

모두를 위한 R 데이터 분석 입문

2판



Chapter 10

워드클라우드와 구매 패턴 분석



목차

1. 워드클라우드 분석
2. 구매 패턴 분석
3. 인터넷 검색어 분석
4. 공공 빅데이터

Section 01

워드클라우드 분석

1. 워드클라우드 분석

1. 워드클라우드의 개념

- 지금까지 숫자 형태의 데이터를 다루는 방법에 관하여 학습
- 분석 대상 데이터 중에는 숫자가 아닌 문자나 문장 형태의 데이터도 있음
ex) 이메일 내용이나 SNS 메시지, 댓글
- 워드클라우드(word cloud)는 문자형 데이터를 분석하는 대표적인 방법으로, 대상 데이터에서 단어(주로 명사)를 추출하고 단어들의 출현 빈도수를 계산하여 시각화하는 기능
- 출현 빈도수가 높은 단어는 그만큼 중요하거나 관심도가 높다는 것을 의미



그림 10-1 워드클라우드의 예

1. 워드클라우드 분석

2.1 워드클라우드 문서 파일 준비

- 워드클라우드를 작성할 대상 문서는 일반적으로 텍스트 파일 형태로 준비
- 파일의 끝부분 처리를 [그림 10-2]와 같이 마지막 문장이 끝나면 반드시 줄 바꿈을 한 후 저장
- 파일을 저장할 때, [다른 이름으로 저장]을 선택하고 [그림 10-3]과 같이 인코딩을 'UTF-8'로 선택을 하여 저장
- 파일 이름이나 파일이 저장된 폴더 경로에 한글이 포함되어 있으면 파일을 읽을 때 에러가 발생하는 경우가 있으므로 파일을 저장할 때는 파일 이름을 영어로 설정

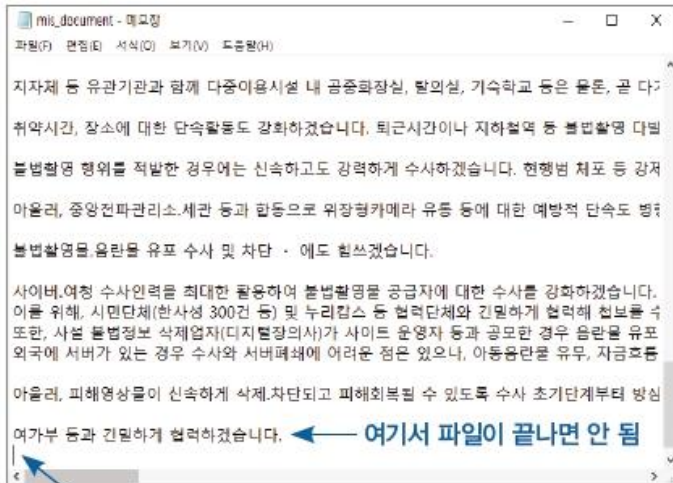


그림 10-2 텍스트 파일 끝부분에서의 줄바꿈

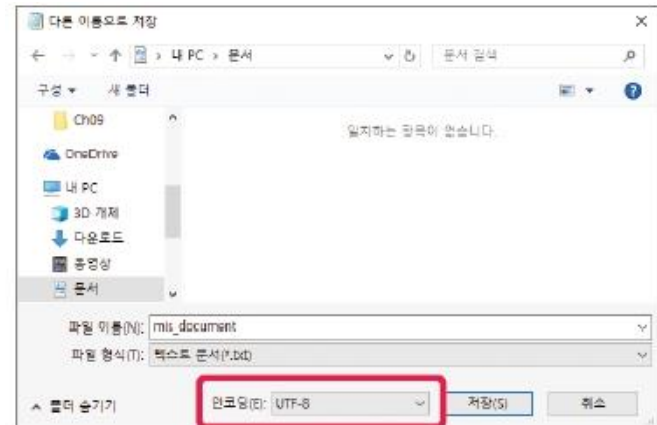


그림 10-3 UTF-8로 텍스트 파일 저장

1. 워드클라우드 분석

3. 워드클라우드의 작성

3.1 대국민 담화문의 명사 추출하기

코드 10-1

```
library(wordcloud)           # 워드클라우드
library(KoNLP)               # 한국어 처리
library(RColorBrewer)        # 색상 선택

setwd("D:/source")
text <- readLines("mis_document.txt", encoding = "UTF-8" ) # 파일 읽기
buildDictionary(ext_dic = "woorimalsam") # '우리말샘' 한글사전 로딩
pal2 <- brewer.pal(8, "Dark2")          # 팔레트 생성
noun <- sapply(text,extractNoun, USE.NAMES=F) # 명사 추출
noun                                     # 추출된 명사 출력
```

```
> library(wordcloud)           # 워드클라우드
> library(KoNLP)               # 한국어 처리
> library(RColorBrewer)        # 색상 선택
```

1. 워드클라우드 분석

```
> setwd("D:/source")
> text <- readLines("mis_document.txt", encoding = "UTF-8" )    # 파일 읽기
> buildDictionary(ext_dic = "woorimalsam")                        # '우리말샘' 한글사전 로딩
> pal2 <- brewer.pal(8, "Dark2")                                  # 팔레트 생성
> noun <- sapply(text, extractNoun, USE.NAMES=F)                  # 명사 추출
> noun                                                            # 추출된 명사 출력

[[1]]
[1] "1"      "여성"   "가족"   "부"     "불법"   "촬영"   "등"     "디지털"
[9] "성범죄" "안전"   "한"     "사회"   "국민"   "들"     "말"

[[2]]
[1] ""

[[3]]
[1] "존경"   "하"     "국민"   "여러분"

[[4]]
[1] "여성"   "가족"   "부"     "장관"   "정현"   "백"
[7] "우리사회" "들불"   "미"     "투"     "운동"   "계기"
[13] "일상"   "화"     "폭력"   "차별"   "맞선"   "여성"
...(이하 생략)
```


1. 워드클라우드 분석

3.2 빈도수 높은 단어를 막대그래프로 작성하기

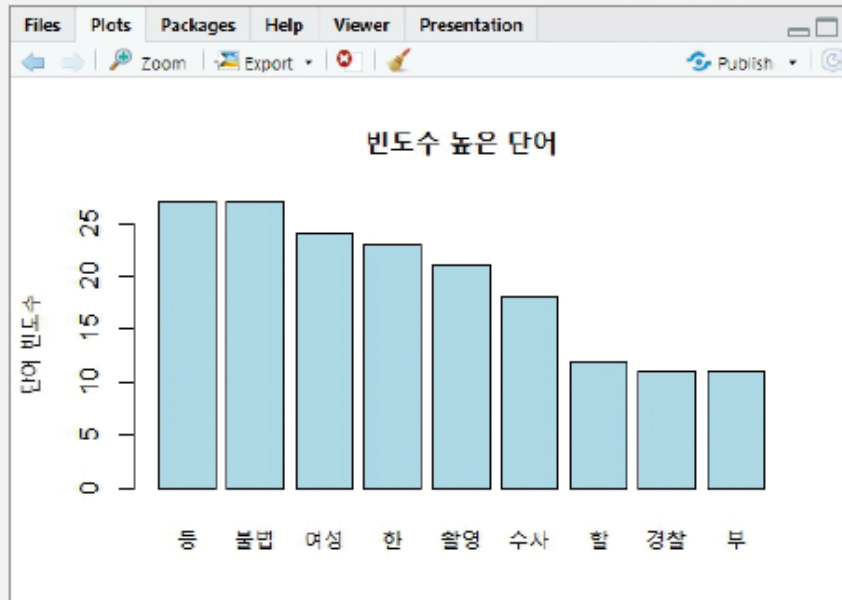
코드 10-2

```
noun2 <- unlist(noun)           # 추출된 명사 통합
wordcount <- table(noun2)       # 단어 빈도수 계산
temp <- sort(wordcount, decreasing=T)[1:10] # 빈도수 높은 단어 10개 추출
temp
temp <- temp[-1]               # 공백 단어 제거
barplot(temp,                  # 막대그래프 작성
          names.arg = names(temp), # 막대 이름을 단어로 표시
          col = "lightblue",      # 막대의 색상 지정
          main = "빈도수 높은 단어", ylab = "단어 빈도수")
```

```
> noun2 <- unlist(noun)           # 추출된 명사 통합
> wordcount <- table(noun2)       # 단어 빈도수 계산
> temp <- sort(wordcount, decreasing=T)[1:10] # 빈도수 높은 단어 10개 추출
> temp
noun2
      등  불법  여성   한  촬영  수사   할  경찰  성범죄
42    27   27   24   23   21   18   12   11    11
> temp <- temp[-1]               # 공백 단어 제거
```

