

# 서울시 문화공간 개수 및 카페 개수 상관 분석

국민대학교 소프트웨어학부 20142770 최 락 준

(HP: 010-5850-9766, E-MAIL: choirak0805@naver.com)

## 목차

1. 서론
2. 프로젝트 과정
3. 프로젝트 결과
4. 코드

## 1. 서론

프로젝트 주제 선정에 대해서 설명하면, 초기에 선택했던 ‘OHSEMED DATA 주제 모델링’은 특별한 HADOOP 프로그래밍이나 MONGODB 등을 사용하지 않음. SPARK의 MLIB에 의존하여 몇 가지 알고리즘을 사용하는 것에 지나지 않음. 따라서 서울시 공공데이터 중 서울시 구 별 문화공간 현황과 서울시 구 별 커피숍 정보 DATA를 이용해 둘의 상관 관계를 도출하고자 한다.

사용 컴퓨팅 환경은 GOOGLE CLOUD PLATFORM이며 HADOOP MAP-REDUCE와 SPARK의 MLIB을 사용함. 또한 사용한 DATA는 서울시 열린 데이터 광장에 업로드 되어있는 DATA를 사용함.

번호	사업장명	소재지전체주소	도로명전체주소	인허가일자	영업상태명	영업일자	휴업시작일자
1	골매커피	서울특별시 성동구 성동동 1가 6-	서울특별시 성동구 서울동 4동 2-	20130605	운영중		
2	학다방	서울특별시 성동구 용답동 196-	서울특별시 성동구 용답리로 29-	19760205	운영중		
3	남방커피	서울특별시 성동구 용답동 2가 2-	서울특별시 성동구 용답리로 11-	20030315	운영중		
4	올	서울특별시 성동구 용답동 234-	서울특별시 성동구 거동저서동 1-	19810723	운영중		
5	아방다방	서울특별시 성동구 성동동 1가 1-	서울특별시 성동구 연무장동 6-	19871022	운영중		
6	무희	서울특별시 성동구 금호동 3가 4-	서울특별시 성동구 독서당동 29-	19760205	운영중		
7	삼거리	서울특별시 성동구 도선동 363-	서울특별시 성동구 용답리로 36-	19760422	운영중		
8	대저리	서울특별시 성동구 용답동 234-	서울특별시 성동구 거동저서동 1-	19911116	운영중		
9	수정	서울특별시 성동구 용답동 234-		19850716	운영중		

Figure 1. 커피숍 정보\_DATA\_SHEET

문화공간코드	문화공간명	문화공간명	대표이미지	주소	전화번호
100319	3	백화점/기타	http://culture.seoul.go.kr/	서울 종로구 가회동 11-32	02-744-1545
100517	1	공공기관	http://culture.seoul.go.kr/	서울 종로구 예장동 8-145	02-3455-8318
100464	8	도서관	http://culture.seoul.go.kr/	서울 강서구 관서동 520-6	02-708-3700
100082	8	도서관	http://culture.seoul.go.kr/	서울 강서구 관서동 520-6	02-3219-7000
100873	1	공공기관	http://culture.seoul.go.kr/	서울 종로구 명동길 2-6	02-739-8288
100457	1	공공기관	http://culture.seoul.go.kr/	서울 종로구 명동길 270	02-708-5001
100813	8	도서관	http://culture.seoul.go.kr/	서울 강서구	02-863-9544-6
100877	6	문화예술공간	http://culture.seoul.go.kr/	서울 종로구 계동 53-1	02-747-0303
100311	7	문화공간	http://culture.seoul.go.kr/	서울 용산구 후암동 339-1	02-2021-2800

Figure 2. 문화 공간 현황 DATA\_SHEET

FIGURE 1,2 의 DATA는 CSV 파일 형식을 가짐.

FIGURE 1의 DATA는 번호/사업장/소재지 주소/도로명 주소/인허가 일자/영업 상태의 순으로 저장되어 있음. FIGURE 2는 문화공간 코드/장르 분류 코드/장르 분류 명/문화 공간 명/대표 이미지/주소/전화/팩스/홈페이지/관람시간/관람료/휴관일/개관일자/객석 수/X좌표/Y좌표/기타 사항/무료 구분으로 이루어짐.

## 2. 프로젝트 과정

먼저, 서울시 각 구별 카페 개수 및 문화 공간 개수 파악을 위해 GOOGLE CLOUD에 CSV 형식의 두 개 파일을 업로드 함. 그리고 각 DATA SET에서 주소 FIELD를 기준으로 COUNT를 한 후 저장함. HADOOP의 MAP-REDUCE를 통하여 각 주소

를 기준으로 COUNT 를 해야 한다.

