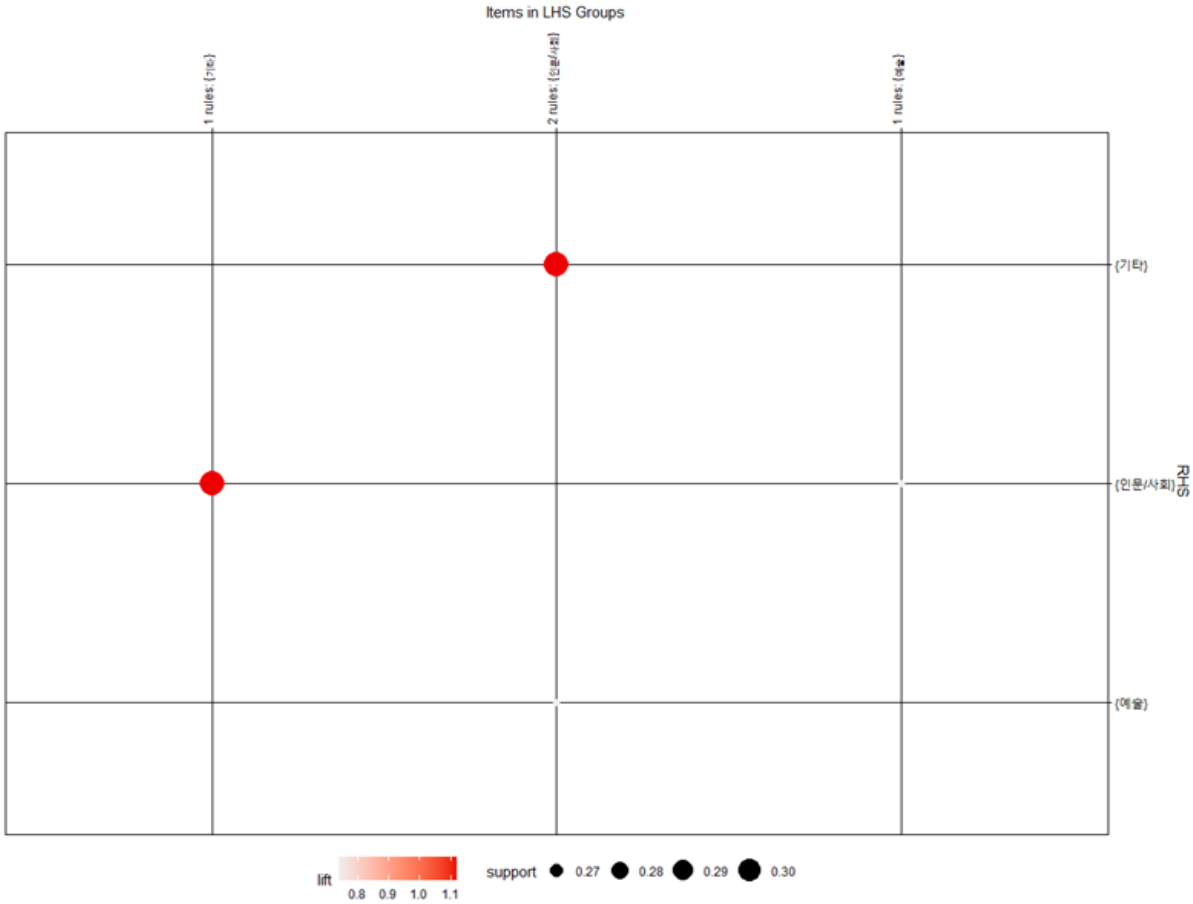



```
> inspect(sort(`basket_rules_0-2`, by="lift"))
     lhs                      rhs              support   confidence coverage  lift      count
[1]  {예술, 인문/사회}      => {기타}           0.3108108 0.7301587  0.4256757 1.4181561 230
[2]  {기타, 인문/사회}      => {예술}           0.3108108 0.7033639  0.4418919 1.2572205 230
[3]  {기타}                 => {예술}           0.3594595 0.6981627  0.5148649 1.2479237 266
[4]  {예술}                 => {기타}           0.3594595 0.6425121  0.5594595 1.2479237 266
[5]  {가정/취미/레저}       => {인문/사회}      0.2432432 0.8955224  0.2716216 1.1156340 180
[6]  {인문/사회}            => {가정/취미/레저} 0.2432432 0.3030303  0.8027027 1.1156340 180
[7]  {기타, 예술}           => {인문/사회}      0.3108108 0.8646617  0.3594595 1.0771879 230
[8]  {기타}                 => {인문/사회}      0.4418919 0.8582677  0.5148649 1.0692224 327
[9]  {인문/사회}            => {기타}           0.4418919 0.5505051  0.8027027 1.0692224 327
[10] {예술}                 => {인문/사회}      0.4256757 0.7608696  0.5594595 0.9478846 315
[11] {인문/사회}            => {예술}           0.4256757 0.5303030  0.8027027 0.9478846 315
> inspect(sort(`basket_rules_0-2`, by="confidence"))
     lhs                      rhs              support   confidence coverage  lift      count
[1]  {가정/취미/레저}       => {인문/사회}      0.2432432 0.8955224  0.2716216 1.1156340 180
[2]  {기타, 예술}           => {인문/사회}      0.3108108 0.8646617  0.3594595 1.0771879 230
[3]  {기타}                 => {인문/사회}      0.4418919 0.8582677  0.5148649 1.0692224 327
[4]  {예술}                 => {인문/사회}      0.4256757 0.7608696  0.5594595 0.9478846 315
[5]  {예술, 인문/사회}      => {기타}           0.3108108 0.7301587  0.4256757 1.4181561 230
[6]  {기타, 인문/사회}      => {예술}           0.3108108 0.7033639  0.4418919 1.2572205 230
[7]  {기타}                 => {예술}           0.3594595 0.6981627  0.5148649 1.2479237 266
[8]  {예술}                 => {기타}           0.3594595 0.6425121  0.5594595 1.2479237 266
[9]  {인문/사회}            => {기타}           0.4418919 0.5505051  0.8027027 1.0692224 327
[10] {인문/사회}            => {예술}           0.4256757 0.5303030  0.8027027 0.9478846 315
[11] {인문/사회}            => {가정/취미/레저} 0.2432432 0.3030303  0.8027027 1.1156340 180
```

▲ 스크랩 이용자의 카테고리 Apriori 분석(0-2시)

두 번째 시간은 12시-14시와 대조되도록 설정한 0-2시이다. 0-2시에 스크랩 한 유저들을 선별하여 카테고리를 정리하고 Apriori 알고리즘을 실행하여 지지도에 대한 내림차순으로 정렬하였다. 생성된 연관 규칙들을 살펴본 결과 {가정/취미/레저}를 스크랩한 유저들의 89%가 {인문/사회}에 해당하는 책을 스크랩하였으며, {기타, 예술}를 스크랩한 유저들의 86%가 {인문/사회} 책을 스크랩한 것을 확인하였다. lift(지지도)와 Confidence(신뢰도)를 바탕으로 12-14시와 비교하였을 때, 약간의 패턴 변화가 나타난 것을 확인하였다. lift 값을 기준으로 12-14시는 {가정/취미/레저} 상위 rule에 분포되어 있으나(5개), 0-2시는 2개 밖에 없었다.