1. 워드클라우드 분석

```
> barplot(temp,                                     # 막대그래프 작성
+   names.arg = names(temp),                        # 막대 이름을 단어로 표시
+   col = "lightblue",                             # 막대의 색상 지정
+   main = "빈도수 높은 단어", ylab = "단어 빈도수")
>
```



1. 워드클라우드 분석

3.3 워드클라우드 작성하기

코드 10-3

```
wordcloud(names(wordcount),  
           freq=wordcount,  
           scale=c(6,0.7),  
           min.freq=3,  
           random.order=F,  
           rot.per=.1,  
           colors=pal2)  # 단어들  
                          # 단어들의 빈도  
                          # 단어의 폰트 크기  
                          # 단어의 최소 빈도  
                          # 단어의 출력 위치  
                          # 90도 회전 단어 비율  
                          # 단어의 색
```

1. 워드클라우드 분석



1. 워드클라우드 분석

- `names(wordcount)`

워드클라우드 상에 표시할 단어를 지정한다.

- `freq=wordcount`

워드클라우드 상에 표시할 단어의 빈도수를 지정한다.

- `scale=c(6,0.7)`

표시할 단어의 폰트 크기를 지정한다. 여기서 6은 폰트의 최대 크기, 0.7은 폰트의 최소 크기를 의미한다.

- `min.freq=3`

빈도수가 3 이상인 단어들만 표시한다.

- `random.order=F`

단어가 표시될 위치를 지정한다. T는 단어의 표시 위치를 무작위로 지정할 수 있고, F는 빈도수가 높은 단어 일수록 중앙쪽에 배치된다.

- `rot.per=.1`

단어를 표시할 때 세로 방향으로 표시할 단어의 비율을 지정한다. 여기서 .1은 10%를 의미한다.

- `colors=pal2`

빈도수에 따라 pal2에 있는 색으로 단어의 색을 지정한다.

1. 워드클라우드 분석

3.4 워드클라우드 수정하기

코드 10-4

```
# 빈도수 높은데 워드클라우드에 없으면 사용자 사전에 추가
buildDictionary(ext_dic = "woorimalsam",
               user_dic=data.frame("정치", "ncn"),
               replace_usr_dic = T)
noun <- sapply(text,extractNoun, USE.NAMES=F)
noun2 <- unlist(noun)                # 추출된 명사 통합
# 무의미한 단어 제거
noun2 <- noun2[nchar(noun2)>1]       # 1글자 단어 제거
noun2 <- gsub("하지","", noun2)     # '하지' 제거
noun2 <- gsub("때문","", noun2)     # '때문' 제거
wordcount <- table(noun2)           # 단어 빈도수 계산
wordcloud(names(wordcount),
          freq=wordcount,
          scale=c(6,0.7),
          min.freq=3,
          random.order=F,
          rot.per=.1,
          colors=pal2)
```

1. 워드클라우드 분석

```
> # 빈도수 높은데 워드클라우드에 없으면 사용자 사전에 추가
> buildDictionary(ext_dic = "woorimalsam",
+               user_dic=data.frame("정치", "ncn"),
+               replace_usr_dic = T)
629898 words dictionary was built.
> # 무의미한 단어 제거
> noun2 <- noun2[nchar(noun2)>1]      # 1글자 단어 제거
> noun2 <- gsub("하지","", noun2)    # '하지' 제거
> noun2 <- gsub("때문","", noun2)    # '때문' 제거
> wordcount <- table(noun2)          # 단어 빈도수 계산
> wordcloud(names(wordcount),
+          freq=wordcount,
+          scale=c(6,0.7),
+          min.freq=3,
+          random.order=F,
+          rot.per=.1,
+          colors=pal2)
```

1. 워드클라우드 분석



Section 02

구매 패턴 분석

2. 구매 패턴 분석

- 상품의 유통, 판매 분야는 데이터 분석이 활발히 적용되는 분야중의 하나
- 계산대 부근에 껌이나 캔디류, 건전지 등이 진열되어 있는 것은 우연이 아니고 소비자의 구매 행태에 대한 철저한 분석의 결과
- 소비자의 구매 패턴(행태) 분석은 장바구니 분석(market basket analysis)으로도 알려져 있음



(이미지 출처: <https://pixabay.com/>)

2. 구매 패턴 분석

1. 연관 규칙

- **연관 규칙(association rule)** : 데이터 안에 포함된 일정한 패턴
- 구매 데이터에서 찾을 수 있는 연관 규칙의 예

“맥주를 사는 사람은 땅콩도 함께 구매한다”

“분유를 사는 사람은 기저귀도 함께 구매한다”

- 구매 패턴의 표현

{맥주} → {땅콩}

{분유} → {기저귀}

- 구매 패턴은 영수증을 분석하면 알 수 있다.

2. 구매 패턴 분석

2. 어프리오리 알고리즘

- **어프리오리(Apriori) 알고리즘:** 연관규칙 분석에 널리 이용되는 머신러닝 기법중의 하나로, 1994년 Agrawal 와 Srikant에 의해 제안됨.
- **구매 행렬:** 구매 내역에서 연관된 구매 상품을 찾는 가장 쉬운 방법

표 10-1 구매 내역 예제

거래 번호	구매 상품
1	맥주, 땅콩
2	맥주, 오징어
3	맥주, 라면, 땅콩
4	초콜릿, 껌
5	초콜릿, 생수, 껌

표 10-2 구매 행렬

상품	맥주	땅콩	오징어	라면	초콜릿	껌	생수
맥주	-	2	1	1	0	0	0
땅콩	2	-	1	0	0	0	0
오징어	1	1	-	0	0	0	0
라면	1	0	0	-	0	0	0
초콜릿	0	0	0	0	-	2	1
껌	0	0	0	0	2	-	1
생수	0	0	0	0	1	1	-

2. 구매 패턴 분석

▪ 지지도(support):

상품 X,Y를 함께 구매한 비율이 전체 거래에서 차지하는 비율을 측정하는 척도

- $\text{support}(X \rightarrow Y)$, $\text{support}(Y \rightarrow X)$, $\text{support}(X, Y)$ 모두 같은 의미

$$\text{support}(\{X\} \rightarrow \{Y\}) = \frac{X, Y \text{를 함께 포함한 거래건수}}{\text{전체 거래건수}}$$

▪ {맥주}→{땅콩}의 지지도

$$\text{support}(\{\text{맥주}\} \rightarrow \{\text{땅콩}\}) = \frac{2}{5} = 0.4$$

표 10-1 구매 내역 예제

거래 번호	구매 상품
1	맥주, 땅콩
2	맥주, 오징어
3	맥주, 라면, 땅콩
4	초콜릿, 껌
5	초콜릿, 생수, 껌

2. 구매 패턴 분석

▪ 신뢰도(confidence): 조건부확률을 의미

상품 X를 구매했다는 전제하에 상품 X와 Y를 동시에 구매한 빈도수를 계산하는 척도

$$cconfidence(\{X\} \rightarrow \{Y\}) = \frac{X, Y를 포함한 거래건수}{X를 포함한 거래건수}$$

▪ {맥주} → {땅콩}의 신뢰도

맥주와 땅콩 거래 건수

$$confidence(\{맥주\} \rightarrow \{땅콩\}) = \frac{2}{3} = 0.67$$

맥주 거래 건수

▪ {땅콩} → {맥주}의 신뢰도

$$confidence(\{땅콩\} \rightarrow \{맥주\}) = \frac{2}{2} = 1$$

땅콩 거래 건수

표 10-1 구매 내역 예제

거래 번호	구매 상품
1	맥주, 땅콩
2	맥주, 오징어
3	맥주, 라면, 땅콩
4	초콜릿, 껌
5	초콜릿, 생수, 껌

맥주를 산 경우에는 '많은 경우' 땅콩도 함께 사지만,
땅콩을 산 경우는 '반드시' 맥주를 함께 산다

2. 구매 패턴 분석

X를 구매한 사람이 Y를 구매할 확률과
X의 구매와 상관없이 Y를 구매할 확률의 비

$cconfidence(\{X\} \rightarrow \{Y\})$

- **향상도(lift):** 연관 규칙 $\{X\} \rightarrow \{Y\}$ 에서 X를 구매했을 때 Y를 구매한 비율이 그러한 조건이 없던 때(그냥 Y를 구매한 비율)에 비해 얼마나 증가하는가를 보여주는 척도
 - 값이 1보다 크면 X를 샀을 때 Y를 살 확률이 높은 것을 의미
 - 값이 1 미만이면 X를 샀을 때 Y를 사지않을 확률이 높은 것을 의미
 - 향상도가 1이면 X를 산 것과 Y를 산 것은 관계가 없다는 의미