구 별 CAFÉ 개수 파악 과정을 먼저 설명함. 각 파일을 TEXTINPUTFORAMT 형식으로 받아 각 MAPPER 에서 LINE 단위로 주소 FILED 를 기준으로 KEY 는 구 VALUE 는 ONE 으로 MAP 의 OUTPUT 을 준다. 예를 들어 서울시 강남구에 카페가 하나 있다면 <강남구, ONE> 의 형식으로 MAP 의 OUTPUT 을 준다. 그것을 REDUCER 에서는 주소 기준으로 받아 최종적으로 구 별 카페 개수를 COUNT 한다.

이와 같은 방식으로 구 별 문화 공간 개수 파악 과정을 진행하는데 이 때 발생한 문제점으로는 기존에 입력 받은 CSV 파일의 DATA 가 깔끔하지 않아서 어떤 것은 주소에 앞의 FILED 내용이 있거나 주소 부분이 공백으로 되어 있는 경우가 있어 문제점이 있었지만 이는 MAP 과정에서 읽을 때 몇 가지 OPTION 을 줌으로 해결했다. 따라서 여기서도 DATA 를 각 <구, ONE> 형식으로 문화공간을 MAPPER 에서 OUTPUT 으로 주면 REDUCER 에서 같은 KEY 를 가진 것끼리 묶는 과정을 통해 최종적으로 구 별 문화 공간 COUNT 를 도출했다.

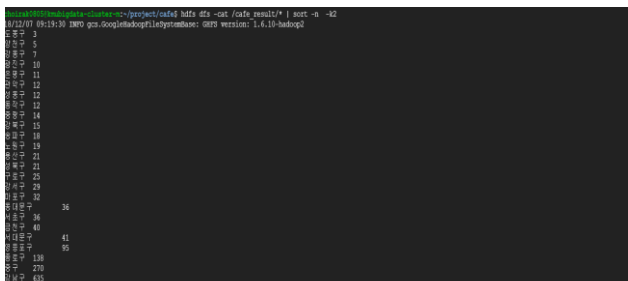


Figure 2. 구 별 카페 개수를 정렬해 표현

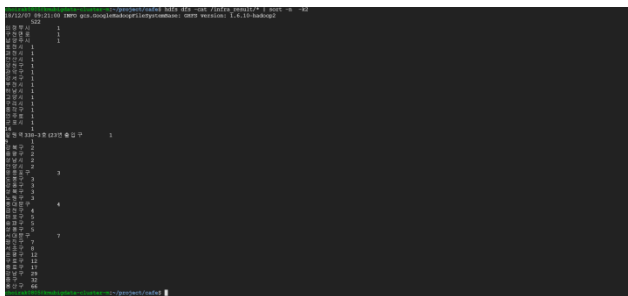


Figure 3. 구 별 문화 공간 개수를 정렬해 표현

위에서 HADOOP MAP-REDUCE 로 얻은 결과를

상위 6 개 기준으로 잘라 나열함.

#### 구 별 문화 공간 개수

1. 용산구 66 개
  2. 중구 32 개
  3. 강남구 29 개
  4. 종로구 17 개
  5. 구로구 12 개
  6. 은평구 12 개
- 결과를 나타낸다.

#### 구 별 카페 개수

1. 강남구 635 개
  2. 중구 270 개
  3. 종로구 138 개
  4. 영등포구 95 개
  5. 서대문구 41 개
  6. 금천구 40 개
- 결과를 나타낸다.

위 결과를 바탕으로 둘의 상관 계수를 파악하기 전에 순위를 기반으로 분석해보면 가설 ‘구 별 문화 공간이 많을수록 카페 개수가 많을 것이다’ 검증을 하면 가설은 성립하지 않는다. 또한 이에 따라 둘의 상관 관계가 낮을 것이라고 예측할 수 있다. 왜냐하면 문화 공간이 많은 순위 용산-> 중구-> 강남-> 종로 -> 구로 ->은평이지만 실제 구 별 카페 개수는 강남->중구->종로 -> 영등포 -> 서대문 -> 금천으로 각각 강남, 중구, 종로가 상위권을 유지하기는 하지만 카페 순위가 문화 공간의 순위를 따르는 않기 때문이다. 하지만 이는 개인적인 분석에 따른 내용이기때 보다 정확하고 통계적인 방법을 위해서

SPARK 의 MLIB 를 사용해 둘의 상관 계수를 파악했다. 상관 관계 분석(CORRELATION ANALYSIS)는 연속 형 또는 순위 자료로 이루어진 두 변수 간 상호 관계만을 알아보려고 할 때 사용하는 분석 방법이다. 특히 두 변수 간의 상관 관계의 크기는 상관 계수를 통해 정량화 되어 계산되는데 상관 관계 계수로는 PEARSON CORRELATION COEFFICIENT, SPEARMAN CORRELATION COEFFICIENT 가 있다. 여기서 둘의 차이는 SPEARMAN CORRELATION COEFFICIENT 의 경우 DATA 가 정규 분포를 벗어난다거나 두 변수가 순위 척도 자료일 때 사용하는 것이다. 또한 이는 단순히 변수가 증가할 때 다른 변수의 증감에 대한 정성적인 관계만을 나타낸다. 예를 들어, 지역별 범죄 율에 강한 상관관계를 보이는 변인을 찾는 데 사용하는 통계 방식이 CORRELATION 인데 문화공간 개수와 카페 개수의 상관 관계를 분석하고자 하기에 이를 선택함. 두 개의 DATA 를 이용해 상관 계수를 도출하면 해당 상관 계수의 값 (-1 ~ 1)을 통해 해석할 수 있다.