● 18-20시 스크랩 카테고리 패턴 분석




```
> inspect(sort(`basket_rules_18-20`, by="lift"))
    lhs             rhs          support   confidence coverage  lift     count
[1] {인문/사회} => {기타}     0.3092784 0.3964758  0.7800687 1.1201403 90
[2] {기타}      => {인문/사회} 0.3092784 0.8737864  0.3539519 1.1201403 90
[3] {예술}      => {인문/사회} 0.2646048 0.5789474  0.4570447 0.7421748 77
[4] {인문/사회} => {예술}     0.2646048 0.3392070  0.7800687 0.7421748 77
```

▲ 스크랩 이용자의 카테고리 Apriori 분석(18-20시)

마지막 시간은 스크랩 빈도가 가장 적었던 18-20시이다. 관측치가 적어 rule이 4개 밖에 생성되지 않았으며, 그마저도 2개의 lift 값은 1에 미치지 않는다.

시간대 별로 스크랩하는 책(카테고리)의 비중은 크게 차이가 없었다. 하지만 시간대별로 스크랩 한 책들의 목록을 분석하니 시간대별로 다양한 룰이 나온 것을 확인하였다. 이후에도 충분한 양의 이용자 데이터가 확보된다면 시간대에 따라서 이용자들의 스크랩 패턴을 분석하고 그에 따른 책 추천이 가능할 것이라 생각한다. 결론적으로 이용자 {A} → {B} 같은 더 유의미한 결과가 도출된다면 {A}라는 책의 스크랩을 완료한 유저에게 “다음은 {B}을 읽어보는 건 어때요?”처럼 책을 추천해줄 수 있지 않을까 제안한다.

시간대와 카테고리간의 연결성에 대해 탐구한 후, 도서 카테고리가 아닌 작가 간의 유사성은 없는가에 대한 궁금증이 생겼다. 따라서, '작가 간의 연결성'이라는 가설을 설정하여 Apriori rule을 적용해보기로 하였다.

● 저자와의 연관성 분석

우선, book 데이터 중 지은이와 엮은이가 합쳐져 나타나는 'author' 항목을 지은이만 따로 추출한 뒤, 'writer'라는 이름을 주었다. Book 데이터의 writer와 id를 빼낸 다음, 그 빈도가 많은 수부터 내림차순 정렬을 하였다.

	Var1	Freq
52178	히가시노 게이고 (지은이)	202
2261	Roderick Hunt (지은이)	197
17297	무라카미 하루키 (지은이)	197
50831	헤르만 헤세 (지은이)	176
16865	메리 폴 어즈번 (지은이)	169
19786	박현숙 (지은이)	160
1216	J.K. 롤링 (지은이)	151
13245	레프 니콜라예비치 톨스토이 (지은이)	150
41880	조정래 (지은이)	143
20908	변승우 (지은이)	138
32495	윌리엄 셰익스피어 (지은이)	138
19183	박완서 (지은이)	123
26702	아서 코난 도일 (지은이)	123
4395	고정욱 (지은이)	113
34597	이문열 (지은이)	108
12082	데일 카네기 (지은이)	107
27920	앙투안 드 생텍쥐페리 (지은이)	106
41977	조지 오웰 (지은이)	104
52246	히로시마 레이코 (지은이)	104
22847	서지원 (지은이)	99

