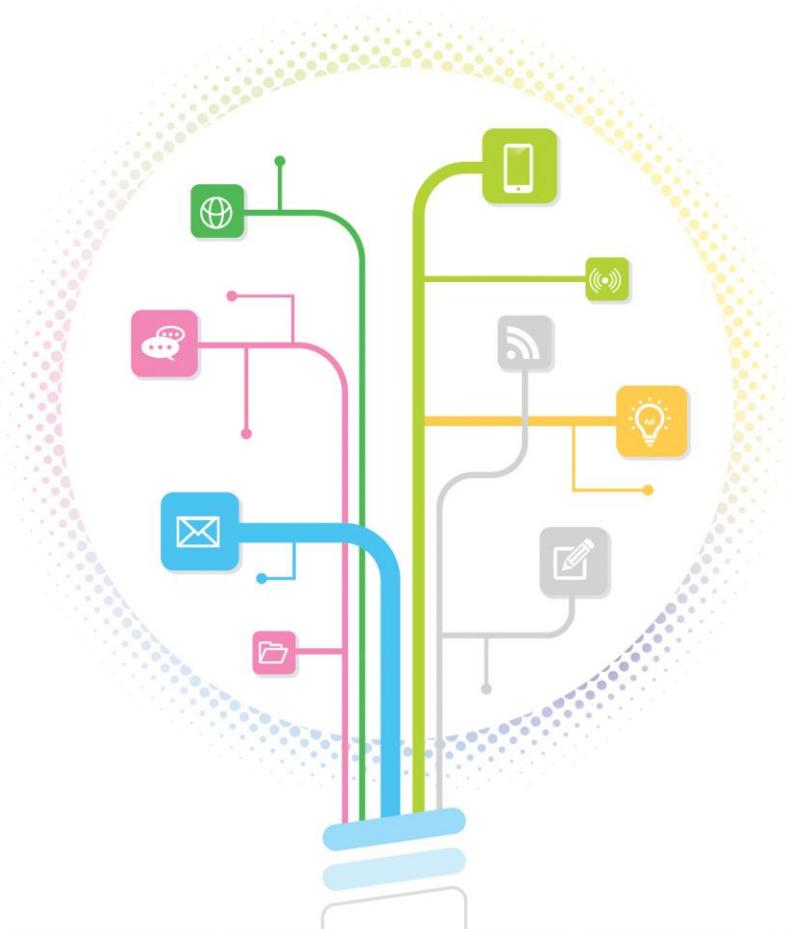


5 교통 데이터 분석 콘텐츠 활용 매뉴얼



미래창조과학부



한국정보화진흥원



CONTENTS

Beginning Level 초급과정

I 개요

개요	9
----	---

II 수집

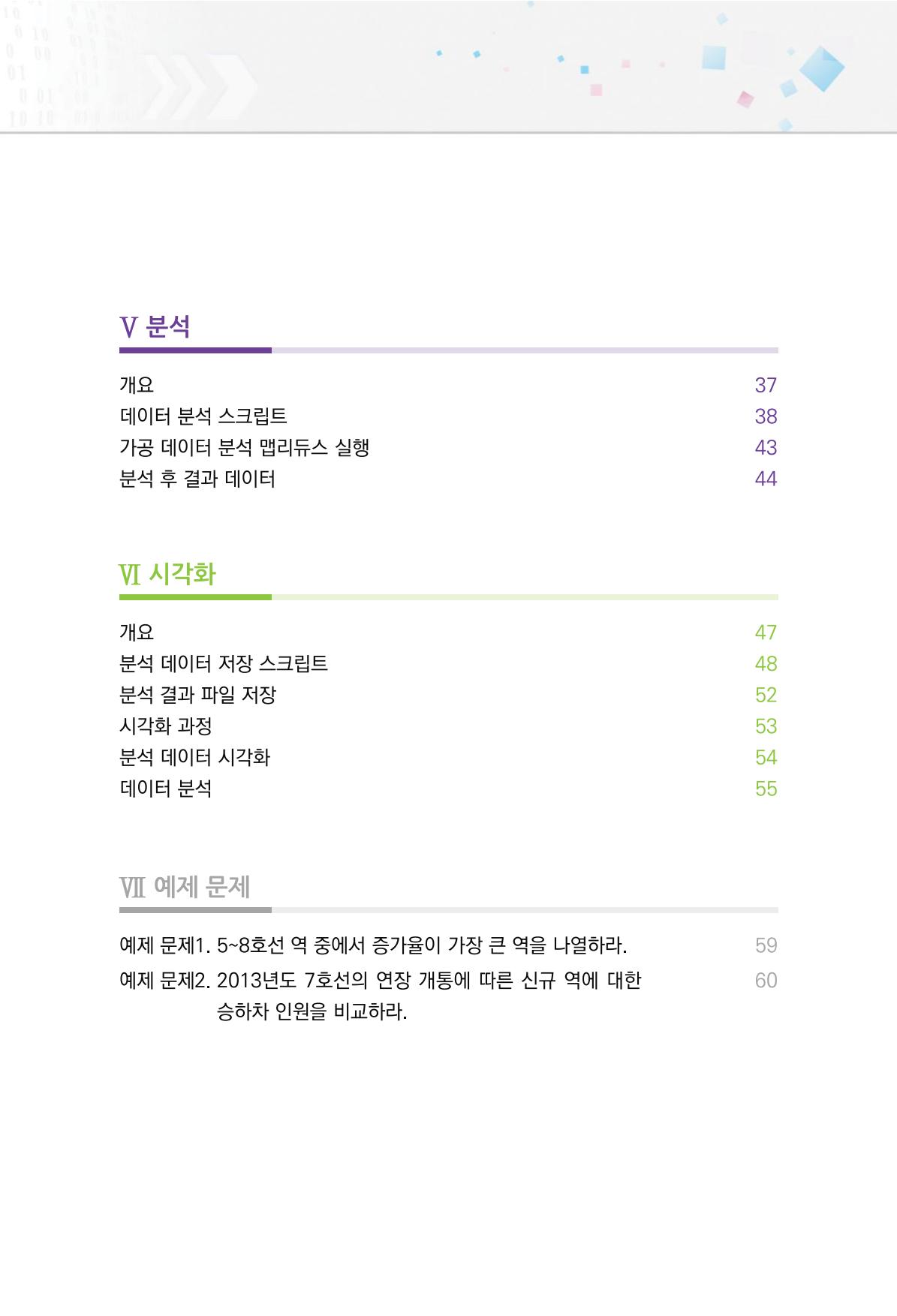
개요	13
교육용 데이터 샘플	14
데이터 수집	15
데이터 작업 영역 이동 스크립트	18

III 가공

개요	23
데이터 가공 스크립트	24

IV 저장

개요	29
가공 데이터 저장	30
몽고DB 저장 데이터 조회	32



V 분석

개요	37
데이터 분석 스크립트	38
가공 데이터 분석 맵리듀스 실행	43
분석 후 결과 데이터	44

VI 시각화

개요	47
분석 데이터 저장 스크립트	48
분석 결과 파일 저장	52
시각화 과정	53
분석 데이터 시각화	54
데이터 분석	55

VII 예제 문제

예제 문제1. 5~8호선 역 중에서 증가율이 가장 큰 역을 나열하라.	59
예제 문제2. 2013년도 7호선의 연장 개통에 따른 신규 역에 대한 승하차 인원을 비교하라.	60

CONTENTS

Intermediate Level **중급과정**

I 개요

개요	65
----	----

II 수집

개요	69
교육용 데이터 샘플	70
데이터 수집	72
데이터 작업 영역 이동 스크립트	75

III 가공

개요	79
데이터 가공 스크립트	80

IV 저장

개요	85
가공 데이터 하둡 파일시스템 업로드	86
가공 데이터 하둡 파일시스템 저장	87
하둡 파일시스템 파일 조회	88
하둡 명령어로 파일 조회	89

V 분석

개요	93
데이터 분석 스크립트	96
데이터 분석 맵리듀스 실행	98
분석 데이터 파일 조회	99
분석 후 결과 데이터	100

VI 시각화

개요	103
분석 데이터 저장 방법	104
시각화 과정	107
분석 데이터 시각화	108
데이터 분석	111

VII 예제 문제

예제 문제1. 여의도 벚꽃축제 기간의 승하차 인원의 증가율과 소셜 데이터를 연계하여 분석하라.	115
예제 문제2. 2009 ~2013년도까지 호선별 수송인원 점유율의 추이를 분석하라.	116



교통



Beginning Level

초급과정







I 개요

개요

9

8

I

개요

> 개요

서울도시철도공사에서 제공받은 2009~2013년 5~8호선 역별 승하차 정보 데이터를 바탕으로 2011~2013년 출퇴근 시간의 승/하차률이 높은 상위 5개 역을 기술 통계분석을 통해 추출하고 그 대상으로 승하차 인원의 변화 추이 분석과 전년대비 증감률에 대한 패턴 분석을 하는 방법을 학습한다. 이러한 방법으로 분석된 데이터를 바탕으로 모든 지하철역에 대한 출퇴근 시간 대의 승하차 패턴분석이 가능하고, 이를 바탕으로 일자리 밀도 지역을 파악해 볼 수 있고, 퇴근시 하차 인원이 많은 지역으로 상권의 발달을 예측해 볼 수 있다. 서울 도시철도공사는 시간대 별 승객 수송계획을 세울 수 있으며, 기상정보와 연동하여 승객의 승하차 변화 패턴을 예측할 수 있다.

> 활용 데이터

- **subway.csv** : 지하철 5~8호선 승하차 정보(2009~2014.6)

> 선행학습

- **오픈오피스** – 피벗테이블 기능, 차트 사용 방법
- **자바스크립트** – 객체(내장객체, 브라우저객체), 속성, 변수, 연산자(연산자 우선순위), 제어문, 함수(내장함수, 함수정의) 사용법
- **몽고 DB** – csv 파일 Import, 맵리듀스 실행 방법
- **D3 차트** – D3 라이브러리 사용법, 차트 설정 방법

▶ 요구사항

- 지하철 승하차 정보를 기반으로 2011, 2012, 2013년도 3년치 데이터 중 출근시간(06시~10시), 퇴근시간(18시~22시) 정보를 가지고 역별 승차, 하차의 인원이 많은 상위 5개 역을 추출하여 승차와 하차의 연도별 변화 추이를 분석하라.

▶ 분석 절차

- 수집된 2009년 ~2013년 5~8호선 승하차 정보 데이터를 로드한다.
- 제공된 지하철(5~8호선) 승하차 정보에서 2011~2013년 출근 시간대 승차 인원과 퇴근 시간대 하차 인원을 시계열 분석에 용이한 데이터 형태로 추출하여 저장한다.
- 2011~2013년 역별 출근 시간대 승차 인원과 퇴근 시간대 하차 인원을 패턴 분석이 용이한 형태로 합산을 하여 저장을 한다.
- 2011~2013년 역별 승하차 인원에 대한 패턴 분석을 하기 위해 맵리듀스 (분석 스크립트) 분석을 실행하고, 그 결과데이터를 활용하여 시계열 분석의 패턴분석을 할 수 있다.
- 분석된 결과 데이터를 엑셀 형식(csv)이나 D3챠트 형식(json)으로 파일로 저장한다.
- 저장된 파일을 불러와서 엑셀이나 D3 차트로 역별 승하차를 변화를 시각화해 본다.
- 2011~2013년 역별 승하차 인원의 변화 추이 분석과 전년대비 증감률에 대한 패턴 분석이 가능하다.



II 수집

개요	13
교육용 데이터 샘플	14
데이터 수집	15
데이터 작업 영역 이동 스크립트	18



수집

> 개요

교통 데이터는 서울도시철도공사에서 제공받은 2009년~2013년 5~8호선 승하차 정보 데이터를 분석에 필요한 정보(연도별 역별 시간대별 승차인원, 하차인원)를 수집/추출하여 분석 목적을 달성할 수 있는 한도 내에서 분석에 용이하게 편집하여 제공한다.

> 수집 방법

- **데이터 제공** : 교통 데이터는 서울도시철도공사에서 제공해 준 데이터를 OpenAPI, 자료수집기(Crawler)로 데이터를 수집하였고, 실습용 자료는 빅 데이터 분석 활용센터에 접속하여 교통 데이터 셋을 다운로드할 수 있도록 원시데이터를 제공하고 있다.

