채널 D
2016125077 최재혁
2015125043 어성준
2015125080 표정진

데이터 사이언스

- 11주차 팀 과제 (공공 데이터를 활용한 연관분석)

목 차

- 데이터 셋 선정, 속성 파악
- 데이터 가공 및 전처리
- Transaction데이터와 eclat함수
- Rule 생성과 지지도, 신뢰도, 향상도
- 여러가지 Rule 생성
- 결과(참신성, 유용성)와 고찰

데이터 셋 선정, 속성

- 데이터 셋 선정
  - 공공 데이터를 알아보던 중, Kaggle에 있는 Seoul Airpollution data set을 선택
  - 링크 : https://www.kaggle.com/bappekim/air-pollution-in-seoul
- 데이터 속성 조사
  - 4개의 데이터 중, 2가지를 선정
  - Measurement_info : 대기 성분을 시간, 장소에 따라 측정한 데이터 (3885066개)
  - Measurement_item_info : 대기 품질 기준 분류

	Measurement.date	Station.code	Item.code	Average.value	Instrument.status
1	2017-01-01 00:00	101	1	0.004	0
2	2017-01-01 00:00	101	3	0.059	0
3	2017-01-01 00:00	101	5	1.200	0
4	2017-01-01 00:00	101	6	0.002	0
5	2017-01-01 00:00	101	8	73.000	0
6	2017-01-01 00:00	101	9	57.000	0
7	2017-01-01 00:00	102	1	0.006	0
8	2017-01-01 00:00	102	3	0.068	0
9	2017-01-01 00:00	102	5	1.300	0
10	2017-01-01 00:00	102	6	0.002	0
11	2017-01-01 00:00	102	8	77.000	0
12	2017-01-01 00:00	102	9	63.000	0
13	2017-01-01 00:00	103	1	0.005	0

<사진 1> Measurement info

	Item.code	Item.name	Unit.of.measurement	Good.Blue.	Normal.Green.	Bad.Yellow.	Very.bad.Red.
1	1	SO2	ppm	0.02	0.05	0.15	1.0
2	3	NO2	ppm	0.03	0.06	0.20	2.0
3	5	CO	ppm	2.00	9.00	15.00	50.0
4	6	O3	ppm	0.03	0.09	0.15	0.5
5	8	PM10	Mircrogram/m3	30.00	80.00	150.00	600.0
6	9	PM2.5	Mircrogram/m3	15.00	35.00	75.00	500.0

<사진 2> Measurement item info

데이터 가공, 전처리

- 데이터 결측치 제거
  - Na.omit()함수를 이용하여 결측치를 제거

```
df <- na.omit(df[df$Instrument.status == 0,]) # 결측치 제거
```

- 데이터 전처리 - 1
  - 데이터 셋을 날짜와 시간으로 구분 하여 정리

```
# 데이터 가공
df <- dcast(data = df, Measurement.date + Station.code ~ Item.code,
            value.var = "Average.value")
# 날짜 나누기
df <- separate(data = df, col = Measurement.date,
               sep = ' ', into = c("date", "time"))
colnames(df) <- c("date", "time", "station", "SO2", "NO2", "CO", "O3", "PM10", "PM2.5")
df
```

<사진 1> 전처리 – 1 코드

	date	time	station	SO2	NO2	CO	O3	PM10	PM2.5
1	2017-01-01	00:00	101	0.004	0.059	1.2	0.002	73	57
2	2017-01-01	00:00	102	0.006	0.068	1.3	0.002	77	63
3	2017-01-01	00:00	103	0.005	0.039	1.4	0.002	70	68
4	2017-01-01	00:00	104	0.005	0.045	0.6	0.003	73	46
5	2017-01-01	00:00	105	0.005	0.044	1.0	0.004	81	44
6	2017-01-01	00:00	106	0.005	0.066	1.5	0.003	71	62
7	2017-01-01	00:00	107	0.005	0.049	0.9	0.002	64	40
8	2017-01-01	00:00	108	0.004	0.045	0.8	0.003	68	63
9	2017-01-01	00:00	109	0.006	0.052	1.1	0.002	76	50
10	2017-01-01	00:00	110	0.005	0.040	0.8	0.002	91	50
11	2017-01-01	00:00	111	0.005	0.047	0.9	0.002	62	38
12	2017-01-01	00:00	112	0.004	0.046	1.2	0.001	63	51
13	2017-01-01	00:00	113	0.006	0.051	0.9	0.002	81	40
14	2017-01-01	00:00	114	0.008	0.055	1.4	0.002	75	62
15	2017-01-01	00:00	115	0.005	0.055	1.3	0.002	75	48
16	2017-01-01	00:00	116	0.007	0.070	1.3	0.002	107	65
17	2017-01-01	00:00	117	0.007	0.045	1.3	0.003	72	63
18	2017-01-01	00:00	118	0.004	0.060	1.2	0.001	67	45
19	2017-01-01	00:00	119	0.005	0.035	1.5	0.004	70	46
20	2017-01-01	00:00	120	0.006	0.062	1.2	0.002	63	49