Showing 1 to 21 of 52,305 entries, 2 total columns

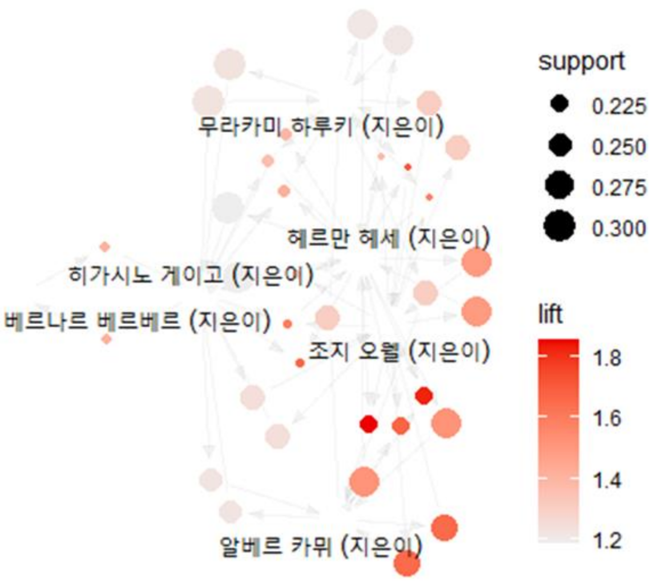
▲동일 작가 서적이 등록된 수에 따라 내림차순한 데이터

위 데이터는 빈도 수에 따라 내림차순한 것이다. 해당 데이터 기준 빈도 상위 30명의 작가를 추출하여 user_book 데이터와 결합하였다. 이때, 유의미한 독서 기록만을 추출하기 위해 독서 중단 상태를 뜻하는 'read_status_stop'을 삭제하였다. 결합한 데이터의 결측값을 확인하고 삭제한 뒤, Apriori rule을 적용하였다. 도출해낸 상위 30명의 작가에 대해 Apriori를 이용해 연관도를 측정하였다. 이때, 최소물품수는 2, 최소지지도는 0.2, 최소신뢰도는 0.1로 설정하였다.

```
> inspect(sort(basket_rules))
     lhs                                              rhs                          support   confidence coverage  lift     count
[1]  {무라카미 하루키 (지은이)}                     => {히가시노 게이고 (지은이)}  0.3066453 0.6288998  0.4875901 1.229258 766
[2]  {히가시노 게이고 (지은이)}                     => {무라카미 하루키 (지은이)}  0.3066453 0.5993740  0.5116093 1.229258 766
[3]  {히가시노 게이고 (지은이)}                     => {헤르만 헤세 (지은이)}      0.3006405 0.5876369  0.5116093 1.186675 751
[4]  {헤르만 헤세 (지은이)}                         => {히가시노 게이고 (지은이)}  0.3006405 0.6071140  0.4951962 1.186675 751
[5]  {조지 오웰 (지은이)}                           => {헤르만 헤세 (지은이)}      0.2986389 0.7408143  0.4031225 1.496002 746
[6]  {헤르만 헤세 (지은이)}                         => {조지 오웰 (지은이)}        0.2986389 0.6030719  0.4951962 1.496002 746
[7]  {알베르 카뮈 (지은이)}                         => {헤르만 헤세 (지은이)}      0.2982386 0.7532861  0.3959167 1.521187 745
[8]  {헤르만 헤세 (지은이)}                         => {알베르 카뮈 (지은이)}      0.2982386 0.6022635  0.4951962 1.521187 745
[9]  {무라카미 하루키 (지은이)}                     => {헤르만 헤세 (지은이)}      0.2922338 0.5993432  0.4875901 1.210315 730
[10] {헤르만 헤세 (지은이)}                         => {무라카미 하루키 (지은이)}  0.2922338 0.5901374  0.4951962 1.210315 730
[11] {알베르 카뮈 (지은이)}                         => {조지 오웰 (지은이)}        0.2646117 0.6683519  0.3959167 1.657937 661
[12] {조지 오웰 (지은이)}                           => {알베르 카뮈 (지은이)}      0.2646117 0.6564052  0.4031225 1.657937 661
[13] {조지 오웰 (지은이)}                           => {무라카미 하루키 (지은이)}  0.2570056 0.6375372  0.4031225 1.307527 642
[14] {무라카미 하루키 (지은이)}                     => {조지 오웰 (지은이)}        0.2570056 0.5270936  0.4875901 1.307527 642
[15] {조지 오웰 (지은이)}                           => {히가시노 게이고 (지은이)}  0.2562050 0.6355511  0.4031225 1.242259 640
[16] {히가시노 게이고 (지은이)}                     => {조지 오웰 (지은이)}        0.2562050 0.5007825  0.5116093 1.242259 640
[17] {알베르 카뮈 (지은이)}                         => {무라카미 하루키 (지은이)}  0.2518014 0.6359960  0.3959167 1.304366 629
[18] {무라카미 하루키 (지은이)}                     => {알베르 카뮈 (지은이)}      0.2518014 0.5164204  0.4875901 1.304366 629
[19] {알베르 카뮈 (지은이)}                         => {히가시노 게이고 (지은이)}  0.2473979 0.6248736  0.3959167 1.221388 618
[20] {히가시노 게이고 (지은이)}                     => {알베르 카뮈 (지은이)}      0.2473979 0.4835681  0.5116093 1.221388 618
[21] {알베르 카뮈 (지은이), 조지 오웰 (지은이)}     => {헤르만 헤세 (지은이)}      0.2197758 0.8305598  0.2646117 1.677234 549
[22] {알베르 카뮈 (지은이), 헤르만 헤세 (지은이)}   => {조지 오웰 (지은이)}        0.2197758 0.7369128  0.2982386 1.828012 549
[23] {조지 오웰 (지은이), 헤르만 헤세 (지은이)}     => {알베르 카뮈 (지은이)}      0.2197758 0.7359249  0.2986389 1.858787 549
[24] {무라카미 하루키 (지은이), 히가시노 게이고 (지은이)} => {헤르만 헤세 (지은이)} 0.2077662 0.6775457 0.3066453 1.368237 519
[25] {무라카미 하루키 (지은이), 헤르만 헤세 (지은이)} => {히가시노 게이고 (지은이)} 0.2077662 0.7109589 0.2922338 1.389652 519
[26] {헤르만 헤세 (지은이), 히가시노 게이고 (지은이)} => {무라카미 하루키 (지은이)} 0.2077662 0.6910786 0.3006405 1.417335 519
[27] {베르나르 베르베르 (지은이)}                   => {히가시노 게이고 (지은이)}  0.2053643 0.7205056  0.2850280 1.408312 513
[28] {히가시노 게이고 (지은이)}                     => {베르나르 베르베르 (지은이)} 0.2053643 0.4014085  0.5116093 1.408312 513
[29] {조지 오웰 (지은이), 히가시노 게이고 (지은이)} => {헤르만 헤세 (지은이)}      0.2025620 0.7906250  0.2562050 1.596590 506
[30] {조지 오웰 (지은이), 헤르만 헤세 (지은이)}     => {히가시노 게이고 (지은이)}  0.2025620 0.6782842  0.2986389 1.325786 506
[31] {헤르만 헤세 (지은이), 히가시노 게이고 (지은이)} => {조지 오웰 (지은이)}      0.2025620 0.6737683  0.3006405 1.671374 506
[32] {무라카미 하루키 (지은이), 조지 오웰 (지은이)} => {헤르만 헤세 (지은이)}      0.2017614 0.7850467  0.2570056 1.585325 504
[33] {조지 오웰 (지은이), 헤르만 헤세 (지은이)}     => {무라카미 하루키 (지은이)}  0.2017614 0.6756032  0.2986389 1.385597 504
[34] {무라카미 하루키 (지은이), 헤르만 헤세 (지은이)} => {조지 오웰 (지은이)}      0.2017614 0.6904110  0.2922338 1.712658 504
```

▲작가 간의 연관성 데이터 분석 결과

다음은, Apriori rule을 통해 나온 데이터 분석 결과이다. 조지 오웰과 헤르만 헤세의 책을 읽을 시, 알베르 카뮈의 책을 읽을 향상도(lift)가 1.85에 달하며, 이 밖에도 위 세 작가의 상호작용에 있어서 높은 향상도가 관측된다는 점이 유의미하다. 또한, 다른 여러 작가들의 연관성도 발견된다.