▶ 교육용 데이터 샘플

▶ 지하철 승하차 데이터(subway.csv)

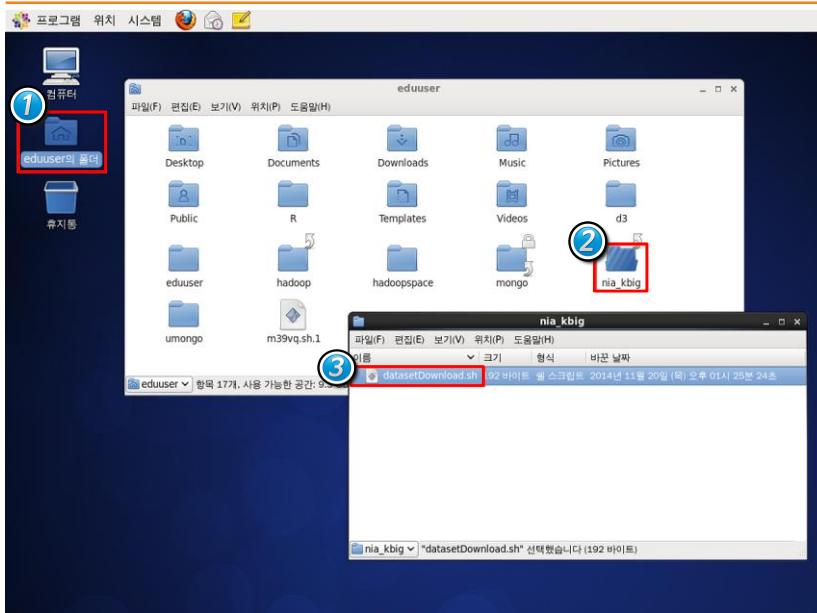
역코드	역명	일자	on_tot	on_05	on_06	on_07	off_11	off_22	off_23	off_24
2511	방화	20130101	4,403	87	145	274	338	338	338	338
2511	방화	20130102	8,467	177	580	383	397	397	397	397
2511	방화	20130103	8,165	160	517	412	375	375	375	375
2511	방화	20130104	8,656	168	526	477	425	425	425	425
2511	방화	20130105	6,236	123	201	425	538	538	538	538
2511	방화	20130106	4,627	94	130	387	365	365	365	365
2511	방화	20130107	8,820	189	571	425	423	423	423	423
2511	방화	20130108	8,702	179	533	480	445	445	445	445
2511	방화	20130109	8,491	183	498	464	403	403	403	403
2511	방화	20130110	8,719	166	537	434	405	405	405	405
2511	방화	20130111	8,930	151	510	458	436	436	436	436
2511	방화	20130112	6,709	140	209	487	554	554	554	554
2511	방화	20130113	4,788	79	134	406	337	337	337	337
2511	방화	20130114	8,730	198	575	458	417	417	417	417
2511	방화	20130115	8,782	174	529	467	389	389	389	389
2511	방화	20130116	8,883	178	516	459	402	402	402	402
2511	방화	20130117	8,750	169	518	438	428	428	428	428

II. 수집

> 데이터 수집(datasetDownload.sh)

- 데이터 저장소에서 서버 로컬로 교통 데이터 셋을 복사해 온다.
 - **subway.csv** : 지하철 승하차 데이터

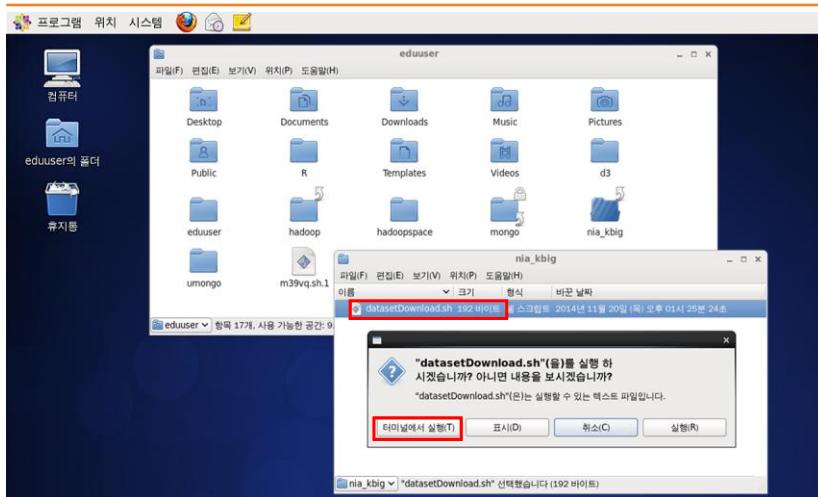
> 실습코드 디렉토리로 이동



- ① 로그인 후 바탕화면에서 eduuser 폴더를 오픈한다.
- ② nia_kbig 폴더를 오픈한다.
- ③ datasetDownload.sh를 더블클릭하여 실행한다.

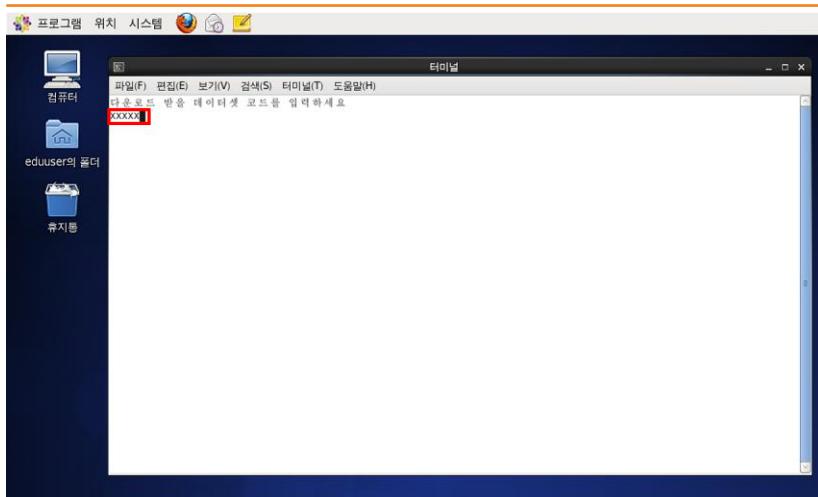
▶ 레파지토리에서 데이터 수집

datasetDownload.sh (원시데이터로 컬서버로 복사)



- '터미널에서 실행' 버튼을 클릭한다.

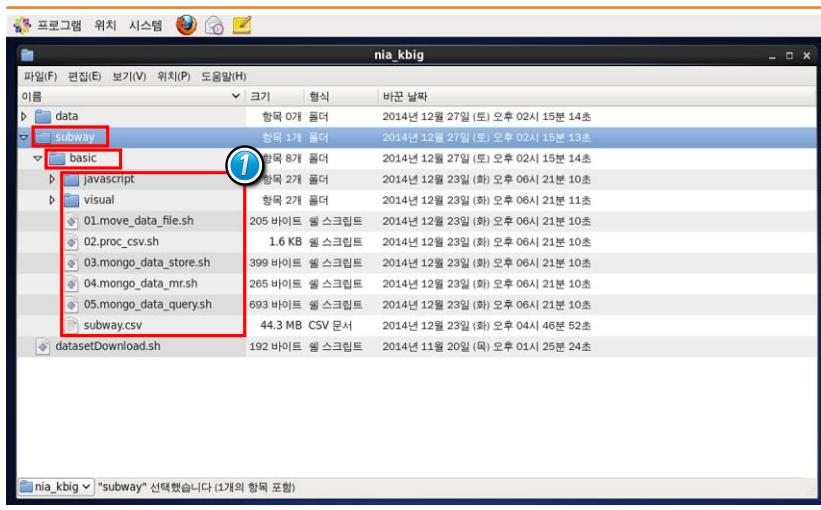
▶ 데이터셋 코드 입력



- 다운로드 받은 데이터셋 코드를 입력 후 엔터

II. 수집

▶ 데이터셋과 실습용 쉘 스크립트



- 실습용 데이터셋과 실습용 스크립트를 확인한다.

▶ ① 데이터 및 스크립트

▪ 01.move_data_file.sh :

작업영역 Data 폴더로 자료 이동하는 스크립트

▪ 02.proc_csv.sh :

원시데이터에서 분석할 대상을 추출하여 저장하는 스크립트

▪ 03.mongo_data_store.sh :

가공데이터를 MongoDB에 저장하는 스크립트

▪ 04.mongo_data_mr.sh :

가공데이터 분석 맵리듀스 실행 스크립트

▪ 05.mongo_data_query.sh :

분석데이터를 저장하는 실행 스크립트

▪ datasetDownload.sh :

레파지토리에서 분석데이터와 실습용 스크립트를 다운로드 스크립트

▪ subway.csv : 지하철 5~8호선 승하차 정보

> 데이터 작업 영역 이동 스크립트(01.move_data_file.sh)

> 데이터 작업 공간으로 이동

- 로컬로 수집해온 데이터를 작업 영역 Data 폴더로 자료를 이동하는 스크립트

01.move_data_file.sh (작업영역 폴더로 원시데이터 이동)

```

01.#!/bin/bash
02. # 복사 대상 파일 정의
03. TARGET_PRODUCT_PRICE=/home/eduuser/nia_kbig/subway/basic/subway
   ↛ .csv
04. # 작업 디렉토리 정의
05. LOCAL_DIR=/home/eduuser/nia_kbig/data/
06. # 작업영역 폴더로 이동
07. mv $TARGET_PRODUCT_PRICE $LOCAL_DIR
08. mv $TARGET_CODE $LOCAL_DIR
09.

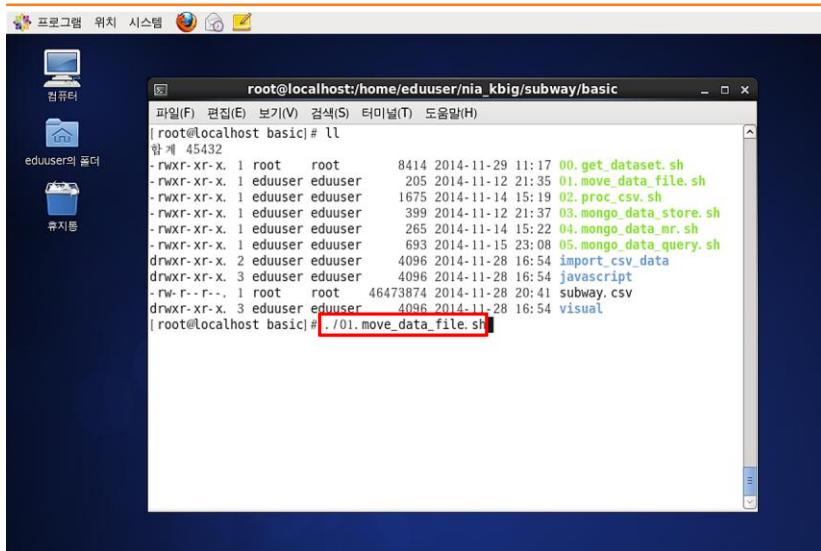
```



- 데이터 작업 영역 이동 스크립트 소스(01.move_data_file.sh)
- 라인 03 : 다운로드 받은 원시데이터 파일들의 위치(path)를 변수(TARGET_PRODUCT_PRICE)로 지정하는 라인이다.
- 라인 05 : 작업영역 디렉토리의 위치(path)를 변수(LOCAL_DIR)로 지정하는 라인이다.
- 라인 07~08 : mv 명령어를 사용하여 다운로드 받은 원시데이터 파일들을 작업영역 디렉토리로 이동시키는 라인이다.

II. 수집

▶ 수집 데이터 셋 작업 영역 폴더 이동



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "root@localhost basic". Inside the terminal, the command `ll` is run to list files in the current directory, which contains several shell scripts. Then, the command `./01.move_data_file.sh` is entered and executed.

```
root@localhost basic# ll
total 45432
drwxr-xr-x 1 root root 8414 2014-11-29 11:17 00.get_dataset.sh
-rw-r--r-- 1 eduuser eduuser 205 2014-11-12 21:35 01.move_data_file.sh
-rw-r--r-- 1 eduuser eduuser 1675 2014-11-14 15:19 02.proc_csv.sh
-rw-r--r-- 1 eduuser eduuser 399 2014-11-12 21:37 03.mongo_data_store.sh
-rw-r--r-- 1 eduuser eduuser 265 2014-11-14 15:22 04.mongo_data_mr.sh
-rw-r--r-- 1 eduuser eduuser 693 2014-11-15 23:08 05.mongo_data_query.sh
drwxr-xr-x 2 eduuser eduuser 4096 2014-11-28 16:54 import_csv_data
drwxr-xr-x 3 eduuser eduuser 4096 2014-11-28 16:54 javascript
drwxr-xr-x 1 root root 46473874 2014-11-28 20:41 subway.csv
drwxr-xr-x 3 eduuser eduuser 4096 2014-11-28 16:54 visual
root@localhost basic# ./01.move_data_file.sh
```

- 로컬에 원시데이터를 작업 영역 폴더로 이동 (/home/eduuser/nia_kbig/data/) 시킨다. `./01.move_data_file.sh` 입력 후 엔터

I. 개요

II. 수집

III. 가공

IV. 저장

V. 분석

VI. 시각화





III 가공

개요

23

데이터 가공 스크립트

24



가공

> 개요

작업 영역 폴더에 복사한 5~8호선 지하철 승하차 데이터에서 2011년 ~ 2013년까지 3년간의 데이터를 추출하여 CSV 파일(2011_2013_subway.csv) 형식으로 저장한다. 데이터 가공은 셀 스크립트를 사용하여 전체 데이터에서 필터링 기법을 적용하여 데이터를 가공한다.

> 가공 방법

- 지하철 승하차 데이터 (subway.csv) 파일에서 2011년 ~ 2013년까지 3년간의 지하철 승하차 인원 데이터를 추출하여 2011_2013_subway.csv 파일을 생성한다.

> 데이터셋

역코드	역명	일자	on_tot	on_05	on_06	on_24	off_11	off_22	off_23	off_24
2511	방화	20130101	4,403	87	145	274	338	338	338	338
2511	방화	20130102	8,467	177	580	383	397	397	397	397
2511	방화	20130103	8,165	160	517	412	375	375	375	375
2511	방화	20130104	8,656	168	526	477	425	425	425	425
2511	방화	20130105	6,236	123	201	425	538	538	538	538

> 데이터 가공 스크립트(02.proc_csv.sh)

- 셀 스크립트를 이용하여 2011~2013년도 데이터만을 추출 하여 2011_2013_subway.csv로 저장한다.