<사진 2> 날짜와 시간별로 정리된 데이터

데이터 가공, 전처리

- 데이터 전처리 - 2
  - 일별로 대기 성분에 대한 평균을 구함

```
# 일 평균내기
df_daymean <- aggregate(cbind(SO2, NO2, CO, O3, PM10, PM2.5) ~ date+station, df, FUN = mean)
```

<사진 1> 전처리 – 2 코드

- 데이터 전처리 - 3
  - 평균을 구한 데이터를 가지고 Measurement_item_info 표와 비교하여 명목형 변수로 변환

```
# 일 평균내기
df_daymean <- aggregate(cbind(SO2, NO2, CO, O3, PM10, PM2.5) ~ date+station, df, FUN = mean)

df_daymean$SO2_eval <- ifelse(df_daymean$SO2 <= eval_table$Good.Blue.[1], 'Good',
                              ifelse(df_daymean$SO2 <= eval_table$Normal.Green.[1],'Normal',
                                     ifelse(df_daymean$SO2 <= eval_table$Bad.Yellow.[1],'Bad',
                                            'Very Bad')))

df_daymean$NO2_eval <- ifelse(df_daymean$NO2 <= eval_table$Good.Blue.[2], 'Good',
                              ifelse(df_daymean$NO2 <= eval_table$Normal.Green.[2],'Normal',
                                     ifelse(df_daymean$NO2 <= eval_table$Bad.Yellow.[2],'Bad',
                                            'Very Bad')))

df_daymean$CO_eval <- ifelse(df_daymean$CO <= eval_table$Good.Blue.[3], 'Good',
                             ifelse(df_daymean$CO <= eval_table$Normal.Green.[3],'Normal',
                                    ifelse(df_daymean$CO <= eval_table$Bad.Yellow.[3],'Bad',
                                           'Very Bad')))

df_daymean$O3_eval <- ifelse(df_daymean$O3 <= eval_table$Good.Blue.[4], 'Good',
                             ifelse(df_daymean$O3 <= eval_table$Normal.Green.[4],'Normal',
                                    ifelse(df_daymean$O3 <= eval_table$Bad.Yellow.[4],'Bad',
                                           'Very Bad')))

df_daymean$PM10_eval <- ifelse(df_daymean$PM10 <= eval_table$Good.Blue.[5], 'Good',
                               ifelse(df_daymean$PM10 <= eval_table$Normal.Green.[5],'Normal',
```

<사진 2> 전처리 – 3 코드

	date	station	so2	no2	co	o3	pm10	pm2.5
1	2017-01-01	101	Good	Normal	Good	Good	Bad	Bad
2	2017-01-02	101	Good	Normal	Good	Good	Bad	Very Bad
3	2017-01-03	101	Good	Normal	Good	Good	Normal	Bad
4	2017-01-04	101	Good	Normal	Good	Good	Normal	Normal
5	2017-01-05	101	Good	Normal	Good	Good	Normal	Normal
6	2017-01-06	101	Good	Normal	Good	Good	Good	Good
7	2017-01-07	101	Good	Normal	Good	Good	Normal	Bad
8	2017-01-08	101	Good	Normal	Good	Good	Normal	Normal
9	2017-01-09	101	Good	Good	Good	Good	Normal	Bad
10	2017-01-10	101	Good	Good	Good	Good	Good	Normal
11	2017-01-11	101	Good	Normal	Good	Good	Good	Good
12	2017-01-12	101	Good	Good	Good	Good	Good	Normal
13	2017-01-13	101	Good	Good	Good	Good	Good	Good
14	2017-01-14	101	Good	Good	Good	Good	Normal	Good
15	2017-01-15	101	Good	Good	Good	Good	Good	Good
16	2017-01-16	101	Good	Normal	Good	Good	Normal	Normal