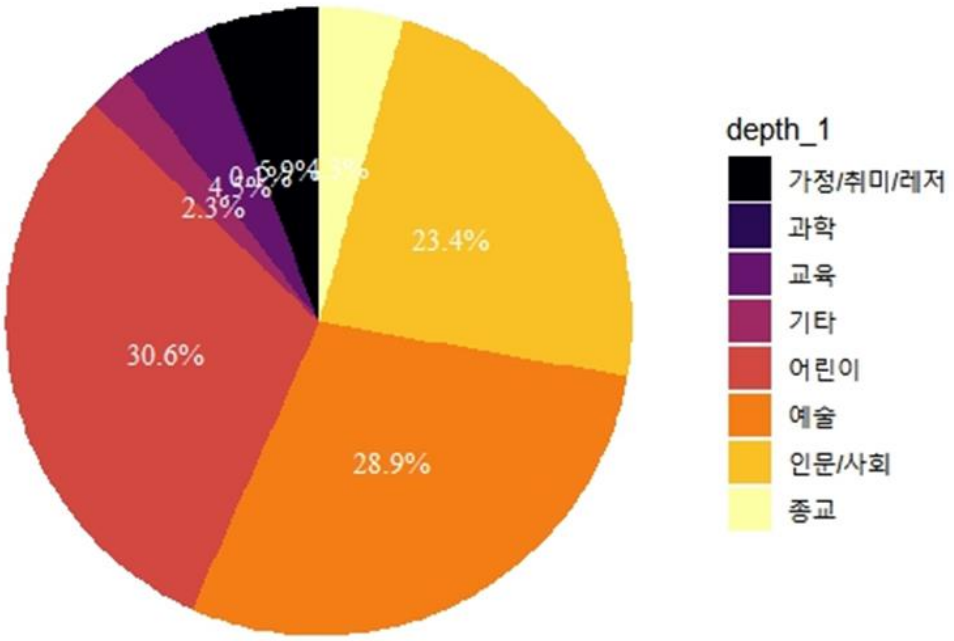
▲작가 간의 연관성 시각화 자료

위 그래프는 작가 간의 연관성에 대한 시각화 그래프이다. 지지도가 클수록 원형의 크기가 커지며, 향상도가 높을수록 해당 원의 색이 진한 빨간색에 가까워진다. 작가 간의 연관성을 바탕으로 이용자에게 추천 시스템을 제공할 수 있을 것이다.

리더스 앱 '내 서재'에서 이용자는 책을 선정해 [읽고 싶은/읽는 중/읽음/잠시 멈춘/중단] 5가지 상태 중 1개를 선택할 수 있으며, 만약 '읽음'으로 상태를 변경한 도서는 별점 부여도 가능하다. 이에 우리 팀은 5가지 독서 상태 중 '읽음(완독)'이 가장 중요하다고 판단 되었고, '완독'과 관련된 데이터 분석을 진행하여 관련 시스템을 제안하고자 한다.

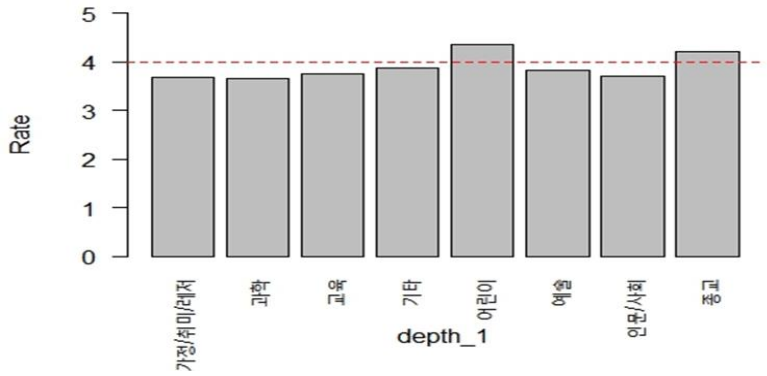
● 완독한 책 분석

유저들이 완독한 책 중 가장 평점이 높은 책을 구하기 위해 book_category, book, user_book 데이터를 결합하여 book_id, 완독 횟수, 평균 평점, 카테고리가 포함된 하나의 데이터를 만들었다.



▲완독 서적 중 평균이 4.0점 이상인 도서의 비율 파이 차트

우리가 실시한 데이터 분석 과정에서, 유저가 완독한 책 중 평점이 4.0점 이상인 도서를 모아보았다. 해당 조건을 충족하는 도서 중 가장 큰 비중을 차지하는 카테고리는 '어린이' 카테고리였다. '어린이' 카테고리는 30.6%를 차지하였고, 그 다음으로는 '예술' 카테고리가 28.9%로 높았으며, '인문/사회' 카테고리가 23.4%로 3위를 차지했다.



▲완독 서적 의 카테고리 별 평균 별점

다음으로 완독한 책들의 카테고리별 평균 평점을 구했다. 가장 눈에 띄는 점은 어린이 카테고리의 평점이 가장 높았으며 예술, 인문/사회, 어린이 카테고리 중 유일하게 어린이 카테고리 만이 평균 평점 4점을 넘겼다.

리더스 어플의 주 사용자가 직장인과 대학생임에도 불구하고, 해당 조건 하에서 어린이 도서 카테고리의 비중이 가장 크게 드러났다. 현재 리더스 어플리케이션의 북클럽을 살펴보면, 대부분 인문학과 투자서적에 할당되어 있다. 북클럽의 연령층을 낮추어 어린이 도서에 대한 북클럽을 개설하면 그에 대한 긍정적인 반응을 기대할 수 있을 것이라 생각한다.

하지만 돈을 내야하는 북클럽 특성상 어린이 도서에 대한 북클럽을 진행 할 시 어플에 수익성을 가져다 줄 수 있는지는 다른 영역이라고 생각하기 때문에 수익성 검토가 필요할 것으로 보인다.

4. 결론 및 제언

리더스 이용자가 독서를 시작한 시간을 기준으로, 데이터를 추출해 각 카테고리 빈도 분포를 살펴봤다. 가장 많이 이용하고 있는 상위 시간대는 13-15시였고, 가장 적게 이용하고 있는 하위 시간대는 18-20시였다. 기존에 '낮 시간대보다 모든 일과가 마무리 된 밤 시간대에 리더스, 혹은 독서를 시작할 것이다'라고 생각했던 가설을 채택할 수 없었다. 가설을 채택하지는 못했지만, 해당 데이터를 동일비율로 놓고, 상, 하위 3개의 범주 시간대에 따라 비교하니, 미묘하지만 인문사회와 예술 분야에서 차이가 보임을 확인할 수 있었다.