02.proc_csv.sh (원시데이터에서 분석할 대상을 추출 하여 저장)

```
01. #!/bin/bash
02. # 입력 CSV 파일 지정
03. INPUT_FILE='/home/eduuser/nia_kbig/data/subway.csv'
04. # 출력결과 CSV 파일 지정
05. OUTPUT_FILE='/home/eduuser/nia_kbig/data/2011_2013_subway.csv'
06. # HEADER컬럼 출력
07. echo "station,stat_name,income_date,on_tot,on_05,on_06,on_07,on_08,on_09,on_10,on_1
    ↪ 1,on_12,on_13,on_14,on_15,on_16,on_17,on_18,on_19,on_20,on_21,on_22,on_23,on_
    ↪ 24,off_tot,off_05,off_06,off_07,off_08,off_09,off_10,off_11,off_12,off_13,off_14,off_15,
    ↪ off_16,off_17,off_18,off_19,off_20,off_21,off_22,off_23,off_24" > $OUTPUT_FILE
08. # ''를 구분자로 해서 파일을 읽어들인다.
09. IFS=':'
10. while read station stat_name income_date on_tot on_05 on_06 on_07 on_08 on_09 on_1
    ↪ 0 on_11 on_12 on_13 on_14 on_15 on_16 on_17 on_18 on_19 on_20 on_21 on_22 on_
    ↪ _23 on_24 off_tot off_05 off_06 off_07 off_08 off_09 off_10 off_11 off_12 off_13 off_
    ↪ 14 off_15 off_16 off_17 off_18 off_19 off_20 off_21 off_22 off_23 off_24
11. do
12. # DATA가 2013,2012,2011 시작하는 년도인지 체크한다.
13.     is_valid=0
14.     if [[ $income_date == 2011* ]]; then
15.         is_valid=1
16.     elif [[ $income_date == 2012* ]]; then
17.         is_valid=1
18.     elif [[ $income_date == 2013* ]]; then
19.         is_valid=1
20.     fi
21.     if [[ $is_valid == 1 ]]; then
22.         # 해당년도의 데이터만을 CSV로 출력한다.
23.         echo "$station,$stat_name,$income_date,$on_tot,$on_05,$on_06,$on_07,$on_
    ↪ 08,$on_09,$on_10,$on_11,$on_12,$on_13,$on_14,$on_15,$on_16,$on_17
    ↪ ,$on_18,$on_19,$on_20,$on_21,$on_22,$on_23,$on_24,$off_tot,$off_05,$
    ↪ off_06,$off_07,$off_08,$off_09,$off_10,$off_11,$off_12,$off_13,$off_14,$
    ↪ off_15,$off_16,$off_17,$off_18,$off_19,$off_20,$off_21,$off_22,$off_23,$
    ↪ off_24" >> $OUTPUT_FILE
24.     fi
25. done < $INPUT_FILE
26.
```

I.개요

II.수집

III.가공

IV.저장

V.분석

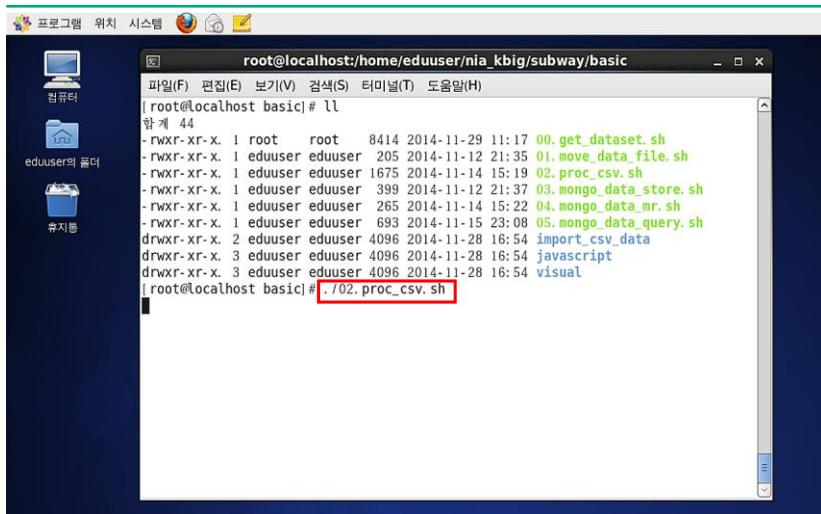
VI.시각화

III. 가공



- 24페이지 데이터 가공 스크립트 소스(02.proc_csv.sh)
- 라인 03~04 : subway.csv 가공 대상인 원시데이터 지정을 하고 가공 후 가공데이터를 2011_2013_subway.csv 파일로 저장하는 라인이다.
- 라인 07 : 가공데이터 파일(2011_2013_subway.csv)의 헤더 정보를 작성하는 라인이다.
- 라인 09~11 : 원시데이터에서 데이터를 읽어오는 라인이다.
- 라인 13~25 : 해당 연도가 2011, 2012, 2013년도 데이터만 선정을 하여 파일로 저장하는 라인이다.

▶ 원시데이터에서 분석 대상 데이터 가공



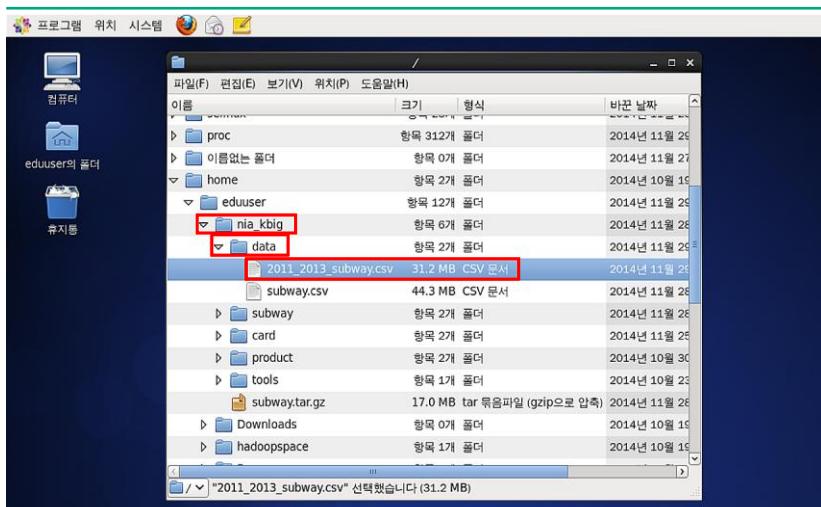
```

    컴퓨터 위치 시스템 도움말(H)
root@localhost:/home/eduuser/nia_kbig/subway/basic
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[ root@localhost basic]# ll
합계 44
-rwxr-xr-x. 1 root      root     8414 2014-11-29 11:17 00.get_dataset.sh
-rwxr-xr-x. 1 eduuser   eduuser  205 2014-11-12 21:35 01.move_data_file.sh
-rwxr-xr-x. 1 eduuser   eduuser 1675 2014-11-14 15:19 02.proc_csv.sh
-rwxr-xr-x. 1 eduuser   eduuser  399 2014-11-12 21:37 03.mongo_data_store.sh
-rwxr-xr-x. 1 eduuser   eduuser  265 2014-11-14 15:22 04.mongo_data_mr.sh
-rwxr-xr-x. 1 eduuser   eduuser  693 2014-11-15 23:08 05.mongo_data_query.sh
drwxr-xr-x. 2 eduuser   eduuser 4096 2014-11-28 16:54 import_csv_data
drwxr-xr-x. 3 eduuser   eduuser 4096 2014-11-28 16:54 javascript
drwxr-xr-x. 3 eduuser   eduuser 4096 2014-11-28 16:54 visual
[ root@localhost basic]# ./02.proc_csv.sh

```

- 원시 데이터 셋에서 분석할 데이터를 가공하여 2013_subway.csv 파일을 생성한다.
- ./02.proc_csv.sh 입력 후 엔터

▶ 가공 데이터 작업 영역 폴더에 생성



- /home/eduuser/nia_kbig/data 폴더에 2011_2013_subway.csv 파일이 생성된다.



IV 저장

개요	29
가공 데이터 저장	30
동고DB 저장 데이터 조회	32

IV

저장

> 개요

시계열 분석의 패턴 분석을 위해서 목표 대상과 분석할 범위가 지정된 가공 데이터(2011_2013_subway.csv)를 몽고DB에 subway 컬렉션을 만들어 저장을 한다. 몽고DB 컬렉션에 저장된 데이터는 몽고DB에서 자체 제공하는 맵리듀스 분석을 이용할 수 있다. 몽고DB는 메모리 특성을 가지고 있기 때문에 데이터 읽기, 쓰기 연산이 빠른 특징을 가지고 있고 스키마 제약이 없이 때문에 좀 더 유연하게 데이터를 처리할 수 있다.

> 저장 방법

- 2011~2013년치 지하철 승하차 데이터 파일(2011_2013_subway.csv)을 몽고DB에 Import 처리한다.
- 몽고에서 제공하는 CSV Import 툴인 mongoimport를 사용하여 몽고 DB에 원시데이터를 Import 한다.
- 몽고 DB에 들어가 있는 내용을 파악하기 위해서 /home/eduuser/nia_kbig/tools/umongo/ 폴더에 있는 lauch-umongo.sh 파일을 실행하여 입력된 데이터를 확인한다.

> 가공 데이터 저장(03.mongo_data_store.sh)

> MongoDB에 가공 데이터 저장 스크립트

- MongoDB로 저장할 CSV 파일을 mongoimport 명령어로 저장 처리를 한다.

03.mongo_data_store.sh (가공데이터를 MongoDB에 Import)

```

01.#!/bin/bash
02. # Import 파일 위치 경로
03. LOCAL_TARGET=/home/eduuser/nia_kbig/data/2011_2013_subway.csv
04. # mongo DB 접속정보 설정
05. MONGO_HOST=127.0.0.1
06. MONGO_PORT=27017
07. #mongo 데이터베이스명
08. MONGO_DATABASE=bigdata
09. #mongo 컬렉션 명
10. MONGO_COLLECTION=subway
11. #mongo Import 파일 형식
12. MONGO_IMPORT_FILE_TYPE=csv
13. # mongo DB로 가공데이터 Import 처리 명령어
14. mongoimport -h $MONGO_HOST --port $MONGO_PORT \
15.           -d $MONGO_DATABASE -c $MONGO_COLLECTION \
16.           --type $MONGO_IMPORT_FILE_TYPE --file $LOCAL_TARGET --head
17.           ↵ erline

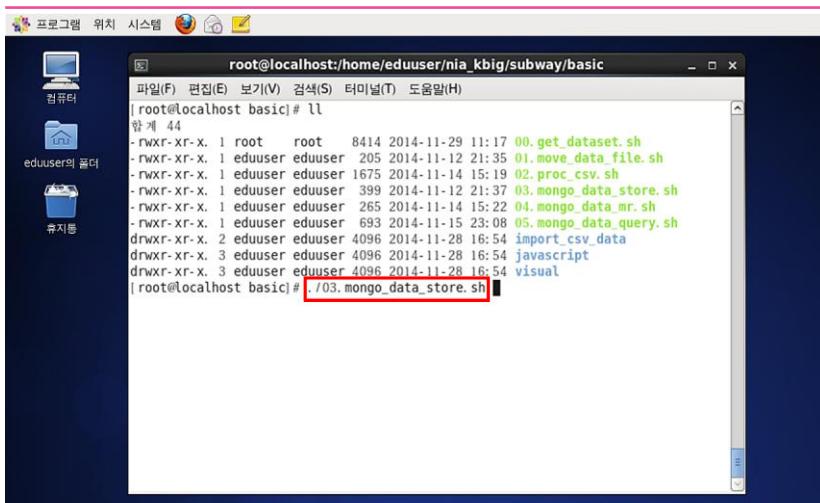
```



- 가공 데이터 저장 스크립트 소스(03.mongo_data_store.sh)
- 라인 03 : MongoDB에 입력할 가공데이터를 지정하는 라인이다.
- 라인 05~06 : 로컬 서버에 있는 MongoDB에 접속을 설정하는 라인이다.
- 라인 08~12 : bigdata 데이터베이스를 정의하고 컬렉션으로 subway를 지정하는 라인이다. MongoDB에 입력되는 데이터 파일 형식이 csv 파일로 지정하는 라인이다.
- 라인 14~16 : mongoimport 명령어에 접속호스트, 포트, 데이터베이스, 컬렉션, 타입을 지정하고, 가공된 데이터(2011_2013_subway.csv)를 MongoDB에 import 처리를 하는 라인이다.