<사진 4> 명목형 변수로 변환한 표

Transaction 데이터와 eclat()함수

- Transaction 데이터
  - 가공해 놓은 data frame() (airpollution data set)을 transaction 데이터로 변환
  - Transaction class는 spase format(희소형태)이며, arules package에 적합하도록 변환
- Eclat()함수
  - Equivalence Class Transformation
  - Apriori 를 보완하기 위해서 등장
  - eclat()함수를 이용해서 전체 transaction data에 대해서 support(지지도)와 itemset갯수를 확인함
  - Parameter : supp – 최소 지지도 설정, minlen – 최소 itemset길이, maxlen – 최대 itemset길이

```
> airpollution.trans <- as(airpollution,"transactions")
> airpollution.trans
transactions in sparse format with
 26518 transactions (rows) and
 1126 items (columns)
> class(airpollution.trans)
[1] "transactions"
attr(,"package")
[1] "arules"
```

<사진 1> transaction data

```
> frequentItems <- eclat (airpollution.trans,
+                         parameter = list(supp = 0.021,minlen=2,maxlen = 10))
Eclat

parameter specification:
 tidLists support minlen maxlen           target  ext
    FALSE   0.021      2     10 frequent itemsets TRUE

algorithmic control:
 sparse sort verbose
      7   -2    TRUE

Absolute minimum support count: 556

create itemset ...
set transactions ...[1126 item(s), 26518 transaction(s)] done [0.06s].
sorting and recoding items ... [37 item(s)] done [0.00s].
creating bit matrix ... [37 row(s), 26518 column(s)] done [0.00s].
writing  ... [742 set(s)] done [0.00s].
Creating S4 object  ... done [0.00s].
```

<사진 2> itemset 확인을 위한 eclat함수

Transaction 데이터와 eclat()함수

- Itemsets 확인

```
inspect(sort(frequentItems)[1:50])
inspect(head(frequentItems, 3))
summary(frequentItems)

itemFrequencyPlot(airpollution.trans,topN=12,
                  type="absolute",
                  main="Item Frequency")
```

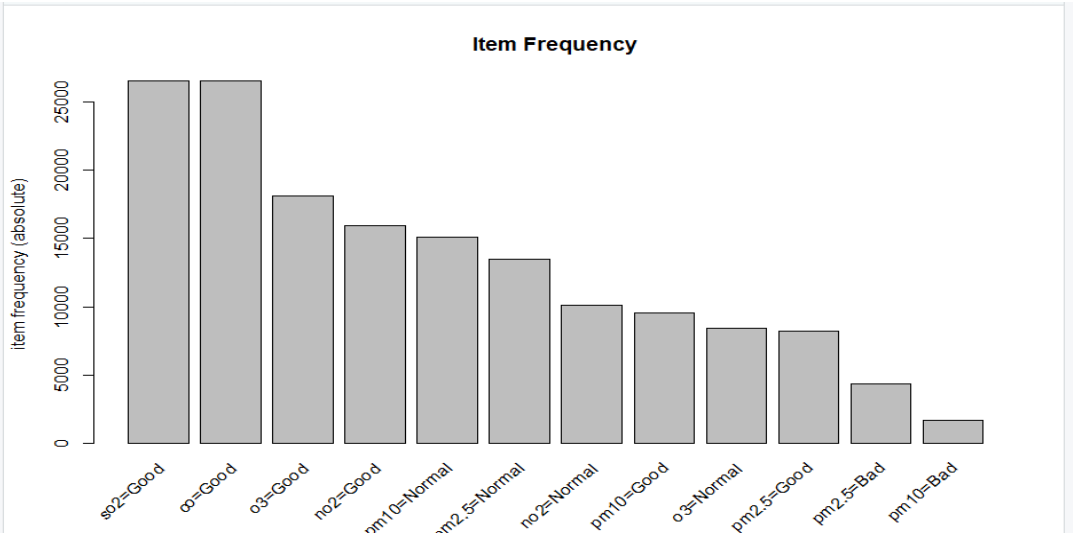
<사진 1> itemsets 확인 코드

```
> summary(frequentItems)
set of 742 itemsets

most frequent items:
   so2=Good      co=Good      o3=Good     no2=Good pm10=Normal      (Other)
        389          389          219          203          203          984

element (itemset/transaction) length distribution:sizes
  2   3   4   5   6
182 295 197  58  10

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   3.000   3.000   3.217   4.000   6.000

summary of quality measures:
     support        transIdenticalToItemsets     count
 Min.   :0.02100   Min.   :  557            Min.   :  557
 1st Qu.:0.02376   1st Qu.:  630            1st Qu.:  630
 Median :0.03292   Median :  873            Median :  873
 Mean   :0.08825   Mean   : 2340            Mean   : 2340
 3rd Qu.:0.09462   3rd Qu.: 2509            3rd Qu.: 2509
 Max.   :1.00000   Max.   :26518            Max.   :26518

includes transaction ID lists: FALSE

mining info:
             data ntransactions support
 airpollution.trans         26518   0.021
```