$support(\{Y\})$

$$lift(\{X\} \rightarrow \{Y\}) = \frac{confidence(\{X\} \rightarrow \{Y\})}{support(\{Y\})}$$

X를 구매 했을 경우,
Y도 구매한 비율

Y를 구매한 비율

- {맥주}→{땅콩}의 향상도

$$lift(\{맥주\} \rightarrow \{땅콩\}) = \frac{2/3}{2/5} = 1.67$$

X(맥주)를 구매했을 때
Y(땅콩)를 구매한 비율

Y(땅콩)를 구매한 비율

맥주를 살 때 땅콩을 구매하는 빈도가
땅콩을 사는 것보다 1.67배 높다

아래의 문제를 통해 연관규칙 알고리즘을 이해해보자.



거래번호	거래 아이템
1	우유, 버터, 시리얼
2	우유, 시리얼
3	우유, 빵
4	버터, 맥주, 오징어

문제 1. 지지도(support)

$$s(\text{우유}, \text{시리얼}) = ?, s(\{\text{우유}\} \rightarrow \{\text{시리얼}\}) = ?, s(\{\text{시리얼}\} \rightarrow \{\text{우유}\}) = ?$$

문제 2. 신뢰도(confidence)

$$c(\{\text{우유}\} \rightarrow \{\text{시리얼}\}) = s(\text{우유}, \text{시리얼}) / s(\text{우유}) = ?$$

문제 3. 향상도

$$\text{lift}(\{\text{우유}\} \rightarrow \{\text{시리얼}\}) = c(\{\text{우유}\} \rightarrow \{\text{시리얼}\}) / s(\text{시리얼}) = ?$$

아래의 문제를 통해 연관규칙 알고리즘을 이해해보자.



거래번호	거래 아이템
1	우유, 버터, 시리얼
2	우유, 시리얼
3	우유, 빵
4	버터, 맥주, 오징어

문제 1. 지지도

$$s(\text{우유}, \text{시리얼}) = n(X \cap Y) / N = 2/4 = 1/2$$

문제 2. 신뢰도

$$c(\text{우유} \rightarrow \text{시리얼}) = n(X \cap Y) / n(X) = n(\text{우유}, \text{시리얼}) / n(\text{우유}) = 1/2 / 3/4 = 2/3$$

문제 3. 향상도

$$\text{lift}(\text{우유} \rightarrow \text{시리얼}) = c(\text{우유} \rightarrow \text{시리얼}) / s(\text{시리얼}) = (2/3) / (2/4) = 1.333$$

우유를 살 때 시리얼을 구매하는 빈도가
시리얼을 사는 것보다 1.333배 높다

2. 구매 패턴 분석

3. 구매 패턴의 분석 과정

- 아프리오리 알고리즘: "arules" 패키지 이용
- 실습 결과의 시각화: "arulesViz" 패키지 이용
- 실습용 데이터셋: Kaggle에서 제공하는 제과점 거래 데이터(BreadBasket_DMS.csv)
(<https://www.kaggle.com/datasets/sulmansarwar/transactions-from-a-bakery>)
- BreadBasket 데이터셋은 어떤 제과점의 1년간 거래(판매) 내역을 정리한 것으로 169개의 상품에 대해 9835건의 거래내역을 포함하고 있음.

Date	Time	Transaction	Item
2016-10-30	9:58:11	1	Bread
2016-10-30	10:05:34	2	Scandinavian
2016-10-30	10:05:34	2	Scandinavian
2016-10-30	10:07:57	3	Hot chocolate
2016-10-30	10:07:57	3	Jam
2016-10-30	10:07:57	3	Cookies
2016-10-30	10:08:41	4	Muffin
2016-10-30	10:13:03	5	Coffee
2016-10-30	10:13:03	5	Pastry
2016-10-30	10:13:03	5	Bread
2016-10-30	10:16:55	6	Medialuna
2016-10-30	10:16:55	6	Pastry
2016-10-30	10:16:55	6	Muffin
2016-10-30	10:19:12	7	Medialuna
2016-10-30	10:19:12	7	Pastry
2016-10-30	10:19:12	7	Coffee
2016-10-30	10:19:12	7	Tea
2016-10-30	10:20:51	8	Pastry
2016-10-30	10:20:51	8	Bread
2016-10-30	10:21:59	9	Bread
2016-10-30	10:21:59	9	Muffin
2016-10-30	10:25:58	10	Scandinavian
2016-10-30	10:25:58	10	Medialuna

2. 구매 패턴 분석

2.1 데이터 준비와 관찰하기

코드 10-5 (계속)

```
library(arules)          # 아프리오리 알고리즘
library(arulesViz)       # 연관규칙 시각화 도구

# 데이터 불러오기와 관찰
setwd("D:/source")
ds <- read.csv("BreadBasket_DMS.csv") # 거래 데이터 읽기
str(ds)
head(ds)
unique(ds$item)

# 'NONE' item 삭제
ds.new <- subset(ds, item != 'NONE')
write.csv(ds.new, "BreadBasket_DMS_upd.csv", row.names = F )
```

2. 구매 패턴 분석

2.1 데이터의 준비와 관찰

코드 10-5

```
# 트랜잭션 포맷으로 데이터 읽기
trans <- read.transactions("BreadBasket_DMS_upd.csv",
                           format="single", header=T,
                           cols=c(3,4), sep="," , rm.duplicates=T)

trans                                # 트랜잭션 데이터 요약정보
dimnames(trans)[[2]]                # 상품 목록 확인
toLongFormat(trans)                 # 거래별 상품 목록
inspect(head(trans, 10))             # 앞부분 10개 트랜잭션 출력
```

2. 구매 패턴 분석

```
> library(arules)           # 아프리오리 알고리즘  
> library(arulesViz)        # 연관규칙 시각화 도구
```

```
> setwd("D:/source")  
> ds <- read.csv("BreadBasket_DMS.csv")  # 거래 데이터 읽기
```

```
> str(ds)  
'data.frame':21293 obs. of  4 variables:  
 $ Date      : chr  "2016-10-30" "2016-10-30" "2016-10-30" "2016-10-30" ...  
 $ Time      : chr  "09:58:11" "10:05:34" "10:05:34" "10:07:57" ...  
 $ Transaction: int   1 2 2 3 3 3 4 5 5 5 ...  
 $ Item      : chr  "Bread" "Scandinavian" "Scandinavian" "Hot chocolate" ...
```

2. 구매 패턴 분석

```
> head(ds)
```

	Date	Time	Transaction	Item
1	2016-10-30	09:58:11	1	Bread
2	2016-10-30	10:05:34	2	Scandinavian
3	2016-10-30	10:05:34	2	Scandinavian
4	2016-10-30	10:07:57	3	Hot chocolate
5	2016-10-30	10:07:57	3	Jam
6	2016-10-30	10:07:57	3	Cookies

2. 구매 패턴 분석

```
> unique(ds$Item)
[1] "Bread"                "Scandinavian"
[3] "Hot chocolate"        "Jam"
[5] "Cookies"              "Muffin"
[7] "Coffee"                "Pastry"
[9] "Medialuna"             "Tea"
[11] "NONE"                  "Tartine"
...(중간 생략)
[89] "Argentina Night"      "Half slice Monster "
[91] "Gift voucher"          "Cherry me Dried fruit"
[93] "Mortimer"              "Raw bars"
[95] "Tacos/Fajita"
```

```
> # 'NONE' item 삭제
> ds.new <- subset(ds, Item != 'NONE')
> write.csv(ds.new, "BreadBasket_DMS_upd.csv", row.names =F )
```

```
> # 트랜잭션 포맷으로 데이터 읽기
> trans <- read.transactions("BreadBasket_DMS_upd.csv", format="single",
+                             header=T, cols=c(3,4), sep=",", rm.duplicates=T)
```

2. 구매 패턴 분석

- `"BreadBasket_DMS_upd.csv"`

읽어올 트랜잭션(거래) 데이터가 저장된 파일을 지정한다.