IV. 저장

▶ 가공 데이터 MongoDB에 저장



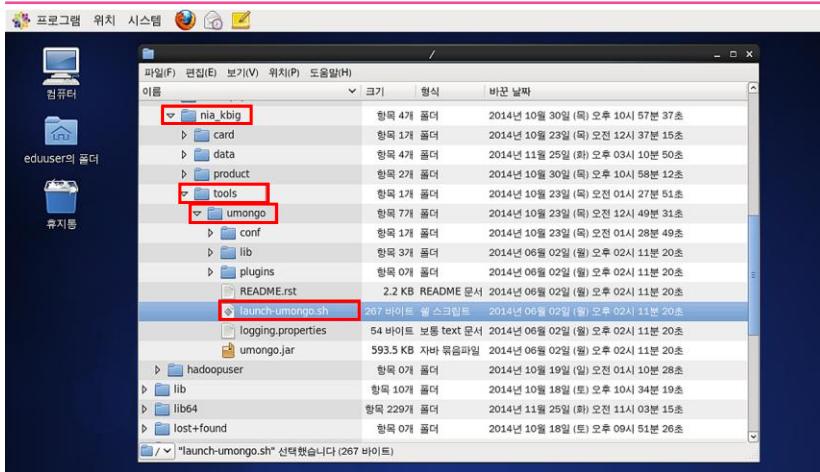
The screenshot shows a terminal window titled "root@localhost:/home/eduuser/nia_kbig/subway/basic". The window displays a file listing with the command "ll". The output shows several shell scripts with timestamps and permissions. A red box highlights the command ". ./03.mongo_data_store.sh".

```
root@localhost basic] # ll
합계 44
-rwxr-Xr-x. 1 root      root    8414 2014-11-29 11:17 00.get_dataset.sh
-rwxr-Xr-x. 1 eduuser   eduuser  205 2014-11-12 21:35 01.move_data_file.sh
-rwxr-Xr-x. 1 eduuser   eduuser 1675 2014-11-14 15:19 02.proc_csv.sh
-rwxr-Xr-x. 1 eduuser   eduuser  399 2014-11-12 21:37 03.mongo_data_store.sh
-rwxr-Xr-x. 1 eduuser   eduuser  265 2014-11-14 15:22 04.mongo_data_mr.sh
-rwxr-Xr-x. 1 eduuser   eduuser  693 2014-11-15 23:08 05.mongo_data_query.sh
drwxr-Xr-x. 2 eduuser   eduuser 4096 2014-11-28 16:54 import_csv_data
drwxr-Xr-x. 3 eduuser   eduuser 4096 2014-11-28 16:54 javascript
drwxr-Xr-x. 3 eduuser   eduuser 4096 2014-11-28 16:54 visual
[ root@localhost basic] # . ./03.mongo_data_store.sh
```

- 원시 데이터셋을 분석을 위해서 MongoDB에 원시데이터를 저장한다.
- `./03.mongo_data_store.sh`를 입력 후 엔터

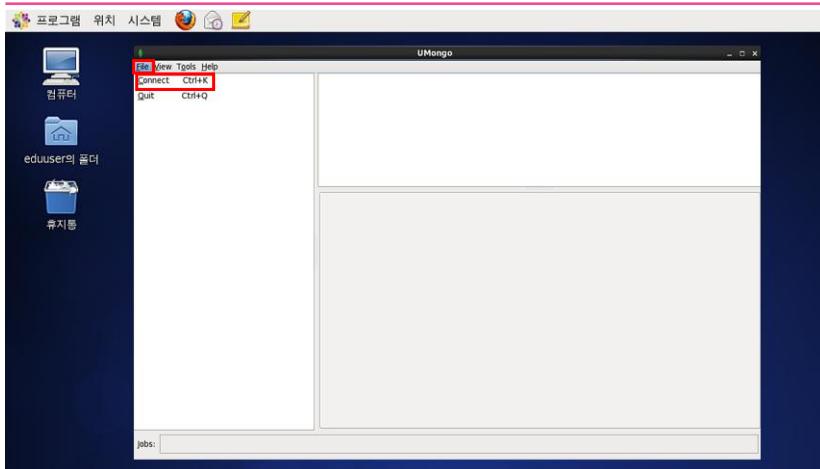
> MongoDB 저장 데이터 조회

> umongo 툴 실행



- /home/eduuser/lia_kbig/tools/umongo/lauch-umongo.sh 더블클릭하여
‘터미널에서 실행’ 버튼을 클릭하여 MongoDB 관리 툴을 실행한다

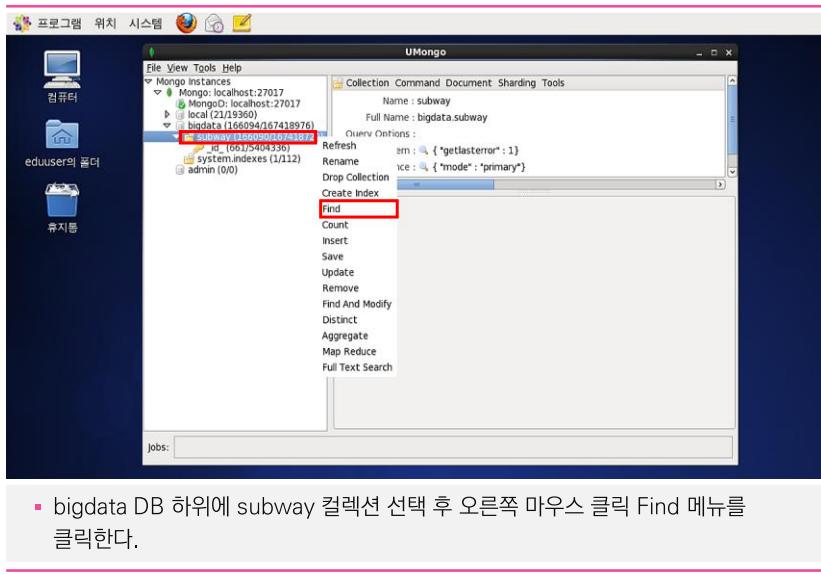
> MongoDB 접속



- File 메뉴 > Connect 메뉴 클릭한다.
- Default를 선택하여 OK 버튼을 클릭한다.

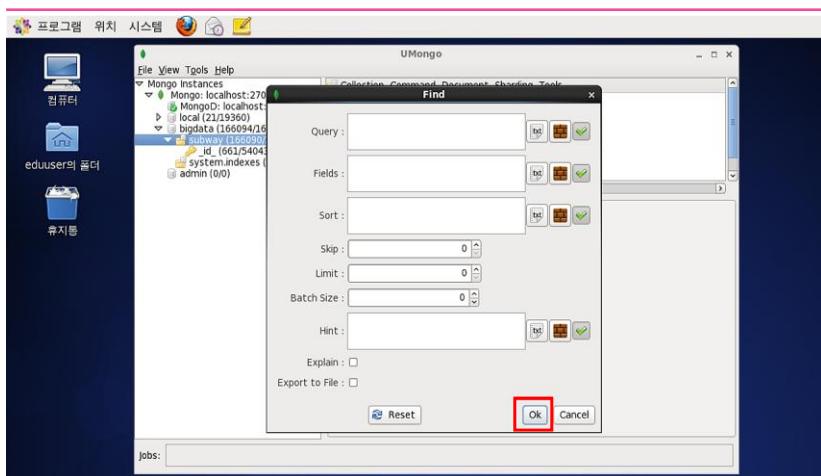
IV. 저장

> subway 컬렉션 선택



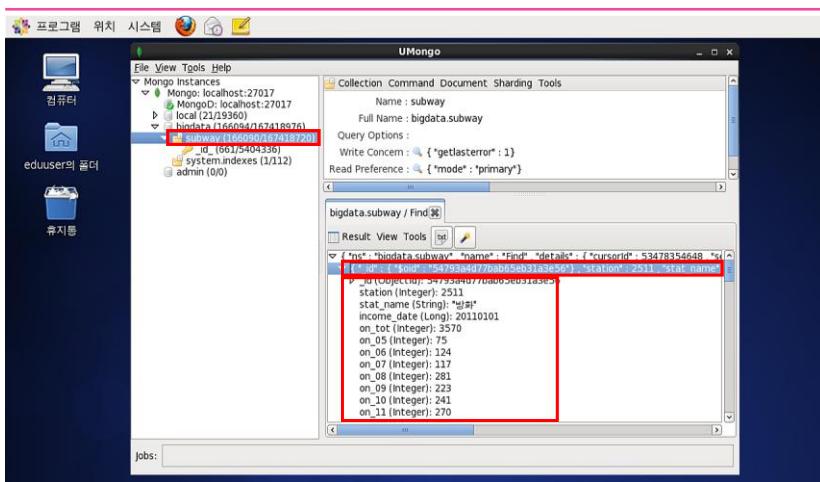
- bigdata DB 하위에 subway 컬렉션 선택 후 오른쪽 마우스 클릭 Find 메뉴를 클릭한다.

> subway 컬렉션 Find



- Find 창에서 OK 버튼을 클릭한다.

➤ subway 컬렉션 데이터 목록



- bigdata MongoDB의 subway 컬렉션에 입력된 데이터 목록이 출력되고 목록을 클릭하면 데이터셋 내용을 확인할 수 있다.

I. 개요

II. 수집

III. 기공

IV. 저장

V. 분석

VI. 시각화

W





V 분석

개요	37
데이터 분석 스크립트	38
가공 데이터 분석 맵리듀스 실행	43
분석 후 결과 데이터	44

V

분석

> 개요

교통 데이터(지하철 승하차)의 분석은 MongoDB 컬렉션(subway)에 저장되어 있는 데이터를 가지고 역별 출근 시간대(오전 6시~9시)와 퇴근 시간대(오후 6시~10시) 사이를 기준으로 연도별로 분류를 하기 위해 MongoDB 분석 스크립트로 작성한다. 패턴 분석에 용이한 형태로 연도별로 출근 시간대, 퇴근 시간대로 인원을 합산을 하여 저장한다.

> 분석 방법

- MongoDB의 subway 컬렉션에 저장되어 있는 2011-2013년도 지하철 승하차 데이터를 맵리듀스 작업을 통해서 출근시간(오전 6~9시) 승차인원, 퇴근시간(오후 6시~10시) 하차 인원의 연도/역별 데이터를 뽑아낸다.
- 연도별 승차, 하차 인원의 상위 5개역을 추출하여 저장한다.

> 가공 데이터

역코드	역명	일자	on_tot	on_05	on_06	on_24	off_11	off_22	off_23	off_24
2511	방화	20130101	4,403	87	145	274	338	338	338	338
2511	방화	20130102	8,467	177	580	383	397	397	397	397
2511	방화	20130103	8,165	160	517	412	375	375	375	375
2511	방화	20130104	8,656	168	526	477	425	425	425	425
2511	방화	20130105	6,236	123	201	425	538	538	538	538

> 데이터 분석 스크립트(04.mongo_data_mr.sh)

> 가공 데이터 분석 실행 스크립트

- MongoDB는 맵리듀스를 자바스크립트로 처리하여 자바스크립트 파일로 작성한 후 MongoDB 명령어로 실행시켜서 결과를 추출한다.

04.mongo_data_mr.sh (가공데이터 분석 맵리듀스 실행)

```

01.#!/bin/bash
02.#Mongo DB 접속
03.MONGO_HOST=127.0.0.1
04.MONGO_PORT=27017
05.MONGO_DATABASE=bigdata
06.# 지하철 년도/역별 승하차인원 산출 M/R 모듈
07.EXECUTE_JS_MODULE=javascript/subway_rush_hour.js
08.# Mongo에서 M/R을 실행하는 명령어
09.mongo --host $MONGO_HOST --port $MONGO_PORT $MONGO_DATABASE
    ↳ SE $EXECUTE_JS_MODULE
10.

```



- 가공데이터 분석 맵리듀스 실행 스크립트 소스(04.mongo_data_mr.sh)
- 라인 03~05 : 로컬 서버에 있는 MongoDB에 접속정보를 설정하는 라인이다.
- 라인 07 : javascript 폴더에 있는 분석스크립트 파일(subway_rush_hour.js)을 맵리듀스 분석 파일로 지정하는 라인이다.
- 라인 09 : mongo 명령어에 접속호스트, 포트, 데이터베이스를 지정하고 가공된 데이터를 MongoDB에서 맵리듀스 분석 작업을 수행하는 라인이다.

> 맵리듀스 분석 스크립트(subway_rush_hour.js)

```

01. /**
02.  * [subway] 콜렉션을 대상으로 '년도','여명','출근(06~09시)시 총승차인원', '퇴근(18~22시)시 총하
03.  ↳ 차인원을 구한다.
04.  */
05. // MapReduce 대상 콜렉션을 선정한다.
06. var subway = db.subway;
07. // Map함수를 선언한다.
08. var map = function() {
09. // '년도(YYYY):여명:출근|퇴근'를 키로 만든다.
10.   var income_date = this.income_date + ""; // 문자열로 인식 시킴
11.   var year = income_date.substring(0, 4); // YYYY
12.   var key1 = year + ":" + this.stat_name + ":출근"; // '년도:여명:출근' 으로 키로 만든다.
13.   // 오전 6시에서 9시까지의 총승차인원
14.   var on_off_count = 0;
15.   if (this.on_06 != "" && this.on_06 != "-") {
16.     var value = this.on_06 + "";
17.     on_off_count += Number(value.replace(",",""));
18.   }
19.   if (this.on_07 != "" && this.on_07 != "-") {
20.     var value = this.on_07 + "";
21.     on_off_count += Number(value.replace(",",""));
22.   }
23.   if (this.on_08 != "" && this.on_08 != "-") {
24.     var value = this.on_08 + "";
25.     on_off_count += Number(value.replace(",",""));
26.   }
27.   if (this.on_09 != "" && this.on_09 != "-") {
28.     var value = this.on_09 + "";
29.     on_off_count += Number(value.replace(",",""));
30.   }
31.   // 출근시간대 승차 인원 총합
32.   var curDoc1 = new Object;
33.   curDoc1.year = year;
34.   curDoc1.stat_name = this.stat_name;
35.   curDoc1.on_off_type = "출근";

```

```

36.    //curDoc.count = 1;
37.    emit(key1, curDoc);
38.    var key2 = year + ":" + this.stat_name + ":퇴근"; // '년도:역명:퇴근' 으로 키로 만든다.
39.    // 오후 18시에서 22시까지의 총하차 인원
40.    var on_off_count2 = 0;
41.    if (this.off_18 != "" && this.off_18 != "-") {
42.        var value = this.off_18 + "";
43.        on_off_count2 += Number(value.replace(",",""));
44.    }
45.    if (this.off_19 != "" && this.off_19 != "-") {
46.        var value = this.off_19 + "";
47.        on_off_count2 += Number(value.replace(",",""));
48.    }
49.
50.
51.    if (this.off_20 != "" && this.off_20 != "-") {
52.        var value = this.off_20 + "";
53.        on_off_count2 += Number(value.replace(",",""));
54.    }
55.    if (this.off_21 != "" && this.off_21 != "-") {
56.        var value = this.off_21 + "";
57.        on_off_count2 += Number(value.replace(",",""));
58.    }
59.    if (this.off_22 != "" && this.off_22 != "-") {
60.        var value = this.off_22 + "";
61.        on_off_count2 += Number(value.replace(",",""));
62.    }

```



- 맵리듀스 분석 스크립트 소스(subway_rush_hour.js)
- 라인 05 : 몽고DB의 subway 컬렉션을 설정을 하는 라인이다.
- 라인 09~11 : 날짜를 읽어서 년도를 분리해서 년도+역명+"출근"으로 키를 생성하는 라인이다.
- 라인 13~29 : 오전 6시 부터 오전 9시까지 시간대별 승차 인원을 구해서 총 승차 인원을 합산하는 라인이다.
- 라인 31~35 : 출근 시간대 승차 인원의 년도, 역, 출근, 종합계를 객체에 저장하는 라인이다.
- 라인 37 : key1에 해당되는 부분을 실행하여 분석을 실행하는 라인이다.
- 라인 38 : 년도+역명+"퇴근"으로 키를 생성하는 라인이다.
- 라인 40~62 : 오후 18시 부터 오후 22시까지 시간대별 하차 인원을 구해서 총 하차 인원을 합산하는 라인이다.