<사진 3> itemsets 요약

```
> inspect(sort(frequentItems)[1:50])
     items                                                  support   transIdenticalToItemsets count
[1]  {so2=Good,co=Good}                                     1.0000000 26518                     26518
[2]  {so2=Good,co=Good,o3=Good}                             0.6832717 18119                     18119
[3]  {so2=Good,o3=Good}                                     0.6832717 18119                     18119
[4]  {co=Good,o3=Good}                                      0.6832717 18119                     18119
[5]  {so2=Good,no2=Good,co=Good}                            0.6015914 15953                     15953
[6]  {so2=Good,no2=Good}                                    0.6015914 15953                     15953
[7]  {no2=Good,co=Good}                                     0.6015914 15953                     15953
[8]  {so2=Good,co=Good,pm10=Normal}                         0.5705181 15129                     15129
[9]  {so2=Good,pm10=Normal}                                 0.5705181 15129                     15129
[10] {co=Good,pm10=Normal}                                  0.5705181 15129                     15129
[11] {so2=Good,co=Good,pm2.5=Normal}                        0.5094653 13510                     13510
[12] {so2=Good,pm2.5=Normal}                                0.5094653 13510                     13510
[13] {co=Good,pm2.5=Normal}                                 0.5094653 13510                     13510
[14] {so2=Good,co=Good,pm10=Normal,pm2.5=Normal}            0.4041406 10717                     10717
[15] {so2=Good,pm10=Normal,pm2.5=Normal}                    0.4041406 10717                     10717
[16] {co=Good,pm10=Normal,pm2.5=Normal}                     0.4041406 10717                     10717
[17] {pm10=Normal,pm2.5=Normal}                             0.4041406 10717                     10717
[18] {so2=Good,no2=Normal,co=Good}                          0.3810242 10104                     10104
[19] {so2=Good,no2=Normal}                                  0.3810242 10104                     10104
[20] {no2=Normal,co=Good}                                   0.3810242 10104                     10104
[21] {so2=Good,co=Good,o3=Good,pm10=Normal}                 0.3757071  9963                      9963
[22] {so2=Good,o3=Good,pm10=Normal}                         0.3757071  9963                      9963
[23] {co=Good,o3=Good,pm10=Normal}                          0.3757071  9963                      9963
```

<사진 2> items를 지지도로 내림차순



<사진 4> 상위 12개의 높은 빈도 item

Rule 생성과 상관관계 분석

- Rule 생성조건
  - 최소 지지도 0.2, 신뢰도 0.5, 최소길이 2로 설정
- Rule 분석
  - 오른쪽 사진 2의 지지도 정렬 결과
  - 모든 결과가 향상도(Lift)가 1이기 때문에
  - 상관관계가 전혀 없다

```
rules <- apriori (airpollution.trans,
                  parameter = list(supp = 0.2,
                                   conf = 0.5,
                                   minlen = 2))

rules_supp <- sort (rules, by="support", decreasing=TRUE)
inspect(head(rules_supp,30))

rules_conf <- sort (rules, by="confidence", decreasing=TRUE)
inspect(head(rules_conf,30))

rules_lift <- sort (rules, by="lift", decreasing=TRUE)
inspect(head(rules_lift,30))
```