이어 살펴본 리더스 이용자의 스크랩 빈도에 따라 12-14시에 가장 활발한 스크랩을, 18-20시에 가장 저조한 스크랩 활동을 하는 것을 확인했다. 앞선 독서 시작 시간과 비슷한 시간대의 양상 추이를 보인다. 이에 우리는 궁금증을 갖고 Apriori 알고리즘으로 더욱 세부적인 분석을 통해 데이터를 추출했다.

그 결과 스크랩 시간이 가장 많았던 12-14시에 {가정/취미/레저}, {기타}를 스크랩한 유저가 {인문/사회}에 해당하는 책을 스크랩했으며, {예술}, {인문/사회}에서 {가정/취미/레저}로 이어지는 스크랩 유저들의 리프트 값이 가장 높음을 확인했다. 또한,12-14시와 가장 대조되는 시간, 즉 12시간 차이가 나는 0시-2시에도 12-14시와 같은 패턴을 보였다. 다만, 지지도 값을 기준으로 12-14시는 {가정/취미/레저}가 상위 룰에 5개가 분포돼 있으나 0-2시는 2개만 있다는 차이를 발견할 수 있었다. 마지막으로 스크랩 빈도가 가장 적었던 18-20시에는 관측치 자체가 적어 룰이 4개밖에 생성되지 않았으며, 이마저도 리프트 값은 1에 미치지 않았다. 즉, 시간대 별 스크랩하는 카테고리의 비중은 크게 의미 차이가 나타나지 않았다. 그러나 시간대별로 다양한 룰이 나왔다는 점에서 의의가 있었다.

더불어, 소비자가 읽는 책들의 저자와의 연관성을 파악하고자 이도 마찬가지로 Apriori를 통해 파악했다. 그 결과 {조지 오웰}과 {헤르만 헤세}의 책을 읽을 시, {알베르 카뮈}의 책을 읽을 리프트가 1.85에 달하며, 이 세 작가의 상호작용에서 높은 리프트 값을 관측할 수 있다는 점을 발견했다. 또한, 소비자가 살펴보는 책뿐만 아니라 그 책의 작가와의 연관성을 데이터 추출을 통한 유의미한 결과로 도출할 수 있었다.

마지막으로, 소비자가 리더스를 통해 책을 읽기 시작한 시간대, 스크랩한 시간대, 소비자가 읽는 책의 저자와의 연관성을 살펴본 후, 높은 평점을 보유한 완독한 책을 살펴봤다. 그 결과, 가장 큰 비중을 차지한 카테고리는 어린이> 예술> 인문/사회 순으로 높았다. 앞서 살펴봤을 때, 리더스의 주 소비자는 직장인과 대학생임에도 어른이 도서 카테고리의 비중이 높다는 점에서 의의가 있다.

리더스에 관한 소비자(유저) 정보, 스크랩한 시간대, 읽기 시작한 시간대, 저자의 정보, 완독률 및 평점 등 다양한 데이터의 활용으로 리더스 어플의 활용 방향에 대해 살펴볼 수 있었다. 기존 우리의 가설과는 달랐던 데이터 결과였지만, 그럼에도 이에 대한 의문점을 갖고 더 많은 데이터 활용을 도전해볼 수 있었고, Apriori 알고리즘의 실현을 통해, 여러 제안을 떠올려 제고해보기도 했다.

* 사용 코드

```r
#파일 불러오기
load("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/리더스 제공 데이터/01_user.RData.rdata")
load("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/리더스 제공 데이터/02_user_cat.RData.rdata")
load("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/리더스 제공 데이터/03_follow.RData.rdata")
load("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/리더스 제공 데이터/04_user_book.RData.rdata")
load("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/리더스 제공 데이터/05_book.RData.rdata")
load("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/리더스 제공 데이터/06_book_cat.RData.rdata")
library(readxl)
cat <- read_excel("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/2조 팀플/07_cat_modi.xlsx")
load("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/리더스 제공 데이터/08_scrap.RData.rdata")

#user_na 불러오기(gender의 na까지 모두 제거)
user_na <- read.csv("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/2조 팀플/user_na.csv", header = T)

#완독기준으로 유저의 직업,성별 시각화#

#user,user_book을 유저를 기준으로 하나의 데이터로 통합
userdata <- merge (user_book, user_na, by='user_id')

#완독한 상태의 데이터만 추출
userdata %>% select (user_id, gender, 직업, read_status) %>% filter(read_status=='READ_STATUS_DONE') -> status_done

#중복 제거
status_done <- unique(status_done)
table(status_done$`직업`)

#write.csv(user_na, file = "C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/2조 팀플/user_na.csv")



#시각화##완독 수가 직업별로, 성별로 어떤 차이가 있는지 파악### -> 기존 데이터와 차이가 없음


#보고서에 사용된 그래프 코드
#user_na는 user 데이터에서 결측치가 제거된 것


#1. 전체 유저의 직업 분포
ggplot(user_na, aes(x='user_id', fill=직업))+geom_bar(position = 'fill')

#2. 완독 유저의 직업 분포
ggplot(status_done, aes(x='user_id', fill=직업))+geom_bar(position = 'fill')

#3. 전체 유저의 성별 분포
ggplot(user_na, aes(x='user_id', fill=gender))+geom_bar(position = 'fill')
```