I. 개요

II. 수집

III. 가공

IV. 저장

V. 분석

VI. 시각화

V. 분석

```
63.      // 퇴근시간대 하차인원 총합
64.      var curDoc2 = new Object;
65.      curDoc2.year = year;
66.      curDoc2.stat_name = this.stat_name;
67.      curDoc2.on_off_type = "퇴근";
68.      curDoc2.on_off_count = on_off_count2 ;
69.      //curDoc.count = 1;
70.      emit(key2, curDoc2);
71.  };
72.
73.  // Reduce함수를 선언한다.
74.  var reduce = function(key, on_off_infos) {
75.      // map에서 생성한 도큐먼트와 동일한 JSON 형태로 만들어주어야 한다.
76.      var reduced = {year: "", stat_name:"", on_off_type : "", on_off_count : 0};
77.      // 동일한 키를 가지는 아이템들을 일순하면서 출퇴근 인원의 총합을 구한다.
78.      on_off_infos.forEach(function(on_off) {
79.          reduced.year = on_off.year;
80.          reduced.stat_name = on_off.stat_name;
81.          reduced.on_off_type = on_off.on_off_type;
82.          reduced.on_off_count += on_off.on_off_count;
83.      });
84.      return reduced;
85.  };
86.
87.  // 콜렉션에 M/R 작업을 건다.
88.  subway.mapReduce(
89.      map,
90.      reduce,
91.      { out: 'subway_mr_result'}, // out에 지정한 'products_mr_result' 콜렉션에 결과 데이터가
92.      // 저장된다.
93.      function(err, coll) {
94.          coll.find().toArray(function(err, arr) {
95.              console.log(arr);
96.          });
97.      });
98. }
```



- 39페이지 맵리듀스 분석 스크립트 소스(subway_rush_hour.js)
- 라인 64~70 :퇴근 시간대 하차 인원의 년도, 역, 퇴근, 총합계를 객체에 저장하고 key2에 해당하는 분석을 실행하는 라인이다.
- 라인 74 :리듀스 함수를 정의하는 라인이다.
- 라인 76 :map 함수에서 생성한 도큐먼트와 동일한 json 형태의 구조와 동일한 형태로 만들어 준다. reduced 함수 형태를 선언하는 라인이다.
- 라인 78~84 :동일한 key를 가지고 있는 항목을 루프 돌면서 출퇴근 승하차 인원의 총합을 구하고 합계를 반환하는 라인이다.
- 라인 88~96 :subway컬렉션에 mapReduce 분석을 실행을 수행하고 결과 데이터를 'subway_mr_result' 컬렉션에 저장하는 라인이다.

I. 개요

II. 수집

III. 가공

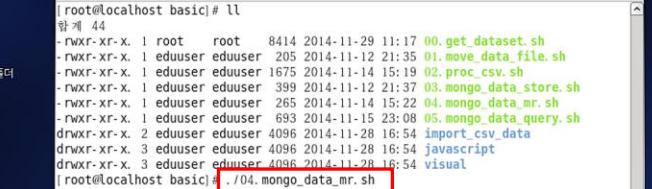
IV. 저장

V. 분석

VI. 시각화

> 가공 데이터 분석 맵리듀스 실행

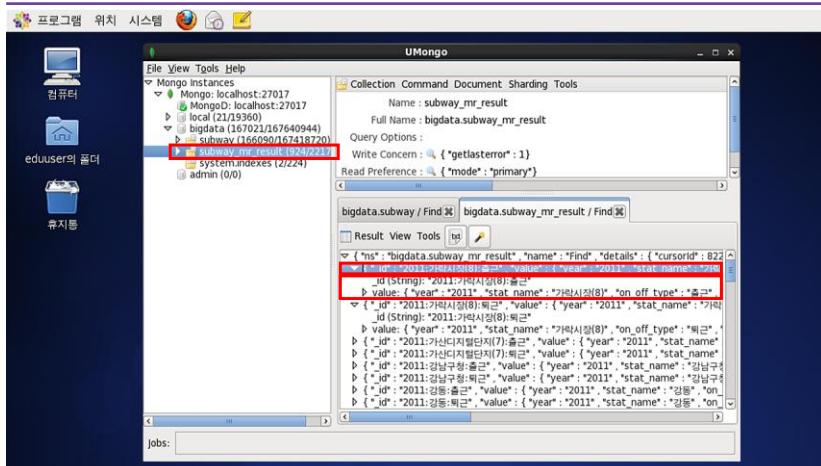
▶ 분석 맵리듀스 실행



```
root@localhost:/home/eduuser/nia_kbgi/subway/basic
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[root@localhost basic]# ll
합계 44
-rwxr-xr-x 1 root root 8414 2014-11-29 11:17 00.get_dataset.sh
-rwxr-xr-x 1 eduuser eduuser 205 2014-11-12 21:35 01.move_data_file.sh
-rwxr-xr-x 1 eduuser eduuser 1675 2014-11-14 15:19 02.proc_csv.sh
-rwxr-xr-x 1 eduuser eduuser 399 2014-11-12 21:37 03.mongo_data_store.sh
-rwxr-xr-x 1 eduuser eduuser 265 2014-11-14 15:22 04.mongo_data_mr.sh
-rwxr-xr-x 1 eduuser eduuser 693 2014-11-15 23:00 05.mongo_data_query.sh
drwxr-xr-x 2 eduuser eduuser 4096 2014-11-28 16:54 import_csv_data
drwxr-xr-x 3 eduuser eduuser 4096 2014-11-28 16:54 javascript
drwxr-xr-x 3 eduuser eduuser 4096 2014-11-28 16:54 visual
[root@localhost basic]# ./04.mongo_data_mr.sh
MongoDB shell version: 2.6.5
connecting to: 127.0.0.1:27017/bigdata
```

- 몽고 DB에 입력된 원시데이터에서 분석하여 결과를 출력한다.
`./04.mongo_data_mr.sh` 입력 후 엔터

➤ 분석 결과 데이터 확인



- subway_mr_result 컬렉션 선택 후 마우스 오른쪽 클릭 후 find 팝업 메뉴를 클릭하면 입력된 데이터 목록을 확인할 수 있다.

> 분석 후 결과 데이터

> card_mr_result 데이터셋

```

01. /* 0 */
02. {
03.   "_id" : "2011:가락시장(8)",
04.   "value" : {
05.     "year" : "2011",
06.     "stat_name" : "가락시장(8)",
07.     "on_off_count" : 1368978.0
08.   }
09. }
10. /* 1 */
11. {
12.   "_id" : "2011:가산디지털단지(7)",
13.   "value" : {
14.     "year" : "2011",
15.     "stat_name" : "가산디지털단지(7)",
16.     "on_off_count" : 8106089.0
17.   }
18. }
19. /* 2 */
20. {
21.   "_id" : "2011:강남구청",
22.   "value" : {
23.     "year" : "2011",
24.     "stat_name" : "강남구청",
25.     "on_off_count" : 4513967.0
26.   }
27. }
28. /* 3 */
29. {
30.   "_id" : "2011:강동",
31.   "value" : {
32.     "year" : "2011",
33.     "stat_name" : "강동",
34.     "on_off_count" : 3205570.0
35.   }
36. }
37.

```



1

2

VI 시각화

개요	47
분석 데이터 저장 스크립트	48
분석 결과 파일 저장	52
시각화 과정	53
분석 데이터 시각화	54
데이터 분석	55

VI

시각화

> 개요

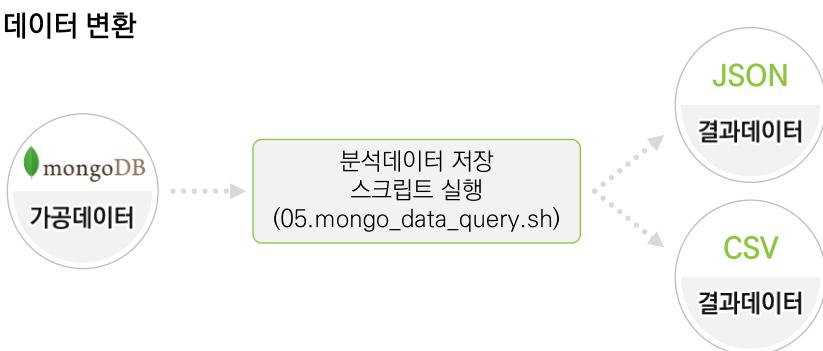
교통 데이터(지하철 승하차) 분석 과정에서 봉고DB에 저장된 데이터를 시각화하기 위해서 CSV, JSON 형태의 파일로 결과 데이터를 저장한다. 교통 데이터에서 분석된 연도별, 역별 승차 정보 상위 5개 역과 하차 상위 5개 역 정보 데이터를 D3 차트 라이브러리를 활용하여 막대그래프 형태로 시각화한다. 2011~2013년 역별 승하차 인원의 변화 추이 분석과 전년대비 증감률에 대한 패턴 분석이 가능하다.



> 시각화 방법 및 활용기술

- 2011~2013년도 역명별 지하철 승하차 인원 데이터 중에서 각 연도별로 출근시간대 승차인원 상위 5역과 퇴근시간대 하차 인원 상위 5역을 시각화에 사용할 형태로 추출한다.
- 가공 변환된 데이터를 JSON 파일과 엑셀 CSV로 저장한다.
- 연도별로 그룹화하여 D3 막대그래프를 활용하여 시각화한다.

> 데이터 변환



> 분석 데이터 저장 스크립트(04.mongo_data_mr.sh)

> MongoDB 분석 결과 데이터 저장

- MongoDB에 저장된 데이터를 파일 형식으로 저장한다.

터미널 커맨드 창에서 **04.mongo_data_mr.sh** 입력 후 엔터

04.mongo_data_mr.sh (분석데이터를 저장하는 커맨드)

```

01.#!/bin/bash
02.MONGO_HOST=127.0.0.1
03.MONGO_PORT=27017
04.MONGO_DATABASE=bigdata
05.OUTPUT_JS_FILE=
    ↪ /home/eduuser/nia_kbig/subway/basic/manual_result/subway_data.js
06.OUTPUT_CSV_FILE
    ↪ /home/eduuser/nia_kbig/subway/basic/manual_result/subway_data.csv
07.# M/R 결과 컬렉션에서 돼지고기 , 상추 평균가격변화 데이터 조회 모듈(JSON
    ↪ format)
08.EXECUTE_JS_MODULE=subway_mr_result_query.js
09.mongo --host $MONGO_HOST --port $MONGO_PORT --quiet --eval "var p
    ↪ aram='json'" $MONGO_DATABASE $EXECUTE_JS_MODULE > $OUTPUT_J
        S_FILE
10.# M/R 결과 컬렉션에서 돼지고기 , 상추 평균가격변화 데이터 조회 모듈(CSV format)
11.mongo --host $MONGO_HOST --port $MONGO_PORT --quiet --eval "var p
    ↪ aram='csv'" $MONGO_DATABASE $EXECUTE_JS_MODULE > $OUTPUT_C
        SV_FILE
12.

```



부연설명

- 분석 데이터 저장 스크립트 소스(**05.mongo_data_query.sh**)
- 라인 02~04 : MongoDB 접속 정보를 설정하는 라인이다.
- 라인 05~06 : MongoDB에서 분석결과 데이터를 저장할 파일 형태를 지정하는 라인이다.
subway_data.js, subway_data.csv 2가지 형태 파일로 설정하는 라인이다.
- 라인 08 : 맵리듀스를 실행하는 스크립트(subway_mr_result_query.js)를 지정하는 라인이다.
- 라인 09~11 : mongo 명령어를 사용하여 호스트, 포트, 데이터형식, 데이터베이스,
저장 스크립트 설정하고 저장파일을 지정하여 데이터를 저장하는 라인이다.