<사진 1> Rule 생성

```
> inspect(head(rules_supp,30))
     lhs                          rhs             support   confidence coverage  lift count
[1]  {so2=Good}                => {co=Good}      1.0000000 1.0000000  1.0000000 1    26518
[2]  {co=Good}                 => {so2=Good}     1.0000000 1.0000000  1.0000000 1    26518
[3]  {o3=Good}                 => {so2=Good}     0.6832717 1.0000000  0.6832717 1    18119
[4]  {so2=Good}                => {o3=Good}      0.6832717 0.6832717  1.0000000 1    18119
[5]  {o3=Good}                 => {co=Good}      0.6832717 1.0000000  0.6832717 1    18119
[6]  {co=Good}                 => {o3=Good}      0.6832717 0.6832717  1.0000000 1    18119
[7]  {so2=Good,o3=Good}        => {co=Good}      0.6832717 1.0000000  0.6832717 1    18119
[8]  {co=Good,o3=Good}         => {so2=Good}     0.6832717 1.0000000  0.6832717 1    18119
[9]  {so2=Good,co=Good}        => {o3=Good}      0.6832717 0.6832717  1.0000000 1    18119
[10] {no2=Good}                => {so2=Good}     0.6015914 1.0000000  0.6015914 1    15953
[11] {so2=Good}                => {no2=Good}     0.6015914 0.6015914  1.0000000 1    15953
[12] {no2=Good}                => {co=Good}      0.6015914 1.0000000  0.6015914 1    15953
[13] {co=Good}                 => {no2=Good}     0.6015914 0.6015914  1.0000000 1    15953
[14] {so2=Good,no2=Good}       => {co=Good}      0.6015914 1.0000000  0.6015914 1    15953
[15] {no2=Good,co=Good}        => {so2=Good}     0.6015914 1.0000000  0.6015914 1    15953
[16] {so2=Good,co=Good}        => {no2=Good}     0.6015914 0.6015914  1.0000000 1    15953
[17] {pm10=Normal}             => {so2=Good}     0.5705181 1.0000000  0.5705181 1    15129
[18] {so2=Good}                => {pm10=Normal}  0.5705181 0.5705181  1.0000000 1    15129
[19] {pm10=Normal}             => {co=Good}      0.5705181 1.0000000  0.5705181 1    15129
[20] {co=Good}                 => {pm10=Normal}  0.5705181 0.5705181  1.0000000 1    15129
[21] {so2=Good,pm10=Normal}    => {co=Good}      0.5705181 1.0000000  0.5705181 1    15129
[22] {co=Good,pm10=Normal}     => {so2=Good}     0.5705181 1.0000000  0.5705181 1    15129
[23] {so2=Good,co=Good}        => {pm10=Normal}  0.5705181 0.5705181  1.0000000 1    15129
[24] {pm2.5=Normal}            => {so2=Good}     0.5094653 1.0000000  0.5094653 1    13510
[25] {so2=Good}                => {pm2.5=Normal} 0.5094653 0.5094653  1.0000000 1    13510
[26] {pm2.5=Normal}            => {co=Good}      0.5094653 1.0000000  0.5094653 1    13510
[27] {co=Good}                 => {pm2.5=Normal} 0.5094653 0.5094653  1.0000000 1    13510
[28] {so2=Good,pm2.5=Normal}   => {co=Good}      0.5094653 1.0000000  0.5094653 1    13510
[29] {co=Good,pm2.5=Normal}    => {so2=Good}     0.5094653 1.0000000  0.5094653 1    13510
```

<사진 2> 지지도 정렬

Rule 생성과 상관관계 분석 (Cont)

- 향상도 정렬을 통한 Rule 분석
  - PM10량이 좋으면 PM2.5량도 좋다.
  - 반대로 PM2.5량이 좋아도 PM10량도 좋다

  - 위 두 패턴은 양방향적이며
  - 적정치의 지지도, 신뢰도, 향상도를 가진다.

  - 역정렬에서는 가장 낮은 향상도 값이 0.83으로
  - 음의 상관관계이지만
  - 비교적 독립적인 결과라고 볼 수 있다.

```
> rules_lift <- sort (rules, by="lift", decreasing=FALSE)
> inspect(head(rules_lift,30))
     lhs                                                rhs          support   confidence coverage  lift     count
[1]  {pm10=Normal,pm2.5=Normal}                      => {no2=Good}    0.2022023 0.5003266  0.4041406 0.8316718 5362
[2]  {so2=Good,pm10=Normal,pm2.5=Normal}             => {no2=Good}    0.2022023 0.5003266  0.4041406 0.8316718 5362
[3]  {co=Good,pm10=Normal,pm2.5=Normal}              => {no2=Good}    0.2022023 0.5003266  0.4041406 0.8316718 5362
[4]  {so2=Good,co=Good,pm10=Normal,pm2.5=Normal}     => {no2=Good}    0.2022023 0.5003266  0.4041406 0.8316718 5362
[5]  {no2=Good}                                      => {o3=Good}     0.3504789 0.5825863  0.6015914 0.8526422 9294
[6]  {o3=Good}                                       => {no2=Good}    0.3504789 0.5129422  0.6832717 0.8526422 9294
```