▲ 전체/완독 유저의 직업/성별 그래프 시각화

```r
#book_cat(6) + cat(7) 합치기
cat_ <- subset(cat, select = -c(name, depth_2, depth_3, depth_4, depth_5))
book_cat_1 <- merge(book_cat, cat_, by='book_category_id', all.x=TRUE)
nrow(unique(book_cat_1))

#크게 분류되지 않은 카테고리는 NA로 나타남
#결측치 제거
book_cat_1 <- na.omit(book_cat_1)
colSums(is.na(book_cat_1))
nrow(unique(book_cat_1))


book_cat_1 <- subset(book_cat_1, select = -c(book_category_id))
book_cat_1 <- unique(book_cat_1)


#book_cat_1 + scrap(8)
scrap_book_cat <- merge(scrap, book_cat_1, by='book_id', all.x=TRUE)
colSums(is.na(scrap_book_cat))
scrap_book_cat



#버리기
#merge 함수로 시간대(읽기 시작한 시간으로 분류)
book_inform <- merge(user_book, scrap_book_cat, by='user_id', all=TRUE)
book_inform

#완독한 시간 추출
#read_ing 완료한 것 추출하기
B1 <- book_inform %>% filter(read_status == 'READ_STATUS_ING')

B1

install.packages("anytime")
library(anytime)
library(lubridate) #lubridate 시간정보추출하기
library(tidyverse)

B1$datetime <- as_datetime(B1$modified_at)
B1$datetime

B1$yyyymmdd <- format(B1$datetime, "%y%m%d") #yyyymmdd: datetime에서 연월일(YYYYMMDD)을 추출한 값
B1$yyyymmdd

#시간만 추출한 값(분은 필요 없을 듯)
B1$hh <- format(B1$datetime, "%H") #hh: datetime에서 시간HH)을 추출한 값
B1$hh

library(dplyr)
B1 <- B1 %>% filter(hh == 13|hh == 14|hh == 15)
B1
```

```r
########################################
#기존 주제 시각화하기 : 시간대별 데이터 추출
library(ggplot2)
library(dplyr)
library(plyr)
library(tidyverse)


##책읽기 시작한 상위 시간 13시부터 15시 사이의 분포 확인
###C1은 우리 팀 이 설정한 카테고리와 책을 읽기 시작한 상위 3개의 시간을 추출해 합친 것
C1_1 <- subset(C1, select = c(hh, depth_1))
C1_1 <- na.omit(C1_1)
G1 <- ggplot(data=C1_1, aes(x=hh, fill=depth_1)) + geom_bar()
G1


##동일한 비율 선상으로 확인하는 '책읽기 시작한 상위 시간대 별 카테고리'
C1_1 <- subset(C1, select = c(hh, depth_1))
C1_1 <- na.omit(C1_1)
G1 <- ggplot(data=C1_1, mapping = aes(x=hh, fill=depth_1)) + geom_bar(position = "fill") +
  ggtitle("책읽기 시작한 상위 시간대 별 카테고리")
G1



##책읽기 시작한 하위 시간 19시부터 21시 사이의 분포 확인
###C2는 우리 팀 이 설정한 카테고리와 책을 읽기 시작한 하위 3개의 시간을 추출해 합친 것
C2_1 <- subset(C2, select = c(hh, depth_1))
C2_1 <- na.omit(C2_1)
G2 <- ggplot(data=C2_1, aes(x=hh, fill=depth_1)) + geom_bar()
G2


##동일한 비율 선상으로 확인하는 '책읽기 시작한 하위 시간대 별 카테고리'
C2_1 <- subset(C2, select = c(hh, depth_1))
C2_1 <- na.omit(C2_1)
G2 <- ggplot(data=C2_1, mapping = aes(x=hh, fill=depth_1)) + geom_bar(position = "fill") +
  ggtitle("책읽기 시작한 하위 시간대 별 카테고리")
G2
```

▲리더스 이용자의 '독서 시작(READ_STATUS_ING)' 관련 코드

```r
#파일 불러오기
load("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/리더스 제공 데이터/01_user.RData.rdata")
load("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/리더스 제공 데이터/02_user_cat.RData.rdata")
load("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/리더스 제공 데이터/03_follow.RData.rdata")
load("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/리더스 제공 데이터/04_user_book.RData.rdata")
load("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/리더스 제공 데이터/05_book.RData.rdata")
load("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/리더스 제공 데이터/06_book_cat.RData.rdata")
library(readxl)
cat <- read_excel("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/2조 팀플/07_cat_modi.xlsx")
load("C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/리더스 제공 데이터/08_scrap.RData.rdata")


#시간대에 따라 스크랩하는 책의 변화를 살펴보기 위해

#book_cat(6) + cat(7) 합치기
cat_1 <- subset(cat, select = -c(name, depth_2, depth_3, depth_4, depth_5))
cat_2 <- merge(book_cat, cat_1, by='book_category_id', all.x=TRUE)
nrow(unique(cat_2))

#크게 분류되지 않은 카테고리는 NA로 나타남
#결측치 제거
cat_3 <- na.omit(cat_2)
colSums(is.na(cat_3))
nrow(unique(cat_3))

cat_4 <- subset(cat_3, select = -c(book_category_id))
cat_4 <- unique(cat_4)
```

```r
#book_cat_1 + scrap(8)
scrap_book_cat <- merge(scrap, cat_4, by='book_id', all.x=TRUE)




#결측치 제거
colSums(is.na(scrap_book_cat))
scrap_book_cat <- subset(scrap_book_cat, select = -c(page))
scrap_book_cat <- na.omit(scrap_book_cat)
colSums(is.na(scrap_book_cat))



#created_at이라는 변수를 날짜, 시간으로 분리
library(lubridate)
scrap_book_cat$datetime <- as_datetime(scrap_book_cat$created_at) #datetime: 일반 날짜 형식으로 변환
scrap_book_cat$datetime
scrap_book_cat$yyyymmdd <- format(scrap_book_cat$datetime, "%y%m%d") #yyyymmdd: datetime에서 연월일(YYYYMMDD)을 추출한 값
scrap_book_cat$yyyymmdd
scrap_book_cat$hhmm <- format(scrap_book_cat$datetime, "%H%S") #hhmm: datetime에서 시분(HHMM)을 추출한 값
scrap_book_cat$hhmm

scrap_book_cat$hour <- hour(scrap_book_cat$datetime)
scrap_book_cat$minute <- minute(scrap_book_cat$datetime)
```

```r
#시계열 그래프 도출
library(ggplot2)
ggplot(scrap_book_cat, aes(hour, fill=depth_1))+geom_bar()+xlab("Scrap time")+ylab("Scrap")+theme_bw()
#12~14시 스크랩이 가장 많고 #18~20시 가장 적음


scrap_content <- subset(scrap_book_cat, select = c(content, depth_1))
scrap_content

#스크랩 문구 + 카테고리만 csv랑 Rdata로 저장
write.csv(scrap_content, file = "C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/2조 팀플/scrap_content.csv")
save(scrap_content, file = "C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/2조 팀플/scrap_content.Rdata")


#csv 저장
#write.csv(scrap_book_cat, "C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/2조 팀플/scrap_book_cat.csv")

#Rdata 저장
#save(scrap_book_cat, file = "C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/2조 팀플/scrap_book_cat.Rdata")
```

```r
##########################################################################################################################################
#시간대 별로 카테고리 패턴 분석

scrap_cat_final <- subset(scrap_book_cat, select=-c(book_id, content, created_at, datetime, yyyymmdd, hhmm, minute))

scrap_cat_final

write.csv(scrap_cat_final, "C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/2조 팀플/scrap_cat_final.csv")
colSums(is.na(scrap_cat_final))


library(dplyr)

hour_0 <- scrap_cat_final %>% filter(hour == 0)
hour_0 <- hour_0 %>% group_by(user_id) %>% summarise(category = paste(depth_1, collapse = ","))
write.csv(hour_0, "C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/2조 팀플/hour_0.csv")
```