- `format="single"`

읽어올 파일의 포맷을 지정한다.

- `"single"`

예제 파일과 같이 한 줄에 하나의 상품만 저장된 경우(즉, 하나의 거래 데이터가 여러 줄에 걸쳐 저장)

- `header=T`

읽어올 파일의 첫째 줄이 열의 변수명인지를 지정한다.

- `cols=c(3,4)`

파일에서 읽어올 열을 지정한다(3번째(트랜잭션 ID)와 4번째(상품) 열만 읽음).

- `sep=","`

파일에서 열과 열의 구분자가 무엇인지 지정한다(예제 파일은 CSV 포맷이므로 구분자가 ","이다).

- `rm.duplicates=T`

동일 트랜잭션 안에 중복된 상품이 있는 경우 중복을 제거할 것인지 지정한다.

2. 구매 패턴 분석

```
> trans                                # 트랜잭션 데이터 요약 정보
transactions in sparse format with
9465 transactions (rows) and
94 items (columns)
```

```
> dimnames(trans)[[2]]                # 상품 목록 확인
[1] "Adjustment"                        "Afternoon with the baker"
[3] "Alfajores"                         "Argentina Night"
[5] "Art Tray"                          "Bacon"
[7] "Baguette"                          "Bakewell"
...(중간 생략)
[87] "Tiffin"                            "Toast"
[89] "Truffles"                          "Tshirt"
[91] "Valentine's card"                  "Vegan Feast"
[93] "Vegan mincepie"                    "Victorian Sponge"
```

2. 구매 패턴 분석

```
> toLongFormat(trans)
```

```
# 거래별 상품 목록
```

	TID	item
1	1	Bread
2	2	Medialuna
3	2	Scandinavian
4	3	Bread
5	4	Chimichurri Oil
6	4	Scandinavian
7	5	Bread
8	5	Truffles
9	6	Brownie
...(이하 생략)		

2. 구매 패턴 분석

```
> inspect(head(trans, 10))      # 앞부분 10개 트랜잭션 출력
```

	items	transactionID
[1]	{Bread}	1
[2]	{Medialuna, Scandinavian}	10
[3]	{Bread}	100
[4]	{Chimichurri Oil, Scandinavian}	1000
[5]	{Bread, Truffles}	1001
[6]	{Brownie, Focaccia}	1002
[7]	{Bread, Coffee}	1003
[8]	{Art Tray, Coffee, Cookies, Tea}	1004
[9]	{Coffee}	1005
[10]	{Bread}	1006

2. 구매 패턴 분석

2.2 연관 규칙의 검색과 시각화

코드 10-6 (계속)

```
# 상품 판매 빈도
```

```
itemFrequencyPlot(trans, topN=10, type="absolute", xlab="상품명",  
  ylab="절대 판매빈도", main="판매량 많은 상품", col="green")
```

```
itemFrequencyPlot(trans, topN=10, type="relative", xlab="상품명",  
  ylab="상대 판매빈도", main="판매량 많은 상품", col="blue")
```

```
# 연관규칙 찾기
```

```
rules <- apriori(trans, parameter = list(supp = 0.001, conf = 0.7))  
rules
```

```
# 앞쪽 10개의 규칙 출력
```

```
options(digits=2)          # 평가 척도 값의 자릿수 지정  
inspect(rules[1:10])
```

2. 구매 패턴 분석

코드 10-6

```
# 신뢰도 상위 10개 규칙 출력
rules.sort <- sort(rules, by='confidence', decreasing = T)
inspect(rules.sort[1:10])

# 산점도 (지지도-향상도)
plot(rules.sort, measure=c("support", "lift"), shading="confidence")

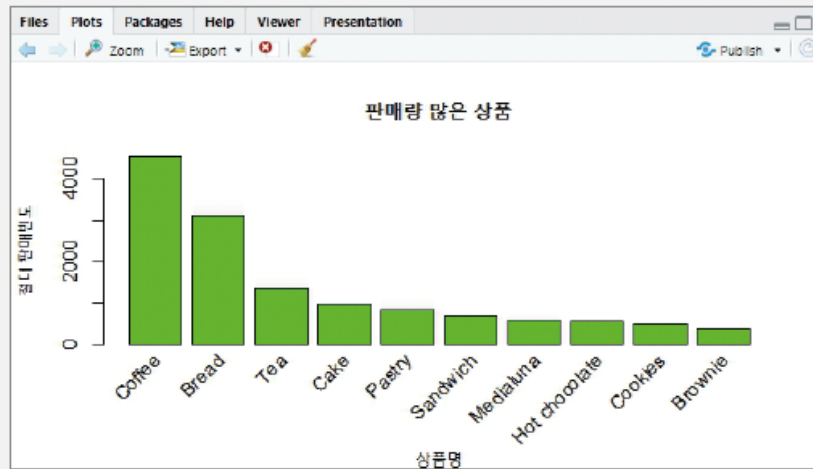
# Graph plot
plot(rules.sort, method="graph")

# Grouped Matrix Plot
plot(rules.sort, method="grouped")

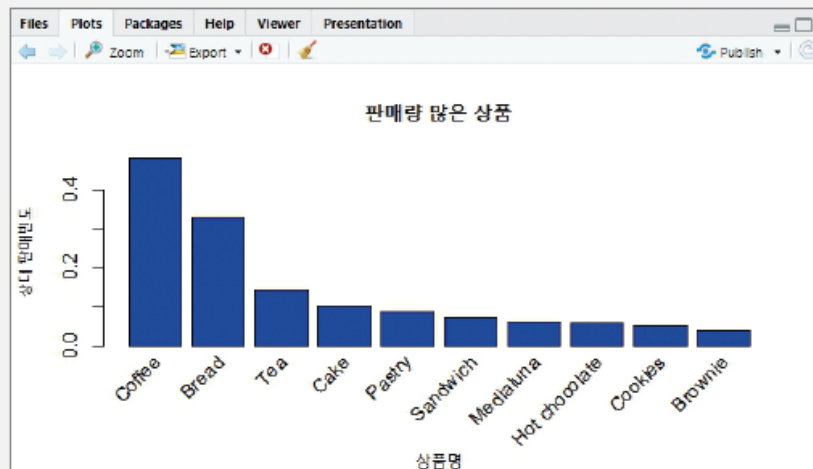
## 연관 규칙의 저장
write(rules.sort, file="BreadBasket_rules.csv", sep=',', quote=T,
row.names=F)
```

2. 구매 패턴 분석

```
> itemFrequencyPlot(trans, topN=10, type="absolute", xlab="상품명",  
+ ylab="절대 판매빈도", main="판매량 많은 상품", col="green")
```



```
> itemFrequencyPlot(trans, topN=10, type="relative", xlab="상품명",  
+ ylab="상대 판매빈도", main="판매량 많은 상품", col="blue")
```



2. 구매 패턴 분석

```
> # 연관규칙 찾기
```

```
> rules <- apriori(trans, parameter = list(supp = 0.001, conf = 0.7))
```

```
Apriori
```

Parameter specification:

confidence	minval	smax	arem	aval	originalSupport	maxtime	support	minlen
0.7	0.1	1	none	FALSE	TRUE	5	0.001	1
maxlen target		ext						
10 rules		TRUE						