<사진 2> 향상도 역 정렬

```
> inspect(head(rules_lift,30))
     lhs                                                rhs          support   confidence coverage  lift     count
[1]  {no2=Good,pm10=Good}                            => {pm2.5=Good}  0.2483973 0.7811907  0.3179727 2.518922 6587
[2]  {so2=Good,no2=Good,pm10=Good}                   => {pm2.5=Good}  0.2483973 0.7811907  0.3179727 2.518922 6587
[3]  {no2=Good,co=Good,pm10=Good}                    => {pm2.5=Good}  0.2483973 0.7811907  0.3179727 2.518922 6587
[4]  {so2=Good,no2=Good,co=Good,pm10=Good}           => {pm2.5=Good}  0.2483973 0.7811907  0.3179727 2.518922 6587
[5]  {no2=Good,pm2.5=Good}                           => {pm10=Good}   0.2483973 0.8807327  0.2820348 2.441488 6587
[6]  {so2=Good,no2=Good,pm2.5=Good}                  => {pm10=Good}   0.2483973 0.8807327  0.2820348 2.441488 6587
[7]  {no2=Good,co=Good,pm2.5=Good}                   => {pm10=Good}   0.2483973 0.8807327  0.2820348 2.441488 6587
[8]  {so2=Good,no2=Good,co=Good,pm2.5=Good}          => {pm10=Good}   0.2483973 0.8807327  0.2820348 2.441488 6587
[9]  {pm2.5=Good}                                    => {pm10=Good}   0.2694019 0.8686770  0.3101290 2.408068 7144
[10] {so2=Good,pm2.5=Good}                           => {pm10=Good}   0.2694019 0.8686770  0.3101290 2.408068 7144
[11] {co=Good,pm2.5=Good}                            => {pm10=Good}   0.2694019 0.8686770  0.3101290 2.408068 7144
[12] {so2=Good,co=Good,pm2.5=Good}                   => {pm10=Good}   0.2694019 0.8686770  0.3101290 2.408068 7144
[13] {pm10=Good}                                     => {pm2.5=Good}  0.2694019 0.7468116  0.3607361 2.408068 7144
[14] {so2=Good,pm10=Good}                            => {pm2.5=Good}  0.2694019 0.7468116  0.3607361 2.408068 7144
[15] {co=Good,pm10=Good}                             => {pm2.5=Good}  0.2694019 0.7468116  0.3607361 2.408068 7144
[16] {so2=Good,co=Good,pm10=Good}                    => {pm2.5=Good}  0.2694019 0.7468116  0.3607361 2.408068 7144
[17] {no2=Good,o3=Good}                              => {pm10=Good}   0.2196621 0.6267484  0.3504789 1.737415 5825
[18] {so2=Good,no2=Good,o3=Good}                     => {pm10=Good}   0.2196621 0.6267484  0.3504789 1.737415 5825
[19] {no2=Good,co=Good,o3=Good}                      => {pm10=Good}   0.2196621 0.6267484  0.3504789 1.737415 5825
[20] {so2=Good,no2=Good,co=Good,o3=Good}             => {pm10=Good}   0.2196621 0.6267484  0.3504789 1.737415 5825
[21] {o3=Good,pm10=Normal}                           => {no2=Normal}  0.2425899 0.6456890  0.3757071 1.694614 6433
[22] {so2=Good,o3=Good,pm10=Normal}                  => {no2=Normal}  0.2425899 0.6456890  0.3757071 1.694614 6433
[23] {co=Good,o3=Good,pm10=Normal}                   => {no2=Normal}  0.2425899 0.6456890  0.3757071 1.694614 6433
[24] {so2=Good,co=Good,o3=Good,pm10=Normal}          => {no2=Normal}  0.2425899 0.6456890  0.3757071 1.694614 6433
[25] {no2=Normal,pm2.5=Normal}                       => {pm10=Normal} 0.2003168 0.8780165  0.2281469 1.538981 5312
[26] {so2=Good,no2=Normal,pm2.5=Normal}              => {pm10=Normal} 0.2003168 0.8780165  0.2281469 1.538981 5312
[27] {no2=Normal,co=Good,pm2.5=Normal}               => {pm10=Normal} 0.2003168 0.8780165  0.2281469 1.538981 5312
[28] {so2=Good,no2=Normal,co=Good,pm2.5=Normal}      => {pm10=Normal} 0.2003168 0.8780165  0.2281469 1.538981 5312
[29] {pm10=Good,pm2.5=Good}                          => {no2=Good}    0.2483973 0.9220325  0.2694019 1.532656 6587
```

<사진 1> 향상도 정렬

Rule 생성2과 상관관계 분석

- Rule 생성조건
  - 최소 지지도 0.001, 신뢰도 0.05, 최소길이 2, lhs 는 station, rhs는 default 로 설정
- Rule 분석
  - 오른쪽 사진 2을 보면 rule이 12개가 생긴 것을 확인
  - 지역마다 대기성분과의 연관성을 파악할 수 있음

```
rules2 <- apriori (data=airpollution.trans,
                   parameter=list (supp=0.001,
                                   conf = 0.05,
                                   minlen=2),
                   appearance = list(default="rhs",
                                     lhs="station=121"),
                   control = list (verbose=F))

rules_conf <- sort (rules2, by="confidence", decreasing=TRUE)
inspect(head(rules_conf,10))

rules_lift <- sort (rules2, by="lift", decreasing=TRUE)
inspect(head(rules_lift,10))

rules_support <- sort (rules2, by="support", decreasing=TRUE)
inspect(head(rules_support,30))
```