▲스크랩 시간대에 따른 이용자의 카테고리 패턴 분석 코드(1)

```r
#Market Basket Analysis 분석
#시간대 별로 보기 위해 그룹 생성
`hour_0-2` <- scrap_cat_final %>% filter(hour == 0|hour == 1|hour == 2)
`hour_12-14` <- scrap_cat_final %>% filter(hour == 12|hour == 13|hour == 14)
`hour_18-20` <- scrap_cat_final %>% filter(hour == 18|hour == 19|hour == 20)


`hour_0-2` <- subset(`hour_0-2`, select=-c(hour))
`hour_12-14` <- subset(`hour_12-14`, select=-c(hour))
`hour_18-20` <- subset(`hour_18-20`, select=-c(hour))



#카테고리 합치기
`hour_0-2` <- `hour_0-2` %>% group_by(user_id) %>% summarise(category = paste(depth_1, collapse = ","))
`hour_12-14` <- `hour_12-14` %>% group_by(user_id) %>% summarise(category = paste(depth_1, collapse = ","))
`hour_18-20` <- `hour_18-20` %>% group_by(user_id) %>% summarise(category = paste(depth_1, collapse = ","))
write.csv(`hour_0-2`, "C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/2조 팀플/hour_0-2.csv")
write.csv(`hour_12-14`, "C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/2조 팀플/hour_12-14.csv")
write.csv(`hour_18-20`, "C:/Users/kse62/Desktop/한양대학교/3학년 1학기/데이터예측모델과기계학습의응용/2조 팀플/hour_18-20.csv")
```

```r
#0-2시 카테고리
write.csv(`hour_0-2`,"hour_0-2.csv", quote = FALSE, row.names = TRUE)
txn = read.transactions(file="hour_0-2.csv",
                        rm.duplicates= TRUE,
                        format="basket",sep=",",cols=1);

`basket_rules_0-2` <- apriori(txn,
                          parameter = list(minlen=2,
                                           sup = 0.2,
                                           conf = 0.1,
                                           target="rules"))

summary(`basket_rules_0-2`)
inspect(sort(`basket_rules_0-2`, by="lift"))

library(arulesViz)
plot(`basket_rules_0-2`)
plot(`basket_rules_0-2`, method = "graph", control = list(type="items"))

# grouped matrix for association rules
plot(sort(`basket_rules_0-2`, by = "lift"), method = "grouped")



library(igraph)
#12-14시 카테고리
write.csv(`hour_12-14`,"hour_12-14.csv", quote = FALSE, row.names = TRUE)
txn_1 = read.transactions(file="hour_12-14.csv",
                        rm.duplicates= TRUE,
                        format="basket",sep=",",cols=1);

`basket_rules_12-14` <- apriori(txn_1,
                        parameter = list(minlen=2,
                                         sup = 0.2,
                                         conf = 0.1,
                                         target="rules"))

summary(`basket_rules_12-14`)
inspect(sort(`basket_rules_12-14`, by="lift"))

plot(`basket_rules_12-14`)
plot(`basket_rules_12-14`, method = "graph", control = list(type="itemsets"))

# grouped matrix for association rules
plot(sort(`basket_rules_12-14`, by = "lift")[1:20], method = "grouped")
```

```r
#18-20시 카테고리
write.csv(`hour_18-20`,"hour_18-20.csv", quote = FALSE, row.names = TRUE)
txn_2 = read.transactions(file="hour_18-20.csv",
                          rm.duplicates= TRUE,
                          format="basket",sep=",",cols=1);

`basket_rules_18-20` <- apriori(txn_2,
                          parameter = list(minlen=2,
                                           sup = 0.2,
                                           conf = 0.1,
                                           target="rules"))

summary(`basket_rules_18-20`)
inspect(sort(`basket_rules_18-20`, by="lift"))

library(arulesViz)
plot(`basket_rules_18-20`)
plot(`basket_rules_18-20`, method = "graph", control = list(type="itemsets"))
# grouped matrix for association rules
plot(sort(`basket_rules_18-20`, by = "lift"), method = "grouped")
```