Algorithmic control:

filter	tree	heap	memopt	load	sort	verbose
0.1	TRUE	TRUE	FALSE	TRUE	2	TRUE

Absolute minimum support count: 9

```
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[94 item(s), 9465 transaction(s)] done [0.00s].
sorting and recoding items ... [57 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [14 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
> rules
set of 14 rules
```

- ``trans``: 입력 데이터셋 (transaction 데이터)
- ``parameter``: 연관 규칙을 생성할 때 사용할 파라미터 목록
 - ``supp = 0.001``: 최소 지지도 (support)를 0.1%로 설정
 - ``conf = 0.7``: 최소 신뢰도 (confidence)를 70%로 설정

- ``support``: 지지도 기준 (0.001)
- ``minlen``: 최소 항목 수 (1)
- ``maxlen``: 최대 항목 수 (10)
- ``target``: 생성할 대상 (rules, 기본값)

절대 최소 지지도는 9: 9465건의 트랜잭션 중 $0.001 * 9465 \approx 9$ 건 이상 등장해야 한다는 의미

다음 R 코드 결과를 자세히 설명해 줘

```
> rules <- apriori(trans, parameter = list(supp = 0.001, conf = 0.7))
```

Apriori

Parameter specification:

```
confidence minval smax arem aval originalSupport maxtime
      0.7    0.1    1 none FALSE      TRUE     5
support minlen maxlen target ext
  0.001     1     10 rules TRUE
```

Algorithmic control:

```
filter tree heap memopt load sort verbose
  0.1 TRUE TRUE FALSE TRUE  2  TRUE
```

Absolute minimum support count: 9

```
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[94 item(s), 9465 transaction(s)] done [0.00s].
sorting and recoding items ... [57 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [14 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
```



```
> rules <- apriori(trans, parameter = list(supp = 0.001, conf = 0.7))
```

Apriori

Parameter specification:

confidence	minval	smax	arem	aval	originalSupport	maxtime	support	minlen	maxlen	target	ext
0.7	0.1	1	none	FALSE	TRUE	5	0.001	1	10	rules	TRUE

주요 파라미터:


- `confidence`: 최소 신뢰도 (Confidence) 설정.
 - 연관 규칙의 신뢰도가 0.7(70%) 이상이어야 규칙으로 포함.
- `support`: 최소 지지도 (Support) 설정.
 - 지지도 값이 0.001(0.1%) 이상인 규칙만 고려.
- `minlen`: 규칙의 최소 길이.
 - 최소 1개의 아이템을 포함한 규칙을 생성.
- `maxlen`: 규칙의 최대 길이.
 - 최대 10개의 아이템으로 이루어진 규칙을 생성.
- `target`: 생성할 대상.
 - `rules`: 연관 규칙을 생성.
- `maxtime`: 알고리즘 실행 시간 제한(초).
 - 5초로 설정(기본값).

기타:

- `originalSupport`: 트랜잭션의 원래 지지도를 기준으로 규칙을 생성할지 여부.
 - `TRUE`: 원래 지지도를 사용.
- `ext`: 확장된 규칙을 허용 여부.
 - `TRUE`: 확장된 규칙을 포함.

2. Algorithmic Control: 알고리즘 제어 설정

R

 코드 복사

Algorithmic control:


filter tree heap memopt load sort verbose

0.1 TRUE TRUE FALSE TRUE 2 TRUE

- `filter`: 항목 필터링 임계값(설정값: 0.1).
- `tree`: 트랜잭션 트리를 사용하여 효율적인 탐색 수행 여부(`TRUE`).
- `heap`: 힙 기반 메모리 최적화를 활성화 여부(`TRUE`).
- `memopt`: 메모리 최적화 옵션 (`FALSE` = 비활성화).
- `sort`: 아이템 정렬 수준 (설정값: 2).
- `verbose`: 실행 중 상세 정보를 출력 여부 (`TRUE`).

3. 절대 최소 지지도

R

 코드 복사

```
Absolute minimum support count: 9
```

- 의미:
 - 설정된 지지도(`supp = 0.001`)는 트랜잭션 개수(9465)의 0.1%에 해당.
 - $0.001 \times 9465 = 9$ 건 이상 나타난 규칙만 고려.

4. 실행 과정 설명

```
R

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[94 item(s), 9465 transaction(s)] done [0.00s].
```

코드 복사

- 데이터셋 준비:
 - 94개의 항목과 9465개의 트랜잭션을 처리.
 - `item appearances` 는 특정 항목에 대한 출현 조건이 없는 경우 0으로 표시.

```
R

sorting and recoding items ... [57 item(s)] done [0.00s].
```

코드 복사

- 항목 정렬 및 재코딩:
 - 빈도가 너무 낮은 항목을 제외하고, 57개의 유효 항목으로 축소.

```
R

creating transaction tree ... done [0.00s].
```

코드 복사

- 트랜잭션 트리 생성:
 - Apriori 알고리즘은 트랜잭션 트리를 생성하여 효율적으로 빈발 항목 집합을 탐색.

```
R

checking subsets of size 1 2 3 4 done [0.00s].
```

코드 복사

- 빈발 항목 집합 탐색:
 - 1개, 2개, 3개, 4개로 구성된 항목 집합의 빈발 항목 탐색 완료.

```
R

writing ... [14 rule(s)] done [0.00s].
```

코드 복사

- 연관 규칙 생성:
 - 설정한 조건(`supp = 0.001`, `conf = 0.7`)을 만족하는 14개의 규칙이 생성.

```
R

creating S4 object ... done [0.00s].
```

코드 복사

- S4 객체 생성:
 - 생성된 규칙을 S4 객체 형식으로 저장(`rules` 객체).

5. 주요 결과


데이터 요약

- 트랜잭션 수: 9465개.
- 항목 수: 94개 (유효 항목 57개).
- 최소 규칙 조건: 지지도 ≥ 0.001 (9건), 신뢰도 ≥ 0.7 .
- 생성된 규칙: 14개.

생성된 규칙 확인

규칙은 다음과 같이 확인할 수 있습니다:


R

 코드 복사

```
inspect(rules)
```

출력 예:

csharp


 코드 복사

	lhs	rhs	support	confidence	lift
[1]	{Bread}	=> {Coffee}	0.015	0.75	1.20
[2]	{Medialuna}	=> {Tea}	0.012	0.85	1.35
...					

규칙 필터링 및 정렬

- 예: `lift` 기준으로 상위 규칙 정렬.

R

 코드 복사

```
inspect(sort(rules, by = "lift")[1:5])
```

2. 구매 패턴 분석

- **trans**

읽어올 트랜잭션 데이터를 지정한다.

- **supp = 0.001**

지지도가 0.001 이상인 규칙만 검색한다.

- **conf = 0.7**

신뢰도가 0.7 이상인 규칙만 검색한다.

트랜잭션(거래) 수: 9465 건
 $9465 * 0.001 = 9.5$,
구매가 9건 이상 일어난 규칙만 검색

옵션 digits=n

```
> options(digits=2) # 평가척도 값의 자리수 지정
> inspect(rules[1:5])
```

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{Extra Salami or Feta}	=> {Coffee}	0.0033	0.82	0.0040	1.7	31
[2]	{Keeping It Local}	=> {Coffee}	0.0054	0.81	0.0067	1.7	51
[3]	{Toast}	=> {Coffee}	0.0237	0.70	0.0336	1.5	224
[4]	{Cake, Vegan mincepie}	=> {Coffee}	0.0011	0.83	0.0013	1.7	10
[5]	{Extra Salami or Feta, Salad}	=> {Coffee}	0.0015	0.88	0.0017	1.8	14

```
> options(digits=5) # 평가척도 값의 자리수 지정
> inspect(rules[1:10])
```