<사진 1> Rule 생성

```
> rules2
set of 12 rules
> inspect(rules2)
     lhs              rhs              support     confidence coverage  lift      count
[1]  {station=121} => {pm10=Bad}      0.002715137 0.07157058 0.0379365 1.1190499   72
[2]  {station=121} => {pm2.5=Bad}     0.008333962 0.21968191 0.0379365 1.3413596  221
[3]  {station=121} => {pm2.5=Good}    0.009540689 0.25149105 0.0379365 0.8109241  253
[4]  {station=121} => {o3=Normal}     0.013236292 0.34890656 0.0379365 1.1017271  351
[5]  {station=121} => {pm10=Good}     0.011652462 0.30715706 0.0379365 0.8514730  309
[6]  {station=121} => {no2=Normal}    0.015913719 0.41948310 0.0379365 1.1009356  422
[7]  {station=121} => {pm2.5=Normal}  0.019307640 0.50894632 0.0379365 0.9989814  512
[8]  {station=121} => {pm10=Normal}   0.023380345 0.61630219 0.0379365 1.0802499  620
[9]  {station=121} => {no2=Good}      0.020137265 0.53081511 0.0379365 0.8823516  534
[10] {station=121} => {o3=Good}       0.024700204 0.65109344 0.0379365 0.9529056  655
[11] {station=121} => {co=Good}       0.037936496 1.00000000 0.0379365 1.0000000 1006
[12] {station=121} => {so2=Good}      0.037936496 1.00000000 0.0379365 1.0000000 1006
```

<사진 2> Rule 파악

Rule 생성2과 상관관계 분석 (Cont)

- 지표 정렬을 통한 Rule 분석

  - 지지도를 통해서 확인한 결과 상위 2가지 품목, co=Good, so2=Good이라는 결과를 확인

    (비교적 지지도가 낮게 나온 이유는 전체 품목의 개수(분모)가 굉장히 많(크)기 때문)

  - 그래서 신뢰도를 확인해 본 결과, 상위 2품목은 1로써 규칙의 신뢰성이 높다는 것을 확인

  - 결론은 지역별(현재는 121 = 관악구)로 대기 품질에 대한 척도를 판별하는 것이 가능하다고 생각 됨

```
> rules_conf <- sort (rules2, by="confidence", decreasing=TRUE)
> inspect(head(rules_conf,10))
     lhs              rhs              support     confidence coverage  lift      count
[1]  {station=121} => {co=Good}       0.037936496 1.0000000  0.0379365 1.0000000 1006
[2]  {station=121} => {so2=Good}      0.037936496 1.0000000  0.0379365 1.0000000 1006
[3]  {station=121} => {o3=Good}       0.024700204 0.6510934  0.0379365 0.9529056  655
[4]  {station=121} => {pm10=Normal}   0.023380345 0.6163022  0.0379365 1.0802499  620
[5]  {station=121} => {no2=Good}      0.020137265 0.5308151  0.0379365 0.8823516  534
[6]  {station=121} => {pm2.5=Normal} 0.019307640 0.5089463  0.0379365 0.9989814  512
[7]  {station=121} => {no2=Normal}    0.015913719 0.4194831  0.0379365 1.1009356  422
[8]  {station=121} => {o3=Normal}     0.013236292 0.3489066  0.0379365 1.1017271  351
[9]  {station=121} => {pm10=Good}     0.011652462 0.3071571  0.0379365 0.8514730  309
[10] {station=121} => {pm2.5=Good}    0.009540689 0.2514911  0.0379365 0.8109241  253
```

<사진 1> 신뢰도 정렬

```
> rules_support <- sort (rules2, by="support", decreasing=TRUE)
> inspect(head(rules_support,30))
     lhs              rhs              support     confidence coverage  lift      count
[1]  {station=121} => {co=Good}       0.037936496 1.00000000 0.0379365 1.0000000 1006
[2]  {station=121} => {so2=Good}      0.037936496 1.00000000 0.0379365 1.0000000 1006
[3]  {station=121} => {o3=Good}       0.024700204 0.65109344 0.0379365 0.9529056  655
[4]  {station=121} => {pm10=Normal}   0.023380345 0.61630219 0.0379365 1.0802499  620
[5]  {station=121} => {no2=Good}      0.020137265 0.53081511 0.0379365 0.8823516  534
[6]  {station=121} => {pm2.5=Normal} 0.019307640 0.50894632 0.0379365 0.9989814  512
[7]  {station=121} => {no2=Normal}    0.015913719 0.41948310 0.0379365 1.1009356  422
[8]  {station=121} => {o3=Normal}     0.013236292 0.34890656 0.0379365 1.1017271  351
[9]  {station=121} => {pm10=Good}     0.011652462 0.30715706 0.0379365 0.8514730  309
[10] {station=121} => {pm2.5=Good}    0.009540689 0.25149105 0.0379365 0.8109241  253
[11] {station=121} => {pm2.5=Bad}     0.008333962 0.21968191 0.0379365 1.3413596  221
[12] {station=121} => {pm10=Bad}      0.002715137 0.07157058 0.0379365 1.1190499   72
```