▲스크랩 시간대에 따른 이용자의 카테고리 패턴 분석 코드(2)

```r
#데이터 불러오기
load("C:/Users/82104/Desktop/readers/04_user_book.RData.rdata")
load("C:/Users/82104/Desktop/readers/05_book.RData.rdata")
View(user_book)
View(book)

#라이브러리 설정
library(dplyr)
library(widyr)
library(tidyr)
library(arules)
library(arulesViz)

#지은이와 책 id만 빼서 b1이라 명명
b1 <- book %>% select(author, id)
b2 <- b1 %>%
  separate(author, sep= ",", into = c("writer", "translator"))
b2<- b2 %>% select(writer, id)
b1 <- b2

#등록되어 있는 책 중 작가가 가장 많이 언급빈도 데이터 추출, b2라 명명
b2 <- table(b1$writer)
View(b2)

#상위 30명의 작가가 쓴 책 추출, w1~30이라 명명
w1 <- b1 %>% filter(writer == "히가시노 게이고 (지은이)")
w2 <- b1 %>% filter(writer == "Roderick Hunt (지은이)")
w3 <- b1 %>% filter(writer == "무라카미 하루키 (지은이)")
w4 <- b1 %>% filter(writer == "헤르만 헤세 (지은이)")
w5 <- b1 %>% filter(writer == "메리 폽 어즈번 (지은이)")
w6 <- b1 %>% filter(writer == "박현숙 (지은이)")
w7 <- b1 %>% filter(writer == "J.K. 롤링 (지은이)")
w8 <- b1 %>% filter(writer == "레프 니콜라예비치 톨스토이 (지은이)")
w9 <- b1 %>% filter(writer == "조정래 (지은이)")
w10 <- b1 %>% filter(writer == "변승우 (지은이)")
w11 <- b1 %>% filter(writer == "윌리엄 셰익스피어 (지은이)")
w12 <- b1 %>% filter(writer == "애거사 크리스티 (지은이)")
w13 <- b1 %>% filter(writer == "박완서 (지은이)")
w14 <- b1 %>% filter(writer == "아서 코난 도일 (지은이)")
w15 <- b1 %>% filter(writer == "고정욱 (지은이)")
w16 <- b1 %>% filter(writer == "이문열 (지은이)")
w17 <- b1 %>% filter(writer == "데일 카네기 (지은이)")
w18 <- b1 %>% filter(writer == "앙투안 드 생텍쥐페리 (지은이)")
w19 <- b1 %>% filter(writer == "조지 오웰 (지은이)")
w20 <- b1 %>% filter(writer == "히로시마 레이코 (지은이)")
w21 <- b1 %>% filter(writer == "서지원 (지은이)")
w22 <- b1 %>% filter(writer == "로저 하그리브스 (지은이)")
w23 <- b1 %>% filter(writer == "앤서니 브라운 (지은이)")
w24 <- b1 %>% filter(writer == "C. S. 루이스 (지은이)")
w25 <- b1 %>% filter(writer == "사이토 다카시 (지은이)")
w26 <- b1 %>% filter(writer == "표도르 도스토예프스키 (지은이)")
w27 <- b1 %>% filter(writer == "송도수 (지은이)")
w28 <- b1 %>% filter(writer == "베르나르 베르베르 (지은이)")
w29 <- b1 %>% filter(writer == "알베르 카뮈 (지은이)")
w30 <- b1 %>% filter(writer == "미야베 미유키 (지은이)")

#rate, modified_at 삭제
user_book <- user_book %>% select(user_id, book_id, read_status)
#read_status_stop 삭제
user_book <- user_book %>% filter(read_status != "read_status_stop")
View(user_book)

#user_book_new의 book_id를 id로 수정
names(user_book) <- c("user_id", "id", "read_status")
#w1~w30까지 합쳐서 w1_30이라 명명명
w1_30 <- rbind(w1, w2, w3, w4, w5, w6, w7, w8, w9, w10, w11, w12, w13, w14, w15,
               w16, w17, w18, w19, w20, w21, w22, w23, w24, w25, w26, w27, w28, w29, w30)

#상위 30개 작가의 책을 읽은 기록(독서 중단 제외)
user_book <- inner_join(user_book, w1_30, by='id')
#user_book에서 read_status 삭제
user_book <- user_book %>%  select(user_id, id, writer)

user_book <- user_book %>% select(user_id, writer)
user_book <- user_book %>% group_by(user_id) %>% summarise(writer_name = paste(writer, collapse = ","))
View(user_book)

#결측값 확인
sum(is.na(user_book))
str(user_book)

write.csv(user_book,"user_book.csv", quote = FALSE, row.names = TRUE)
txn = read.transactions(file="user_book.csv",
                        rm.duplicates= TRUE,
                        format="basket",sep=",",cols=1);

basket_rules <- apriori(txn,
                        parameter = list(minlen=2,
                                         sup = 0.2,
                                         conf = 0.1,
                                         target="rules"))

summary(basket_rules)
inspect(basket_rules)
```

▲ 저자 간의 연관성 분석

```r
library(dplyr)
library(readr)
library(plyr)
library(ggplot2)

install.packages('readxl')
library(readxl)

##cat 데이터 수정
cat <- read_excel("C:/Users/영/Desktop/DB/07_cat_modi.xlsx")

## book 데이터 열이름 맞추기 -> book_id
B1 <- dplyr::rename(book, book_id=id)
B2 <- select(B1, book_id, title)

## book데이터와 book_category데이터 이너조인하기
C1 <- inner_join(x=book_cat, y=cat, by="book_category_id")
C1
C2 <- inner_join(x=C1, y=B2, by="book_id")
C2
TITLE <-distinct(C2, book_id, .keep_all = TRUE)
TITLE

##완독한 책의 RATE 평균 4이상 책들 구하기
detach("package:plyr", unload=TRUE)
F1 <- filter(user_book, read_status == "READ_STATUS_DONE")
F2 <- group_by(F1, book_id)
class(F2)
#book_id별 완독 횟수
I1 <- count(F2, book_id)
I2 <- arrange(I1, desc(n))
I2
#RATE 평균 계산
F3 <- summarise(F2,
                Rate = mean(rate, na.rm=TRUE))
F4 <- filter(F3, Rate >= 4)
F5 <- arrange(F4, desc(Rate))

##Rate 4.5 이상, 책들 book_id, Rate, 카테고리, 제목을 완독 완독 횟수 순으로 정렬
RATE <- inner_join(x=I2, y=F5, by="book_id")
Good_book <- inner_join(x=RATE, y=TITLE, by="book_id")
Good_book
```

▲완독 서적 분석을 위한 데이터 전처리

```r
##Good_book 데이터로 카테고리별 비중을 나타낸 파이차트 만들기
P1 <- ungroup(Good_book)
P2 <- select(P1, book_id, depth_1)
P3 <- count(P2, depth_1)
P4 <- arrange(P3, desc(n))

#비중을 pct(퍼센트로 나타냄)
Good_book_pct <- mutate(P4, pct=n/sum(n)*100)
Good_book_pct


library(tidyverse)
library(viridis)

ggplot(Good_book_pct, aes(x='', y=pct, fill=depth_1))+
  geom_bar(width = 1, stat='identity')+
  theme_void()+
  coord_polar('y', start=0)+
  geom_text(aes(label=paste0(round(pct,1), '%')),
            position=position_stack(vjust=0.5),
            color='white', family='serif', size=4)+
  scale_fill_viridis(option='inferno', discrete=TRUE)
```

▲완독 서적 중 평균이 4.0점 이상인 도서의 비율 파이 차트

```
## 카테고리별 평균평점 bar plot(막대그래프)
#평균 평점 4점 이상인 책 뿐만 아니라 모든 책들을 기준으로 나타냄
RATE_R <- inner_join(x=I2, y=F3, by="book_id")
Good_book_R <- inner_join(x=RATE_R, y=TITLE, by="book_id")
Good_book_R

#카테고리별 평균 Rate
R1 <- ungroup(Good_book_R)
R2 <- select(R1, Rate, depth_1)
R3 <- aggregate(Rate ~ depth_1, R2, mean)

#카테고리별 평균 Rate 막대 그래프
Good_book_barplot <- barplot(Rate~depth_1,R3,
                     ylim=c(0,5),
                     las=2)+
                     abline(h=4, col="red", lty=2, lwd=1)
Good_book_barplot
```

▲완독 서적 의 카테고리 별 평균 별점(막대그래프)