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{Extra Salami or Feta}	=> {Coffee}	0.0032752	0.81579	0.0040148	1.7053	31
[2]	{Keeping It Local}	=> {Coffee}	0.0053883	0.80952	0.0066561	1.6922	51
[3]	{Toast}	=> {Coffee}	0.0236661	0.70440	0.0335975	1.4724	224
[4]	{Cake, Vegan mincepie}	=> {Coffee}	0.0010565	0.83333	0.0012678	1.7419	10
[5]	{Extra Salami or Feta, Salad}	=> {Coffee}	0.0014791	0.87500	0.0016904	1.8290	14
[6]	{Hearty & Seasonal, Sandwich}	=> {Coffee}	0.0012678	0.85714	0.0014791	1.7917	12
[7]	{Salad, Sandwich}	=> {Coffee}	0.0015848	0.83333	0.0019017	1.7419	15
[8]	{Cake, Salad}	=> {Coffee}	0.0010565	0.76923	0.0013735	1.6079	10
[9]	{Juice, Spanish Brunch}	=> {Coffee}	0.0020074	0.73077	0.0027470	1.5275	19
[10]	{Pastry, Toast}	=> {Coffee}	0.0013735	0.86667	0.0015848	1.8116	13

2. 구매 패턴 분석

```
> # 앞쪽 10개의 규칙 출력
```

```
> options(digits=2)
```

```
# 평가척도 값의 자리수 지정
```

```
> inspect(rules[1:10])
```

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{Extra Salami or Feta}	⇒ {Coffee}	0.0033	0.82	0.0040	1.7	31
[2]	{Keeping It Local}	⇒ {Coffee}	0.0054	0.81	0.0067	1.7	51
[3]	{Toast}	⇒ {Coffee}	0.0237	0.70	0.0336	1.5	224
[4]	{Cake, Vegan mincepie}	⇒ {Coffee}	0.0011	0.83	0.0013	1.7	10
[5]	{Extra Salami or Feta, Salad}	⇒ {Coffee}	0.0015	0.88	0.0017	1.8	14
[6]	{Hearty & Seasonal, Sandwich}	⇒ {Coffee}	0.0013	0.86	0.0015	1.8	12
[7]	{Salad, Sandwich}	⇒ {Coffee}	0.0016	0.83	0.0019	1.7	15
[8]	{Cake, Salad}	⇒ {Coffee}	0.0011	0.77	0.0014	1.6	10
[9]	{Juice, Spanish Brunch}	⇒ {Coffee}	0.0020	0.73	0.0027	1.5	19
[10]	{Pastry, Toast}	⇒ {Coffee}	0.0014	0.87	0.0016	1.8	13

2. 구매 패턴 분석

```
> # 신뢰도 상위 10개 규칙 출력
```

```
> rules.sort <- sort(rules, by='confidence', decreasing = T)
```

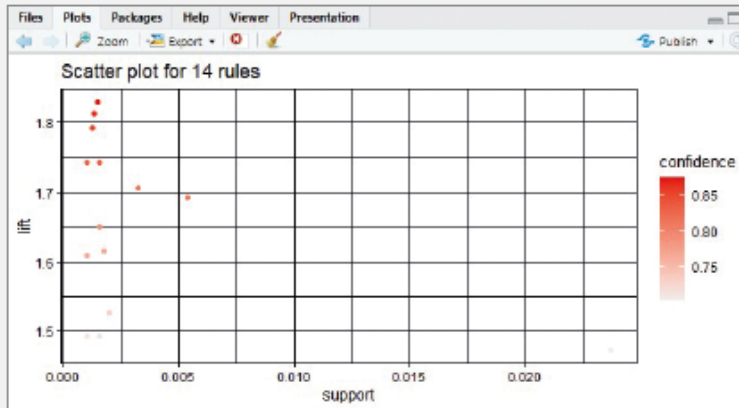
```
> inspect(rules.sort[1:10])
```

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{Extra Salami or Feta, Salad}	⇒ {Coffee}	0.0015	0.88	0.0017	1.8	14
[2]	{Pastry, Toast}	⇒ {Coffee}	0.0014	0.87	0.0016	1.8	13
[3]	{Hearty & Seasonal, Sandwich}	⇒ {Coffee}	0.0013	0.86	0.0015	1.8	12
[4]	{Cake, Vegan mincepie}	⇒ {Coffee}	0.0011	0.83	0.0013	1.7	10
[5]	{Salad, Sandwich}	⇒ {Coffee}	0.0016	0.83	0.0019	1.7	15
[6]	{Extra Salami or Feta}	⇒ {Coffee}	0.0033	0.82	0.0040	1.7	31
[7]	{Keeping It Local}	⇒ {Coffee}	0.0054	0.81	0.0067	1.7	51
[8]	{Cookies, Scone}	⇒ {Coffee}	0.0016	0.79	0.0020	1.7	15
[9]	{Juice, Pastry}	⇒ {Coffee}	0.0018	0.77	0.0023	1.6	17
[10]	{Cake, Salad}	⇒ {Coffee}	0.0011	0.77	0.0014	1.6	10

2. 구매 패턴 분석

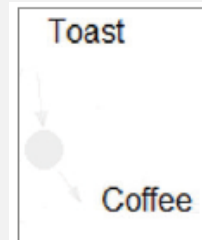
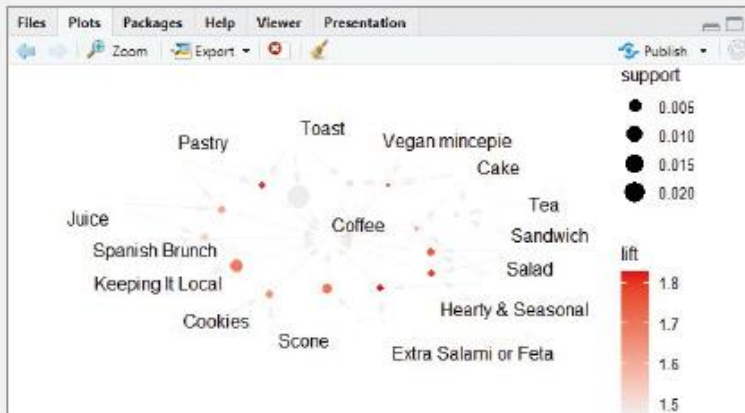
> # 산점도 (지지도-향상도)

> plot(rules.sort, measure=c("support", "lift"), shading="confidence")



> # Graph plot

> plot(rules.sort, method="graph")



(a) {Toast} → {Coffee}



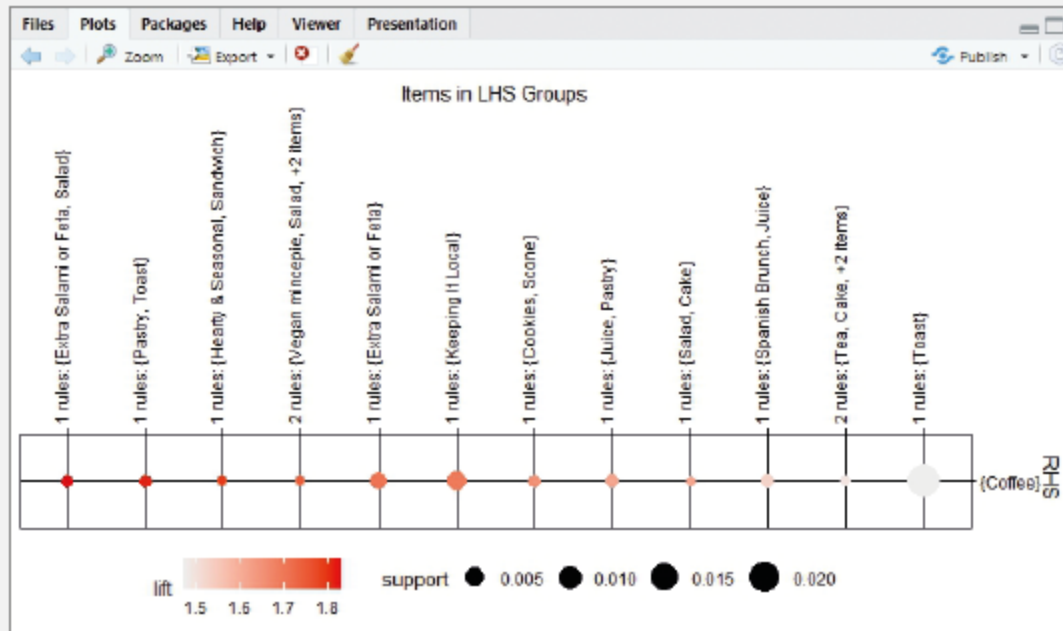
(b) {Pastry, Toast} → {Coffee}

그림 10-4 연관규칙의 그래프 표현

2. 구매 패턴 분석

> # Grouped Matrix Plot

> plot(rules.sort, method="grouped")