▶ 분석 결과 출력 스크립트

javascript/subway_mr_result_query.js

```

01. /**
02. * [subway_mr_result] 콜렉션을 조회한다.
03. * 결과값은 JSON ARRAY 또는 CSV 형태로 출력한다.
04. */
05. // 대상 콜렉션을 선정한다.
06. var subway_mr_result = db.subway_mr_result;
07. function getRankingList(targetYear) {
08.     var ranks = [];
09.     // 연도별로 인원이 가장 많은 역을 기준으로 10개 조회한다.
10.     subway_mr_result.find({ "value.year" : targetYear }).sort({"value.on_off_c
    ↴ ount": -1}).limit(10).forEach(get_results);
11.     var ranking = 1;
12.     function get_results (result) {
13.         var rank = new Object();
14.         rank.stat_name = result.value.stat_name;
15.         rank.on_off_count = result.value.on_off_count;
16.         rank.ranking = ranking;
17.         ranks.push(rank);
18.         ranking++;
19.     }
20.     return ranks;
21. }
22. function printResultCSV(targetYear) {
23.     var startIdx = 0;
24.     // 연도별로 인원이 가장 많은 역을 기준으로 10개 조회한다.
25.     subway_mr_result.find({ "value.year" : targetYear }).sort({"value.on_off_c
    ↴ ount": -1}).limit(10).forEach(get_results);
26.     // 조회결과를 callback처리하며 CSV 포맷을 맞춘다.
27.     function get_results (result) {
28.         var stat_name = result.value.stat_name;
29.         var on_off_count = result.value.on_off_count;
30.         print( targetYear + "," + (startIdx + 1) + "," + stat_name + "," + on_off_
    ↴ count);
31.         startIdx++;
}

```

```

32.     }
33. }
34. // 년도를 지정하여 결과 값을 Javascript JSON Array로 출력한다.
35. function printAsJSON() {
36.     var ranks_1 = getRankingList("2011");
37.     var ranks_2 = getRankingList("2012");
38.     var ranks_3 = getRankingList("2013");
39.     print("var data = ");
40.     print("[");
41.     for (var i = 0 ; i < 10 ; i++) {
42.         if (i != 0) {
43.             print(",");
44.         }
45.         print("}");
46.         print(' "item" : "' + (i+1) + '위' , );
47.         print(' "Data" : [ ');
48.         print(' { ');
49.         print(' "year" : "2013" , );
50.         print(' "stat_name" : "' + ranks_3[i].stat_name + " , ");
51.         print(' "on_off_count" : ' + ranks_3[i].on_off_count );
52.         print(' },');
53.         print(' { ');
54.         print(' "year" : "2012" , );
55.         print(' "stat_name" : "' + ranks_2[i].stat_name + " , ");
56.         print(' "on_off_count" : ' + ranks_2[i].on_off_count );
57.         print(' },');
58.         print(' ');

```



- 분석 결과 출력 스크립트 소스(javascript/subway_mr_result_query.js)
- 라인 06 : 맵리듀스 분석결과를 저장할 컬렉션을 설정하는 라인이다.
- 라인 07~21 : 년도별 인원이 가장 많음을 기준으로 10개역을 조회하는 라인이다.
- 라인 22~33 : 결과 데이터를 CSV파일로 저장하는 라인이다.
- 라인 35~38 : 결과 데이터를 년도별로 지정하고 JSON 형태로 저장하는 라인이다.
- 라인 39~68 : 연도별 1위~10까지 역명, 승하차 인원의 합계를 JSON 형태로 저장하는 라인이다.

I. 개요

II. 수집

III. 가공

IV. 저장

V. 분석

VI. 시각화

VI. 시각화

```
59.         print(' "year" : "2011" ,);
60.         print(' "stat_name" : "' + ranks_1[i].stat_name + " ,");
61.         print(' "on_off_count" : ' + ranks_1[i].on_off_count );
62.         print(' },);
63.         print(']');
64.         print('}');
65.     }
66.     print("]");
67.     print(":");
68. }
69. // 년도를 지정하여 결과 값을 CSV 로 출력한다.
70. function printAsCSV() {
71.     print("year,ranking,stat_name,on_off_count");
72.     printResultCSV("2011");
73.     printResultCSV("2012");
74.     printResultCSV("2013");
75. }
76. // mongo shell에서 parameter로 지정한 출력 포맷에 따라서 처리한다.
77. if (param == 'json') {
78.     printAsJSON();
79. } else {
80.     printAsCSV();
81. }
```

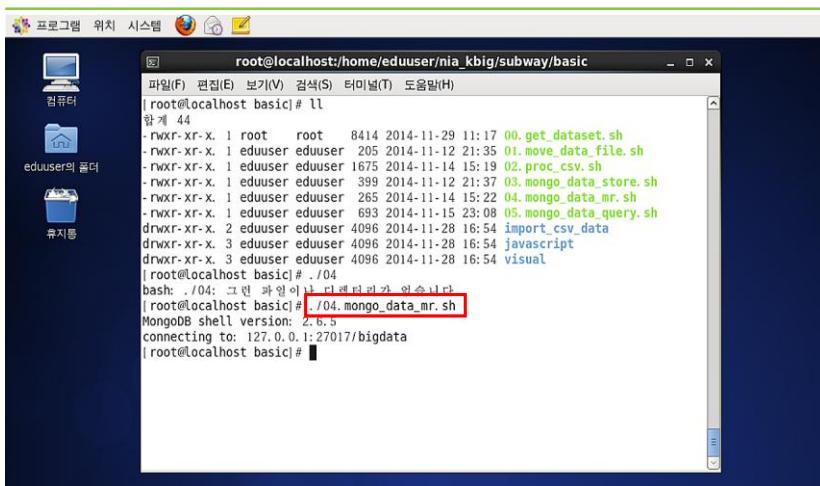


부연설명

- 49페이지 분석 결과 출력 스크립트 소스(javascript/subway_mr_result_query.js)
- 라인 70~74 : 년도별 결과데이터를 CSV형태로 저장하는 라인이다.
- 라인 77~80 : 파라미터의 형태(json, csv)에 따라 해당 파일을 저장하는 함수를 호출하는 라인이다.

> 분석 결과 파일 저장

> 분석 결과 데이터 저장 스크립트 실행



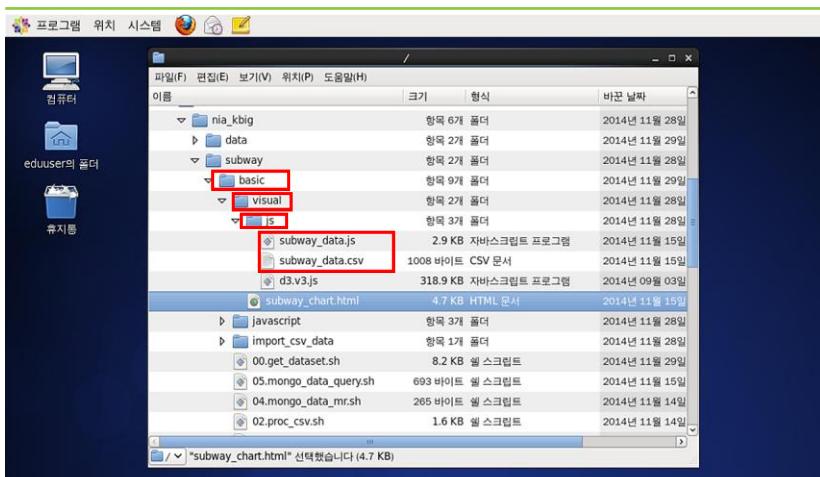
```

프로그램 위치 시스템 웹 브라우저 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[ root@localhost basic]# ll
합계 44
-rwxr-xr-x 1 root root 8414 2014-11-29 11:17 00.get_dataset.sh
-rwxr-xr-x 1 eduuser eduuser 205 2014-11-12 21:35 01.move_data_file.sh
-rwxr-xr-x 1 eduuser eduuser 1675 2014-11-14 15:19 02.proc_csv.sh
-rwxr-xr-x 1 eduuser eduuser 399 2014-11-12 21:37 03.mongo_data_store.sh
-rwxr-xr-x 1 eduuser eduuser 265 2014-11-14 15:22 04.mongo_data_mr.sh
-rwxr-xr-x 1 eduuser eduuser 693 2014-11-15 23:08 05.mongo_data_query.sh
drwxr-xr-x 2 eduuser eduuser 4096 2014-11-28 16:54 import_csv_data
drwxr-xr-x 3 eduuser eduuser 4096 2014-11-28 16:54 javascript
drwxr-xr-x 3 eduuser eduuser 4096 2014-11-28 16:54 visual
[ root@localhost basic]# ./04.mongo_data_mr.sh
MongoDB shell version: 2.6.5
connecting to: 127.0.0.1:27017/bigdata
[ root@localhost basic]#

```

- 몽고 DB에 입력된 분석 결과를 json 파일이나 csv 파일로 저장한다.
./04.mongo_data_mr.sh 입력 후 엔터

> 분석 결과 파일 저장 폴더



- 분석 결과 저장되는 위치는 /home/eduuser/nia_kbig/subway/basic/manual_result/ 폴더 밑으로 subway_data.js, subway_data.csv 파일이 생성된다.

> 시각화 과정

> 시각화 절차



> 시각화 과정

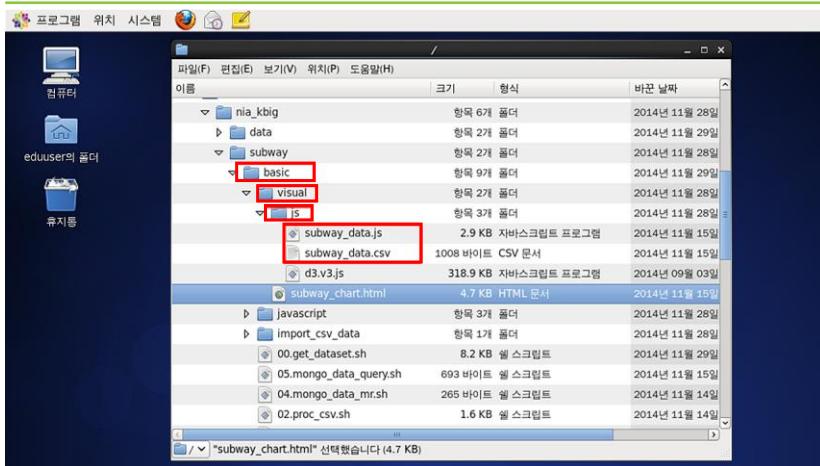
- d3.v3.js 라이브러리 파일을 제공 사이트에서 다운로드하여 저장한다.
- 시각화할 HTML 페이지를 생성한다.
- D3 Chart 라이브러리 모듈에 페이지를 삽입한다.
- D3 Chart Data를 읽어 오는 부분(subway_data.js)에 결과 데이터를 삽입 한다.
- X축과 Y축의 값을 지정한다.
- html 페이지를 웹브라우저에서 실행한다.

> D3 Chart 모듈 삽입과 결과 데이터 삽입

```
<!-- d3 모듈을 불러온다 -->
<script src="js/d3.v3.js"></script>
<-----d3 Chart data 연계 -->
<script src="js/subway_data.js"></script>
```

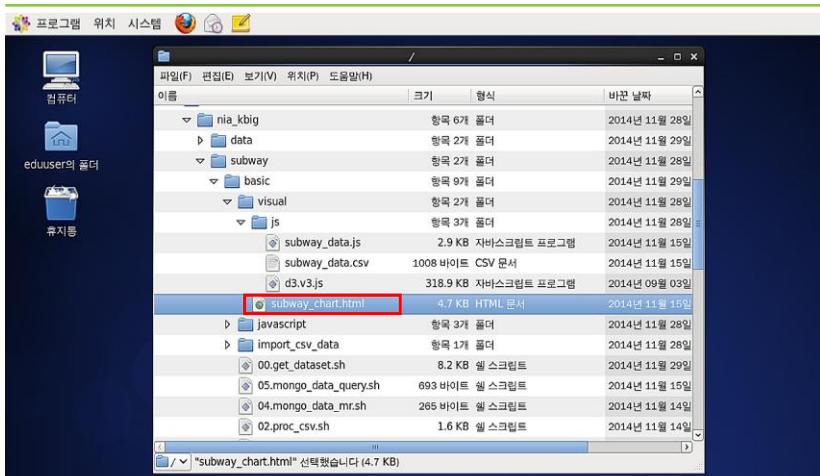
▶ 분석 데이터 시작화

▶ 분석 데이터 파일 복사



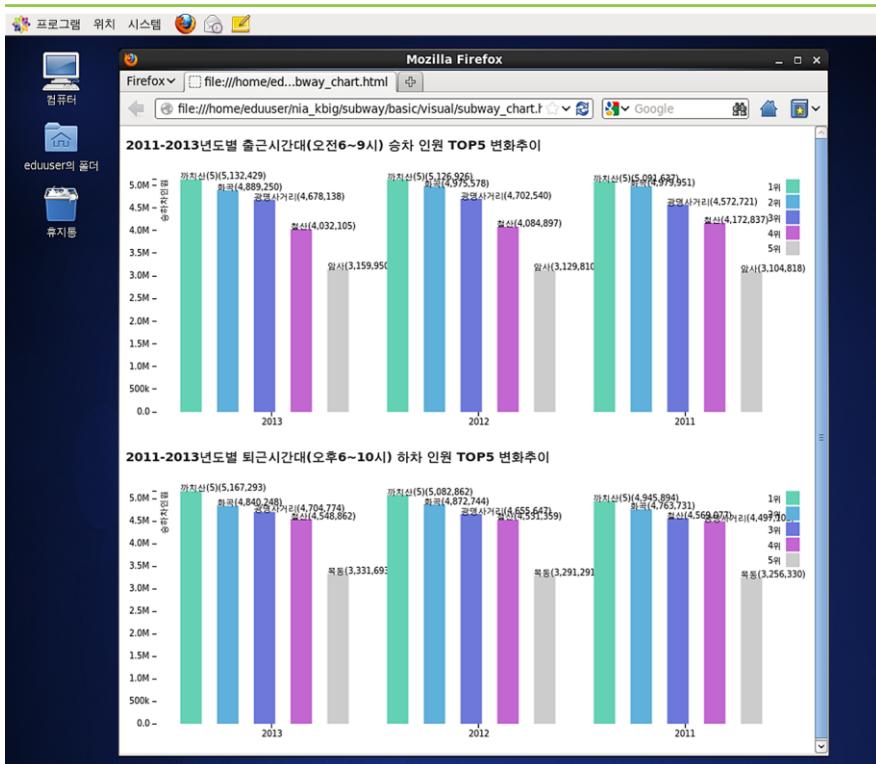
- 분석 결과가 저장된 `/home/eduuser/nia_kbig/subway/basic/manual_result/` 폴더 밑에 `subway_data.js`, `subway_data.csv` 파일을 복사하여 `visual/js/` 폴더 밑으로 붙여넣기를 한다.