각 값에 따른 해석을 나열하겠다.

- -1 에 가까운 값: 누가 봐도 매우 강력한 음(-)의 상관.

- -0.5 정도의: 강력한 음(-)의 상관. 연구자는 변인 x 가 증가하면 변인 y 가 감소한다고 자신 있게 말할 수 있다.
- -0.2 정도의 값: 음(-)의 상관이지만 한테 너무 약해서 모호하다. 상관관계가 없다고는 할 수 없지만 좀 더 의심해 봐야 한다.
- 0 정도의: 대부분의 경우, 상관 관계가 존재하지 않을 것이라고 예측 가능하다.
- 0.2 정도의 값: 너무 약해서 의심스러운 양(+)의 상관.
- 0.5 정도의 값: 강력한 양(+)의 상관. 변인 x 가 증가하면 변인 y 가 증가한다고 판단 가능함.
- 1 에 가까운 값: 이상할 정도로 강력한 양(+)의 상관.

하지만 현재 분석 할 DATA 는 순위 DATA 가 아닌 연속 형 DATA 이고 가설의 목적에 따라 선형적인 상관 관계 크기를 따지고자 하기 때문에 'PEARSON CORRELATION' 방식의 SPARK MLIB 를 사용했다.

MLIB 에 대한 DATA 입력으로는 HADOOP MAP-REDUCE 에 따른 결과 값을 기준으로 구 별 카페 개수, 문화 공간 개수를 줌. Vectors.dense (카페 개수, 문화 공간 개수) 형태로 줬다. 즉, (강남 카페 개수, 강남 문화 공간 개수), (용산 카페 개수, 용산 문화 공간 개수) 형태로 총 MR 작업을 바탕으로 서울 25 개 구에 대한 DATA 입력을 줬다.

```
corr: org.apache.spark.mllib.linalg.Matrix =
1.0      0.4050482853110014
0.4050482853110014  1.0
```

Figure 4. 서울시 문화공간과 카페 간 상관 계수

### 3. 프로젝트 결과

위와 같은 결과를 바탕으로 상관 계수가 0.4050482853110014 임을 도출했다. 이를 통해 위의 상관 계수 범위 별 해석기준 기반으로 해석해보자면 강력한 양의 상관 관계를 가진다. 즉, 변수 X 가 증가하면 Y 가 증가한다고 판단할 수 있다.

## 4. 프로젝트 코드

### 4-1. 서울 특별 시 구 별 CAFE 개수 도출 MAP-REDUCE

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import java.util.Arrays;
import java.util.ArrayList;

public class cafe_per_ku {

    public static void main(String[] args) throws Exception {

        Configuration conf = new Configuration();

        Job job = Job.getInstance(conf, "cafe_per_ku");

        job.setJarByClass(cafe_per_ku.class);

        // job.setNumReduceTasks(1); // To SUM "FILM-NOIR" GENRE TOGETHER AT ONE-REDUCE

        job.setOutputKeyClass(Text.class); // KEY-VALUE (MOVIE_TITLE - COUNT)
        job.setOutputValueClass(IntWritable.class);

        job.setMapperClass(cafe_per_ku_MAP.class);
        job.setReducerClass(cafe_per_ku_REDUCE.class);

        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        job.waitForCompletion(true);

    }

    public static class cafe_per_ku_MAP extends Mapper<LongWritable, Text, Text, IntWritable> {

        private final static IntWritable ONE = new IntWritable(1);
        private Text word = new Text();

        public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {

            String line = value.toString(); // READ LINE-BY-LINE
            String [] fields = line.split(" "); // DECODE -CSV [" ", " ", " ....."]
            String address = fields[2];
            if (address != null) {
                String [] add = address.split("\\.");
                if (add.length > 2) {
                    String for_key = add[1];
                    word.set(for_key);
                    context.write(word, ONE);
                }
            }
        }
    }

    public static class cafe_per_ku_REDUCE extends Reducer<Text, IntWritable, Text, IntWritable> {

        /*
         * REDUCE RETURN UNIQUE TITLE-ONE PER BLOCK. SO WE SHOULD GET UNIQUE IN REDUCE
         */

        public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {

            int sum = 0;

            for (IntWritable val : values) {

                sum += val.get();