2. 구매 패턴 분석

> ## 연관규칙 저장

> write(rules.sort, file="BreadBasket_rules.csv", sep=',', quote=T, row.names=F)

	A	B	C	D	E	F
1	rules	support	confidence	coverage	lift	count
2	{Extra Salami or Feta,Salad} => {Coffee}	0.001479	0.875	0.00169	1.829036	14
3	{Pastry,Toast} => {Coffee}	0.001373	0.866667	0.001585	1.811617	13
4	{Hearty & Seasonal,Sandwich} => {Coffee}	0.001268	0.857143	0.001479	1.791709	12
5	{Cake,Vegan mincepie} => {Coffee}	0.001057	0.833333	0.001268	1.741939	10
6	{Salad,Sandwich} => {Coffee}	0.001585	0.833333	0.001902	1.741939	15
7	{Extra Salami or Feta} => {Coffee}	0.003275	0.815789	0.004015	1.705267	31
8	{Keeping It Local} => {Coffee}	0.005388	0.809524	0.006656	1.692169	51
9	{Cookies,Scone} => {Coffee}	0.001585	0.789474	0.002007	1.650258	15
10	{Juice,Pastry} => {Coffee}	0.001796	0.772727	0.002324	1.615253	17
11	{Cake,Salad} => {Coffee}	0.001057	0.769231	0.001373	1.607944	10
12	{Juice,Spanish Brunch} => {Coffee}	0.002007	0.730769	0.002747	1.527547	19
13	{Cake,Toast} => {Coffee}	0.001585	0.714286	0.002219	1.493091	15
14	{Cake,Sandwich,Tea} => {Coffee}	0.001057	0.714286	0.001479	1.493091	10
15	{Toast} => {Coffee}	0.023666	0.704403	0.033597	1.472431	224

그림 10-5 BreadBasket_rules.csv 파일

Section 03

인터넷 검색어 분석

2. 인터넷 검색어 분석

- 인터넷 검색어를 중심으로 사용자들의 관심사를 분석할 수 있도록 지원해주는 많은 사이트들이 있음
- 네이버 데이터랩과 구글 트렌드가 대표적
- 네이버 데이터랩에서는 주로 국내의 관심사를 알아볼 수 있고, 구글 트렌드에서는 전 세계적인 관심사를 확인



그림 10-6 네이버 데이터랩 초기화면(<http://datalab.naver.com/>)

2. 인터넷 검색어 분석

1. 분야별 인기 검색어 확인

분야별 인기 검색어		인기분야
디지털/가전		
2023.02.16.(목)	2023.02.17.(금)	2023.02.18.(토)
1 유무선공유기	1 유무선공유기	1 유무선공유기
2 공유기	2 공유기	2 공유기
3 스피커	3 스피커	3 스피커
4 로봇청소기	4 로봇청소기	4 키보드
5 가습기	5 가습기	5 로봇청소기

그림 10-7 디지털/가전 분야의 인기 검색어

3. 인터넷 검색어 분석

2. 관심 키워드로 트렌드 분석

검색어트렌드 네이버통합검색에서 특정 검색어가 얼마나 많이 검색되었는지 확인해보세요. 검색어를 기간별/연월별/성공금한 주제어를 설정하고, 하위 주제어에 해당하는 검색어를 콤마(,)로 구분입력해 주세요. 입력한 단어의 추이를 하나 선택되는지 조회할 수 있습니다. 예) 주제어 캠핑 : 캠핑, Camping, 캠핑용품, 겨울캠핑, 캠핑장, 글램핑, 오토캠핑, 캠핑카

주제어1 주제어 1에 해당하는 모든 검색어를 콤마(,)로 구분하여 최대 30개까지 입력

주제어2 주제어 2에 해당하는 모든 검색어를 콤마(,)로 구분하여 최대 30개까지 입력

기간 전체 1개월 3개월 **년** 3월입력 월간

2022 02 21 - 2023 02 21

* 2010년 1월 이후 조회할 수 있습니다.

범위 ☒ 전체 ☒ 모바일 ☒ PC

성별 ☒ 전체 ☒ 여성 ☒ 남성

연령선택 ☐ 전체 ☐ -12 ☐ 13-18 ☐ 19-24 ☒ 25-29 ☒ 30-34 ☐ 35-39 ☐ 40-44 ☐ 45-49 ☐ 50-54 ☐ 55-59

그림 10-8 키워드를 통한 검색어 트렌드 조회

3. 인터넷 검색어 분석

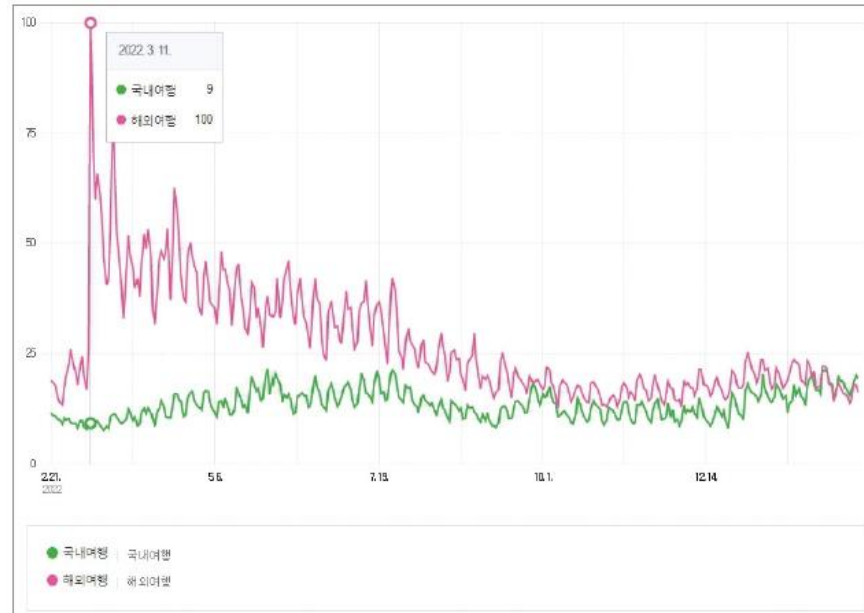


그림 10-9 '국내여행'과 '해외여행'에 대한 검색어 트렌드

- 전반적으로 국내여행보다는 해외여행에 대한 검색 비중이 높음.
- 3월 중순부터 검색 횟수가 급증하여 5월 말까지 지속적으로 이어짐.
- 이는 여행하기 좋은 계절이 다가오는 4, 5월에 해외여행에 대한 수요가 높아진 것으로 볼 수 있음.
- 또한 코로나19 팬데믹이 점차 안정화되면서 그동안 억눌렸던 해외여행에 대한 수요가 폭발한 것으로 분석.