▶ 시작화 차트 실행



- 바로 밑에 폴더 있는 `subway_chart.html`을 더블클릭하여 ‘표시’ 버튼을 클릭하여 브라우저로 오픈한다.

▶ 데이터 분석



- 승차인원을 연도별로 보면 상위 5순위는 까치산(5)역, 화곡역, 광명 사거리, 철산, 암사 순으로 분석이 되었다.
- 하차 인원은 승차인원 순위와 비슷하게 분석이 되었으나 암사, 목동이 출퇴근 시간대 순위가 바뀌었다.
- 철산역 경우는 하차 인원이 승차인원에 비해 다른 역에 비해 더 높게 나와있다.
- 철산역 주변에 식당 및 주점이 많이 있어 퇴근 시간대 더 많은 인원이 하차하는 것으로 판단된다.

I. 개요

II. 수집

III. 가공

IV. 저장

V. 분석

VI. 시각화



VII 예제문제

예제 문제1

59

예제 문제2

60

예 / 제 / 문 / 제

예제 1

5~8호선 역 중에서 증가율이 가장 큰 역을 나열하라.

- 2012~2013년도 승하차 인원 데이터를 추출하여 승차, 하차 증감율이 가장 높은 상위 10개 역을 분석하여 막대그래프로 증가율을 시각화하라.

- 2012년, 2013승하차 인원을 원시 데이터 셋에서 추출하여 저장한다.
- 역별로 승하, 하차 인원의 합계를 구한다.
- 2012년도 승차, 하차 인원 2013년도 승차, 하차와의 비율을 구한다.
- 역별 승차, 하차 증감이 가장 많이 증가한 역을 상위 10개를 추출한다.
- 증감율을 시각화하여 나열한다.

예제 2

2013년도 7호선의 연장 개통에 따른 신규 역에 대한 승하차 인원을 비교하라.

- 2013년도 7호선 연장 개통한 역에 대해서 승하차 인원 변화를 월별로 시각화하여 분석하라.
 - 2013년도 개통한 7호선 온수역 ~ 부평구청역까지의 역에 대해서 승하차 인원을 추출한다.
 - 추출된 역별 승하차 인원 데이터를 월별로 합계를 구한다.
 - 월별 합계 인원을 막대그래프로 시각화하여 출력한다.



교통



Intermediate Level

중급과정







I 개요

개요

65

I

개요

> 개요

서울도시철도공사에서 제공 받은 2009~2013년 5~8호선 역별 승하차 정보 데이터를 바탕으로 가을 문화축제 기간(2013.10.24~2013.11.02)의 지하철 5~8호선 승하차 인원과 승하차 인원이 증가한 역을 대상으로 전주 승하차률과 비교하여 증가한 상위 5개 역을 추출하고, 가을 문화축제 기간(2013.10.24~2013.11.02)의 뉴스 데이터 중 가을 문화 축제, 지하철 축제와 관련한 뉴스의 키워드 분석을 하여, 지하철 승하차률과 가을 문화축제와의 패턴 분석을 통하여 승하차률에 영향력을 미치는지에 대한 연관성 분석을 하는 방법을 알아보고자 한다. 이러한 방법으로 분석되어진 데이터를 바탕으로 가을 문화축제뿐만 아니라 지하철 역주변 4계절 축제(행사)에 대한 승하차률 분석을 통해 지하철 역 문화행사(이벤트) 활성화에 기초자료로 활용할 수 있다. 더불어 통신 데이터(CDR)와 역주변 카드사 매출 데이터를 매쉬업 분석하여 상권 분석 자료로 활용할 수 있다.

> 활용 데이터

- **subway.csv** : 지하철 5~8호선 승하차 정보(2009~2014.6)
- **news_2013_9_10.json** : 뉴스 데이터 정보(2013년 9~10월)



용 어 정 리

- CDR(Call Data Record) : 유무선 전화통화에 대한 로그 데이터

> 선행학습

- **하둡 에코시스템** – 하둡 시작, 종료, 하둡 파일 시스템 명령어, 맵리듀스 실행방법
- **자바** – 자바 코딩, 자바 컴파일, JDK 설치, jar 파일 만들기
- **자바스크립트** – 객체(내장객체, 브라우저객체), 속성, 변수, 연산자(연산자 우선순위), 제어문, 함수(내장함수, 함수정의) 사용법
- **D3 차트** – D3 라이브러리 사용법, 차트 설정 방법

> 요구사항

- 지하철 5~8 호선의 가을 문화축제 기간의 승하차율을 행사 전주와 비교 분석하여 축제 기간 중 인원이 증가한 역을 대상으로 행사 내용을 파악해 보고 뉴스 데이터의 키워드를 분석하여 연관성을 파악하라.

> 분석 절차

- 수집된 2009년 ~2013년 5~8호선 승하차 정보 데이터를 로드한다.
- 제공된 지하철(5~8호선) 승하차 정보에서 2013년도 가을 문화축제 기간의 지하철 5~8호선 승하차 인원을 시계열 분석에 용이한 데이터 형태로 추출하여 저장한다.
- 2013년 지하철 5~8호선 승하차 인원 정보 데이터 중 가을 문화축제 기간 (2013.10.25~2013.11.02)의 승하차 인원을 역별로 패턴 분석에 용이한 데이터 형태로 추출하여 저장한다.
- 지하철 5~8호선의 역별로 축제 기간 전주 승하차 인원 추출하여 전주 대비 승하차률을 계산한다.
- 2013년 9월~10월 뉴스 데이터 중 가을 문화축제, 지하철 축제와 관련하여 키워드를 분석하여 키워드 분석 랭킹을 추출한다.
- 승하차 승객이 증가한 역 상위 5개 역을 추출하여 저장한다.
- 분석된 결과 데이터를 엑셀 형식(csv)이나 D3 차트 형식(json)으로 파일로 저장한다
- 저장된 파일을 불러와서 엑셀이나 D3 차트로 역별 승하차률 변화를 시각화해 본다.



II 수집

개요	69
교육용 데이터 샘플	70
데이터 수집	72
데이터 작업 영역 이동 스크립트	75



수집

> 개요

교통 데이터는 서울도시철도공사 제공받은 2009년~2013년 5~8호선 승하차 정보 데이터를 분석에 필요한 정보(연도별 역별 시간대별 승차 인원, 하차인원)를 수집/추출하였고, 뉴스데이터는 개인정보 비식별화 및 특정 내용 비식별화 처리를 통해 분석 목적을 달성할 수 있는 한도 내에서 분석에 용이하게 편집하여 제공한다.

> 수집 방법

- **데이터 제공** : 교통 데이터는 서울도시철도공사 제공해 준 데이터를 OpenAPI, 자료수집기(Crawler)로 데이터를 수집하였고, 실습용 자료는 빅 데이터 분석 활용센터에 접속하여 교통 데이터 셋을 다운로드 할 수 있도록 원시데이터를 제공하고 있다.



용 어 정 리

- **비식별화** : 데이터 값 삭제, 가명처리, 총계처리, 범주화, 데이터 마스킹 등을 통해 개인정보의 일부 또는 전부를 삭제하거나 대체함으로써 다른 정보와 쉽게 결합하여도 특정 개인을 식별할 수 없도록 하는 조치를 말한다.

- *출처: 방송통신위원회, “빅데이터 개인정보보호 가이드라인”, 작성일 2014.12.23

> 교육용 데이터 샘플

> 지하철 승하차 데이터 (subway.csv)

역코드	역명	일자	on_tot	on_05	on_06	on_24	off_11	off_22	off_23	off_24
2511	방화	20130101	4,403	87	145	274	338	338	338	338
2511	방화	20130102	8,467	177	580	383	397	397	397	397
2511	방화	20130103	8,165	160	517	412	375	375	375	375
2511	방화	20130104	8,656	168	526	477	425	425	425	425
2511	방화	20130105	6,236	123	201	425	538	538	538	538
2511	방화	20130106	4,627	94	130	387	365	365	365	365
2511	방화	20130107	8,820	189	571	425	423	423	423	423
2511	방화	20130108	8,702	179	533	480	445	445	445	445
2511	방화	20130109	8,491	183	498	464	403	403	403	403
2511	방화	20130110	8,719	166	537	434	405	405	405	405
2511	방화	20130111	8,930	151	510	458	436	436	436	436
2511	방화	20130112	6,709	140	209	487	554	554	554	554
2511	방화	20130113	4,788	79	134	406	337	337	337	337
2511	방화	20130114	8,730	198	575	458	417	417	417	417

II. 수집

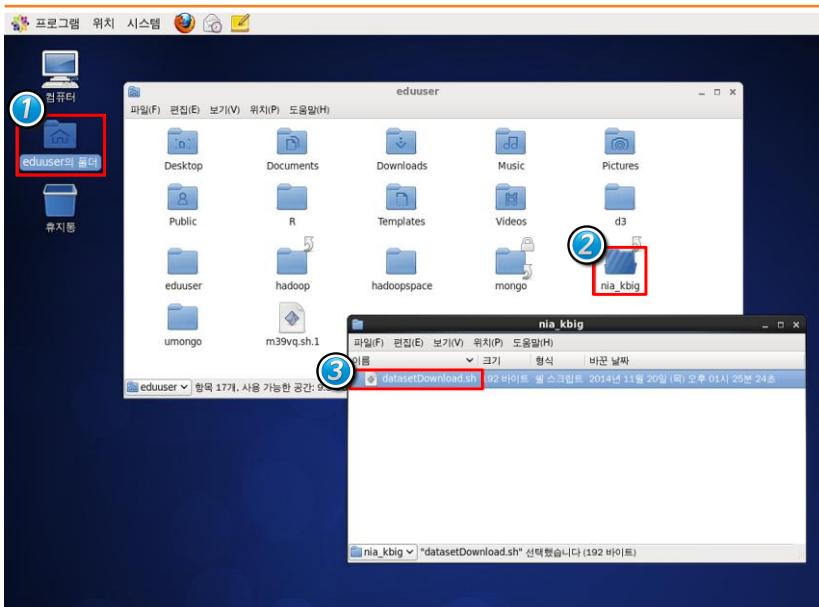
▶ 뉴스 데이터 (news_2013_9_10.json)

일자	제목	기사내용
20131025	지하철 5~8호선에서 가을문화축제 '풍성'	5678호선을 운영하는 서울도시철도공사(사장 김기춘)는 10월 24일(목)부터 11월 2일(토)까지 「2013년도 가을문화축제」를 개최한다. 기울문화축제는 매년 10월 5~8호선 지하철역에서 열리는 시민 참여형 문화행사로 각종 공연과 전시회, 체험 프로그램 등을 지역 주민들이 직접 진행하는 방식으로 이뤄진다. 올해로 18회째를 맞아 지하철역이 시민과 소통하는 열린 공간으로서 시민들에게 즐거움을 선사하는 한편 보다 가깝고 친근한 인상으로 다가가는데 중점을 두었다. 축제기간 동안 5~8호선 지하철역에서는 악기연주와 노래, 춤, 전시회 등 약 320여회의 다채로운 문화행사가 연일 이어진다. 통기타와 색소폰은 물론 오카리나와 플루트, 하모니카 연주 등 다양한 장르의 음악공연은 가을의 정취를 더하고, 그림과 사진, 만화, 화폐, 공예작품에 이르기 까지 특색 있는 전시는 색다른 볼거리를 제공 한다.
20131025	도시철도공사, 5~8호선 역사내 가을문화축제	서울도시철도공사가 5~8호선 역사에서 시민들이 직접 만드는 가을문화축제를 열었다. 25일 서울도시철도 공사는 다음달 2일까지 「2013년도 가을문화축제」를 개최한다고 밝혔다. 기울문화축제는 매년 10월 5~8호선 지하철역에서 열리는 시민 참여형 문화행사다. 각종 공연과 전시회, 체험 프로그램 등을 지역 주민들이 직접 진행하는 방식으로 이뤄진다. 축제기간 동안 악기연주와 노래, 춤, 전시회 등 약 320여회의 다채로운 문화행사가 이어진다. 통기타와 색소폰은 물론 오카리나와 플루트, 하모니카 연주 등 다양한 장르의 음악 공연은 가을의 정취를 더할 예정이다. 그림과 사진, 만화, 화폐, 공예작품에 이르기까지 특색 있는 전시는 색다른 볼거리를 제공한다.
20131026	서울 지하철역에서 즐기는 풍성한 문화행사	서울 지하철 5·6·7·8호선 역사를 찾으면 시민 참여형의 다채로운 문화행사들을 즐길 수 있다. 지하철 5~8호선을 운영하는 서울도시철도공사는 다음달 2일까지 「2013년도 가을문화축제」를 연다고 26일 밝혔다. 축제는 올해로 18회째로, 모두 37개 지하철역에서 악기연주와 노래, 춤, 전시회, 체험교육 등 320여회의 문화행사가 연일 이어진다. 철교놀이를 이용해 역사 벽면을 꾸민 5호선 김포공항역에서는 26~27일 전통 철교놀이 체험전을 한다. 북한산과 가까운 6호선 독바위역에서는 북한산 둘레길 소개와 안전한 산행을 위한 심폐소생술 체험교육이 26일 진행된다.