            }

            context.write(key, new IntWritable(sum));

        }
    }
}
```

#### 4-2. 서울 특별 시 구 별 문화 공간 개수 도출 MAP-REDUCE

```
import java.io.IOException;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import java.util.Arrays;
import java.util.ArrayList;

public class infra_per_ku {

    public static void main(String[] args) throws Exception{

        Configuration conf=new Configuration();

        Job job = Job.getInstance(conf,"infra_per_ku");

        job.setJarByClass(infra_per_ku.class);

//        job.setNumReduceTasks(1);//To SUM "FILM-NOIR" GENRE TOGETHER AT ONE-REDUCE

        job.setOutputKeyClass(Text.class); //KEY-VALUE (MOVIE_TITLE - COUNT)
        job.setOutputValueClass(IntWritable.class);

        job.setMapperClass(infra_per_ku_MAP.class);
        job.setReducerClass(infra_per_ku_REDUCE.class);

        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        FileInputFormat.addInputPath(job,new Path(args[0]));
        FileOutputFormat.setOutputPath(job,new Path(args[1]));

        job.waitForCompletion(true);

    }

    public static class infra_per_ku_MAP extends Mapper<LongWritable,Text,Text,IntWritable> {

        private final static IntWritable ONE=new IntWritable(1);
        private Text word=new Text();

        public void map(LongWritable key,Text value,Context context) throws IOException,InterruptedException{

            String line=value.toString();//READ LINE-BY-LINE
            String [] fields=line.split(",");//DECODE -CSV [" ", " ", " "]
            if(fields.length>5){
                String address=fields[5];
                if (address!=null){
                    String [] add=address.split(" ");
                    if (add.length>2){
                        String for_key=add[1];
                        word.set(for_key);
                        context.write(word,ONE);
                    }
                }
            }
        }
    }

    public static class infra_per_ku_REDUCE extends Reducer<Text,IntWritable,Text,IntWritable> {

        /*
         * REDUCE RETURN UNIQUE TITLE-ONE PER BLOCK. SO WE SHOULD GET UNIQUE IN REDUCE
         */

        public void reduce(Text key,Iterable<IntWritable> values,Context context) throws IOException,InterruptedException{

            int sum=0;

            for (IntWritable val:values) {

                sum+=val.get();

            }

            context.write(key, new IntWritable(sum));

        }
    }
}

Get Help Write Out Where Is Cut Text Justify Cur Pos Prev Page
```

## 4. 프로젝트 코드

### 4-3. SCALA 언어를 사용해 SPARK MLIB 이용한 상관분석(CORRELATION) 코드 및 결과

```
choirak0805@kmubigdata-cluster-m: ~ - Chrome
https://ssh.cloud.google.com/projects/sound-decoder-221702/zones/asia-northeast1-c/instances/kmubigdata-cluster-m?authuser=1&hl=ko&projectNumber=176045424343

type in expressions to have them evaluated.
type :help for more information.

scala> sc
res0: org.apache.spark.SparkContext = org.apache.spark.SparkContext@47f39279

scala> import org.apache.spark.mllib.linalg._
import org.apache.spark.mllib.linalg._

scala> import org.apache.spark.mllib.stat.Statistics
import org.apache.spark.mllib.stat.Statistics

scala> val sp=sc.parallelize(List(
  | Vectors.dense(3,3),
  | Vectors.dense(5,1),
  | Vectors.dense(7,3),
  | Vectors.dense(10,7),
  | Vectors.dense(11,12),
  | Vectors.dense(12,1),
  | Vectors.dense(12,5),
  | Vectors.dense(12,1),
  | Vectors.dense(14,2),
  | Vectors.dense(15,2),
  | Vectors.dense(18,5),
  | Vectors.dense(19,3),
  | Vectors.dense(21,66),
  | Vectors.dense(21,3),
  | Vectors.dense(25,12),
  | Vectors.dense(29,1),
  | Vectors.dense(32,5),
  | Vectors.dense(36,4),
  | Vectors.dense(36,8),
  | Vectors.dense(40,4),
  | Vectors.dense(41,7),
  | Vectors.dense(95,3),
  | Vectors.dense(138,17),
  | Vectors.dense(270,32),
  | Vectors.dense(635,29)
  | ))
sp: org.apache.spark.rdd.RDD[org.apache.spark.mllib.linalg.Vector] = ParallelCollectionRDD[0] at
parallelize at <console>:28

scala> val corr = Statistics.corr(sp)
corr: org.apache.spark.mllib.linalg.Matrix = 
1.0      0.4050482853110014
0.4050482853110014  1.0

scala>
```

LIST	PARALLELIZE	LOCAL	WORKER
(	,		)

→ : 0.4050