3. 인터넷 검색어 분석

3. 지역별 관심업종과 카드지출 추이 분석

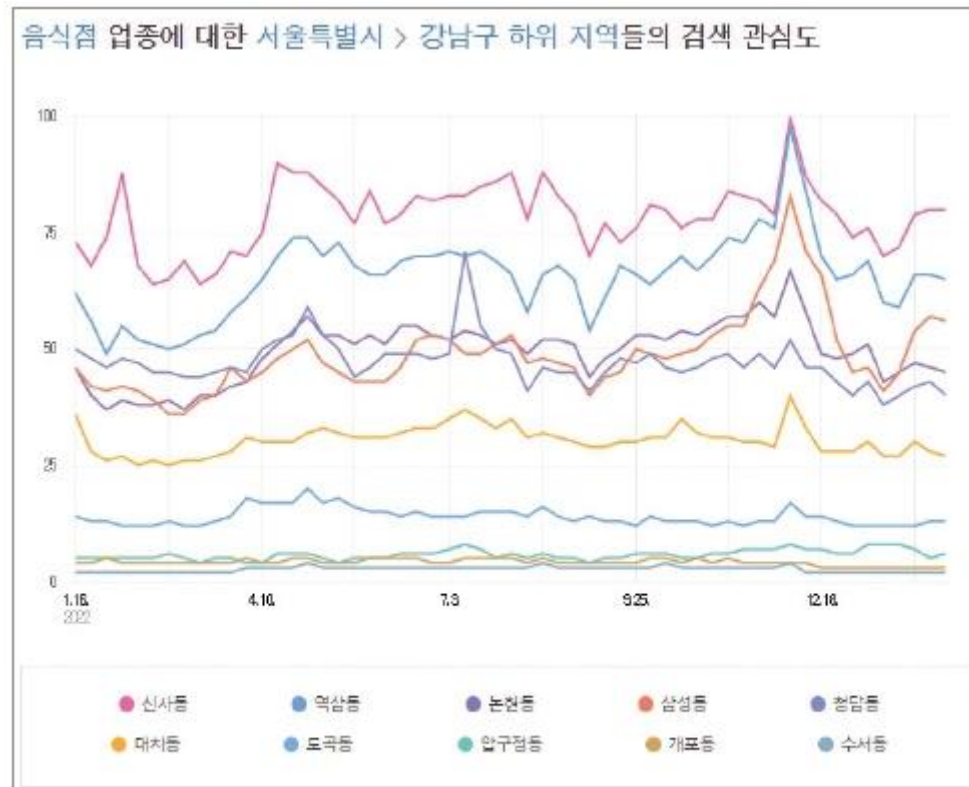


그림 10-10 서울시 강남구의 음식점 업종에 대한 검색어 트렌드

3. 인터넷 검색어 분석

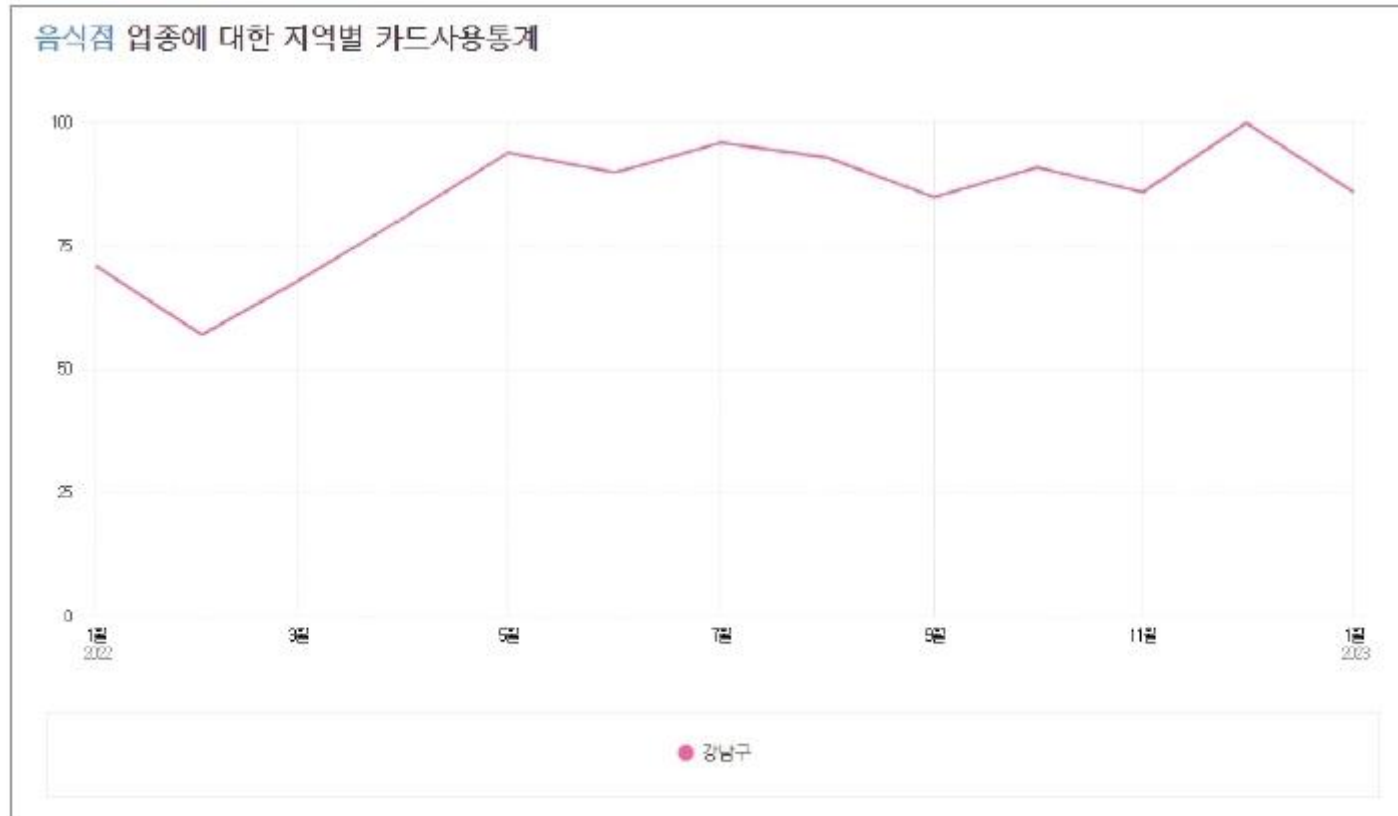


그림 10-11 서울시 강남구의 음식점에 대한 20대의 카드지출 추이

Section 04

공공 빅데이터

4. 공공 빅데이터

1. 공공데이터 포털

- 가장 풍부한 공공데이터를 제공하는 사이트
- 엑셀 형식의 파일을 직접 다운로드하는 방식과 컴퓨터 프로그램 안에서 API를 이용하여 가져오는 방식으로 데이터 제공



그림 10-12 공공데이터포털(<https://www.data.go.kr>)

4. 공공 빅데이터

- 서울시에서는 공공데이터를 활용하여 공중화장실의 위치정보를 모바일 앱을 통하여 제공
- 제공하는 정보는 공중화장실 정보를 포함하여 화장실 사진, 화장실 유형(외부인 개방 및 남녀구분 여부), 화장실 편의시설(장애인 전용칸, 기저귀 교환대 및 세면대 유무), 화장실 청결도 및 안전도 등



그림 10-13 공공데이터를 이용한 화장실 위치 안내 앱

4. 공공 빅데이터

2. 기상청 날씨누리

- 최근 미세먼지와 날씨에 대한 관심이 높아지고 있음
- 기상청에서는 기상 관련 데이터를 공개하며, 다운로드도 가능
- 기상예보, 태풍, 황사, 위성, 레이더 등 25종 자료를 쉽게 이용
- 현재 기상 자료를 실시간으로 얻을 수 있으므로 기상 관련 앱을 개발 시 이용 가능



그림 10-14 기상청 날씨누리(<https://www.weather.go.kr>)

4. 공공 빅데이터

3. 국가통계포털

- 국내외 주요 통계를 한 곳에 모아 이용자가 원하는 통계를 한 번에 찾을 수 있도록 통계청이 제공하는 원스톱(One-Stop) 통계 서비스 웹사이트
- 현재 300여 개 기관이 작성하는 경제·사회·환경에 관한 1,000여 종의 국가승인통계를 수록
- 국제금융과 경제에 관한 국제통화기금(IMF), 월드뱅크(Worldbank), 경제협력개발기구(OECD) 등의 최신 통계도 제공
- 편리한 검색 기능과 일반인들도 쉽게 이해할 수 있는 다양한 콘텐츠 및 통계 설명 자료 서비스를 제공



그림 10-15 국가통계포털(<https://kosis.kr>)

4. 공공 빅데이터

4. 통합 데이터 지도

- 공공과 민간에서 제공하는 데이터를 쉽게 검색·활용할 수 있도록 지원하는 것을 목표로 국가적 차원에서 구축
- 이 사이트를 통해 16대 빅데이터 플랫폼과 AI Hub, 데이터스토어, 공공데이터 포털 등을 아우르는 방대한 데이터에 접근 가능.



그림 10-16 통합데이터지도(<https://www.bigdata-map.kr/>)

Thank you!