<사진 2> 지지도 정렬

```
> rules_lift <- sort (rules2, by="lift", decreasing=TRUE)
> inspect(head(rules_lift,10))
     lhs              rhs              support     confidence coverage  lift      count
[1]  {station=121} => {pm2.5=Bad}     0.008333962 0.21968191 0.0379365 1.3413596  221
[2]  {station=121} => {pm10=Bad}      0.002715137 0.07157058 0.0379365 1.1190499   72
[3]  {station=121} => {o3=Normal}     0.013236292 0.34890656 0.0379365 1.1017271  351
[4]  {station=121} => {no2=Normal}    0.015913719 0.41948310 0.0379365 1.1009356  422
[5]  {station=121} => {pm10=Normal}   0.023380345 0.61630219 0.0379365 1.0802499  620
[6]  {station=121} => {co=Good}       0.037936496 1.00000000 0.0379365 1.0000000 1006
[7]  {station=121} => {so2=Good}      0.037936496 1.00000000 0.0379365 1.0000000 1006
[8]  {station=121} => {pm2.5=Normal} 0.019307640 0.50894632 0.0379365 0.9989814  512
[9]  {station=121} => {o3=Good}       0.024700204 0.65109344 0.0379365 0.9529056  655
[10] {station=121} => {no2=Good}      0.020137265 0.53081511 0.0379365 0.8823516  534
```

<사진 3> 향상도 정렬

결론 및 고찰

- 결론

< 유용성 >

- 지역별로 대기 오염 성분에 대한 척도를 파악할 수 있게 됨
- 이 분석 결과를 바탕으로 대기오염과 각 지역의 환경요소, 폐 질환 발병률 등의 요소에 대해 연관성을 분석할 수 있다

< 참신성 >

- 처음에는 지역별로 오염 성분 여부를 알아내기 위했으나, 대기 오염 성분간의 연관성이 존재한다는 것을 확인함

결론 및 고찰

- 고찰

  - Eclat vs Apriori

    Eclat은 빈발집합을 생성(자주 팔리는 아이템들의 집합)

    Apriori는 규칙을 생성(자주 팔리는 아이템들의 집합은 그 부분집합도 자주 팔린다)

    Apriori는 큰 데이터 셋에 적합, Eclat은 중소 규모의 데이터 셋에 적합

    Apriori 보다 Eclat의 속도가 더 빠름

    Apriori는 지지도와 신뢰도를 필요로 하지만 Eclat은 지지도만 필요로 함

결론 및 고찰

- 고찰

  - Eclat vs Apriori

```
frequentItems <- sort(frequentItems, by="support", decreasing=TRUE)
inspect(frequentItems)
```

```
     items             support     transIdenticalToItemsets count
[1] {PM10,PM2.5}      0.070628916 45733                    45733
[2] {NO2,PM2.5}       0.029988680 19418                    19418
[3] {NO2,PM10}        0.017037548 11032                    11032
[4] {NO2,PM10,PM2.5} 0.016688520 10806                    10806
[5] {O3,PM2.5}        0.003700323  2396                     2396
[6] {O3,PM10}         0.001423914   922                      922
[7] {O3,PM10,PM2.5}   0.001303453   844                      844
```

```
rules <- sort(rules, by="support", decreasing=TRUE)
inspect(rules)
```

```
     lhs             rhs      support     confidence coverage    lift     count
[1] {PM10}       => {PM2.5} 0.070628916 0.8051159  0.087725151 4.267565 45733
[2] {NO2}        => {PM2.5} 0.029988680 0.6948649  0.043157568 3.683173 19418
[3] {NO2,PM10}   => {PM2.5} 0.016688520 0.9795141  0.017037548 5.191973 10806
[4] {NO2,PM2.5}  => {PM10}  0.016688520 0.5564940  0.029988680 6.343608 10806
[5] {O3}         => {PM2.5} 0.003700323 0.5802858  0.006376726 3.075839  2396
[6] {O3,PM10}    => {PM2.5} 0.001303453 0.9154013  0.001423914 4.852139   844
```

실제 데이터 셋에 적용한 결과