> 데이터 수집(datasetDownload.sh)

- 데이터 저장소에서 서버 로컬로 교통 데이터 셋을 복사해 온다.
 - subway.csv** : 지하철 승하차 데이터
 - news_2013_9_10.json** : 뉴스 데이터

> 실습코드 디렉토리로 이동

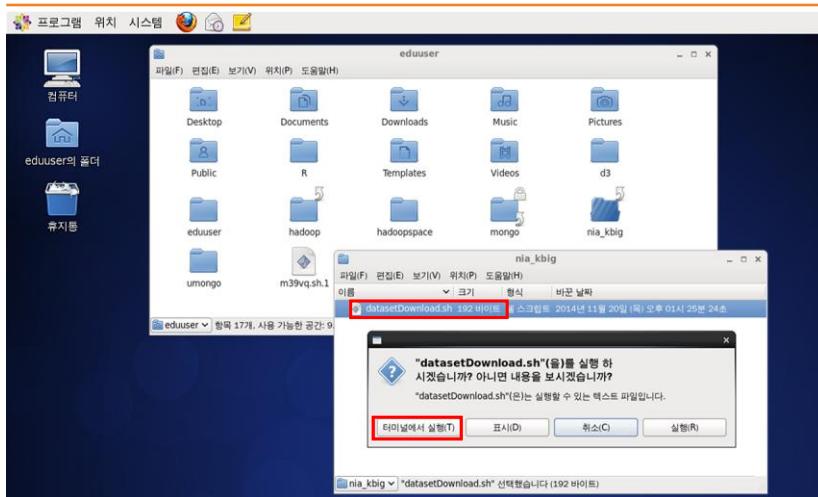


- ① 로그인 후 바탕화면에서 eduuser 폴더를 오픈한다.
- ② nia_kbig 폴더를 오픈한다.
- ③ datasetDownload.sh를 더블클릭하여 실행한다.

II. 수집

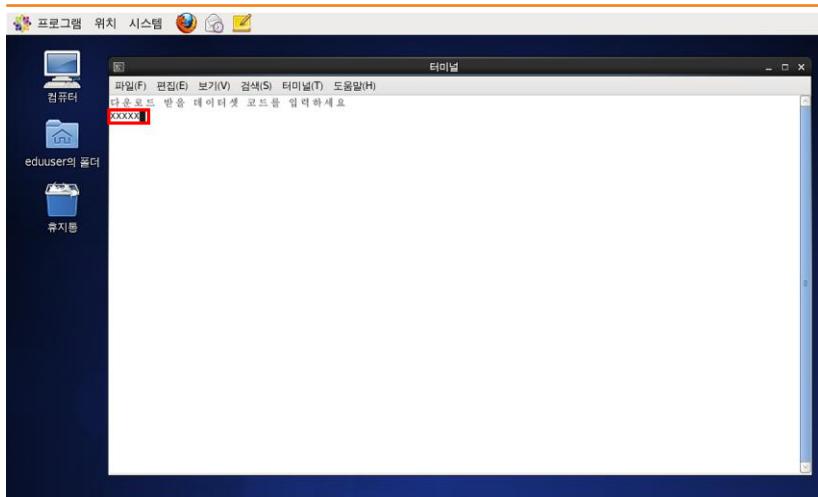
▶ 레파지토리에서 데이터 수집

datasetDownload.sh (원시데이터로 컬서버로 복사)



- '터미널에서 실행' 버튼을 클릭한다.

▶ 데이터셋 코드 입력



- 다운로드 받을 데이터셋 코드를 입력 후 엔터

▶ 데이터셋과 실습용 쉘 스크립트



▶ ① 데이터 및 스크립트

- 01.move_data_file.sh** : 작업영역 Data 폴더로 자료 이동하는 스크립트
- 02.proc_csv.sh** : 원시데이터에서 분석할 대상을 추출하여 저장하는 스크립트
- 03.upload_csv.sh** : 하둡파일시스템 가공데이터 파일 업로드 실행 스크립트
- 04.run.sh** : 가공데이터 분석 맵리듀스 실행 스크립트
- 05.hadoop_filecopy.sh** :
하둡 파일시스템에서 분석데이터를 로컬 서버로 데이터 파일을 복사하는 스크립트
- datasetDownload.sh** :
레파지토리에서 분석데이터와 실습용 스크립트를 다운로드 스크립트
- news_2013_9_10.json** : 2013년도 09월~10월 뉴스데이터
- subway.csv** : 5~8호선 지하철 승하차 데이터
- subway.r** : 지하철 승하차 분석 R 스크립트
- subway_1-4.csv** : 2013년도 1~4호선 지하철 승하차 데이터
- wheather.csv** : 기상데이터

II. 수집

▶ 데이터 작업 영역 이동 스크립트(01.move_data_file.sh)

▶ 데이터 이동 스크립트

- 로컬로 수집해온 데이터를 작업 영역 Data 폴더로 자료를 이동하는 스크립트

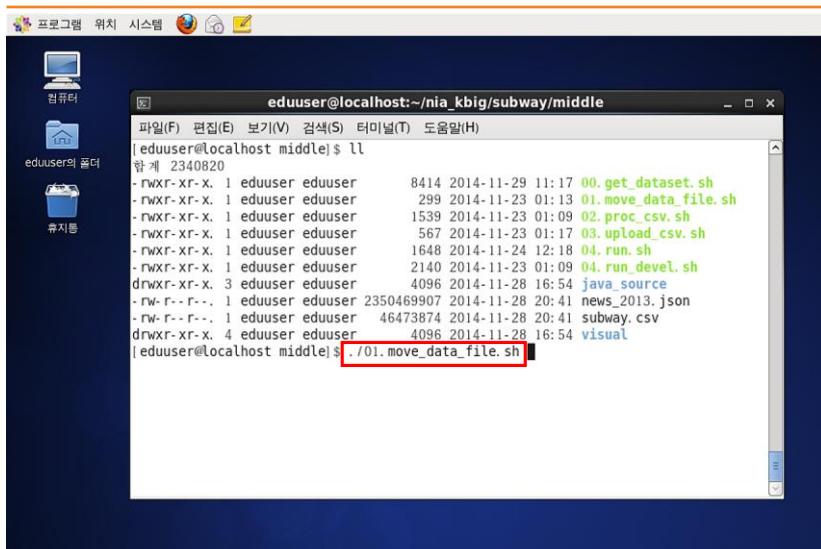
01.move_data_file.sh (작업영역 폴더로 원시데이터 이동)

```
01.#!/bin/bash
02. # 복사 대상 파일 정의
03. # 지하철 승하차 인원
04. TARGET_SUBWAY=/home/eduuser/nia_kbig/subway/middle/subway.csv
05. #뉴스데이터
06. TARGET_NEWS=/home/eduuser/nia_kbig/subway/middle/news_2013_9_10
07. ↛ .json
08. # 작업 디렉토리 정의
09. LOCAL_DIR=/home/eduuser/nia_kbig/data/
10. #작업영역으로 데이터 이동
11. mv $TARGET_SUBWAY $LOCAL_DIR
12. mv $TARGET_NEWS $LOCAL_DIR
```



- 데이터 작업 영역 이동 스크립트 소스(01.move_data_file.sh)
- 라인 04~06 : 다운로드 받은 원시데이터 파일들의 위치(path)를 변수(TARGET_SUBWAY, TARGET_NEWS)로 지정하는 라인이다.
- 라인 08 : 작업영역 디렉토리의 위치(path)를 변수(LOCAL_DIR)로 지정하는 라인이다.
- 라인 10~11 : mv 명령어를 사용하여 다운로드 받은 원시데이터 파일들을 작업영역 디렉토리로 이동시키는 라인이다.

▶ 수집 데이터 셋 작업 영역 폴더 이동



A screenshot of a terminal window titled "eduuser@localhost:~/nia_kbig/subway/middle". The window shows a file listing with the command "ls -l". One line of the output, ". ./01.move_data_file.sh", is highlighted with a red box. The terminal is running on a Linux system with a dark blue desktop background.

```
[eduuser@localhost middle]$ ls -l
합계 2340820
-rwxr-xr-x 1 eduuser eduuser 8414 2014-11-29 11:17 00.get_dataset.sh
-rwxr-xr-x 1 eduuser eduuser 299 2014-11-23 01:13 01.move_data_file.sh
-rwxr-xr-x 1 eduuser eduuser 1539 2014-11-23 01:09 02.proc_csv.sh
-rwxr-xr-x 1 eduuser eduuser 567 2014-11-23 01:17 03.upload_csv.sh
-rwxr-xr-x 1 eduuser eduuser 1648 2014-11-24 12:18 04.run.sh
-rwxr-xr-x 1 eduuser eduuser 2140 2014-11-23 01:09 04.run-devel.sh
drwxr-xr-x 3 eduuser eduuser 4096 2014-11-28 16:54 java_source
-rw-r--r-- 1 eduuser eduuser 2350469907 2014-11-28 20:41 news_2013.json
-rw-r--r-- 1 eduuser eduuser 46473874 2014-11-28 20:41 subway.csv
drwxr-xr-x 4 eduuser eduuser 4096 2014-11-28 16:54 visual
[eduuser@localhost middle]$ . ./01.move_data_file.sh
```

- 로컬에 원시데이터를 작업 영역 폴더로 이동 (/home/eduuser/nia_kbig/data/) 시킨다.
- ./01.move_data_file.sh 입력 후 엔터

I. 개요

II. 수집

III. 가공

IV. 저장

V. 분석

VI. 시각화







가공

▶ 개요

작업 영역 폴더에 복사한 5~8호선 지하철 승하차 데이터에서 2013년 데이터를 추출하여 CSV 파일(2013_subway.csv) 형식으로 저장한다. 데이터 중 on_ 시간으로 시작하는 컬럼은 승차인원이고, off_시간은 하차 인원이다. 데이터 가공은 셀 스크립트를 사용하여 전체 데이터에서 필터링 기법을 적용하여 데이터를 가공한다.

▶ 가공 방법

- 지하철 승하차 데이터 (subway.csv) 파일에서 2013년 지하철 승하차 인원 데이터를 추출하여 2013_subway.csv 파일을 생성한다.

▶ 데이터셋

-2013_subway.csv(2013 지하철 승하차 정보)

역코드	역명	일자	on_tot	on_05	on_06	on_24	off_11	off_22	off_23	off_24
2511	방화	20130101	4,403	87	145	274	338	338	338	338
2511	방화	20130102	8,467	177	580	383	397	397	397	397
2511	방화	20130103	8,165	160	517	412	375	375	375	375
2511	방화	20130104	8,656	168	526	477	425	425	425	425
2511	방화	20130105	6,236	123	201	425	538	538	538	538

> 데이터 가공 스크립트(02.proc_csv.sh)

- 셀 스크립트를 이용하여 2013년도 데이터만을 추출한다.
추출한 데이터는 2013_subway.csv로 저장한다.

02.proc_csv.sh (원시데이터에서 분석할 대상을 추출 하여 저장)

```
01. #!/bin/bash
02. # 입력 CSV 파일 지정
03. INPUT_FILE='/home/eduuser/nia_kbig/data/subway.csv'
04. # 출력결과 CSV 파일 지정
05. OUTPUT_FILE='/home/eduuser/nia_kbig/data/2013_subway.csv'
06. # HEADER컬럼 출력
07. echo "station,stat_name,income_date,on_tot,on_05,on_06,on_07,on_08,on_09,on_10,on_
    ↪ 11,on_12,on_13,on_14,on_15,on_16,on_17,on_18,on_19,on_20,on_21,on_22,on_23,on_
    ↪ _24,off_tot,off_05,off_06,off_07,off_08,off_09,off_10,off_11,off_12,off_13,off_14,off_
    ↪ _15,off_16,off_17,off_18,off_19,off_20,off_21,off_22,off_23,off_24" > $OUTPUT_FILE
08. # ''를 구분자로 해서 파일을 읽어들인다.
09. IFS=':'
10. while read station stat_name income_date on_tot on_05 on_06 on_07 on_08 on_09 on_10
    ↪ on_11 on_12 on_13 on_14 on_15 on_16 on_17 on_18 on_19 on_20 on_21 on_22 on_2
    ↪ 3 on_24 off_tot off_05 off_06 off_07 off_08 off_09 off_10 off_11 off_12 off_13 off_14
    ↪ off_15 off_16 off_17 off_18 off_19 off_20 off_21 off_22 off_23 off_24
11. do
12.     # DATA가 2013 시작하는 년도인지 체크한다.
13.     is_valid=0
14.     if [[ $income_date == 2013* ]]; then
15.         is_valid=1
16.     fi
17.     if [[ $is_valid == 1 ]]; then
18.         # 해당년도의 데이터만을 CSV로 출력한다.
19.         echo "$station,$stat_name,$income_date,$on_tot,$on_05,$on_06,$on_07,$on_
    ↪ _08,$on_09,$on_10,$on_11,$on_12,$on_13,$on_14,$on_15,$on_16,$on_1
    ↪ 7,$on_18,$on_19,$on_20,$on_21,$on_22,$on_23,$on_24,$off_tot,$off_05,
    ↪ $off_06,$off_07,$off_08,$off_09,$off_10,$off_11,$off_12,$off_13,$off_14,
    ↪ $off_15,$off_16,$off_17,$off_18,$off_19,$off_20,$off_21,$off_22,$off_23,
    ↪ $off_24" >> $OUTPUT_FILE
20.     fi
21. done < $INPUT_FILE
```



- 데이터 가공 스크립트 소스(02.proc_csv.sh)
- 라인 03 : 가공할 데이터 파일(subway.csv)을 설정하는 라인이다.
- 라인 05 : 가공 후 저장할 파일(2013_subway)을 설정하는 라인이다.
- 라인 06~07 : 가공 데이터 결과물 헤더를 설정하는 라인이다.
- 라인 09~10 : 원시데이터에서 데이터를 1라인씩 루프 돌면서 읽어 들이는 라인이다.
- 라인 13~21 : 읽어 드린 데이터에서 2013년도 데이터만을 선정하여 파일에 저장하는 라인이다.

I. 개요

II. 수집

III. 가공

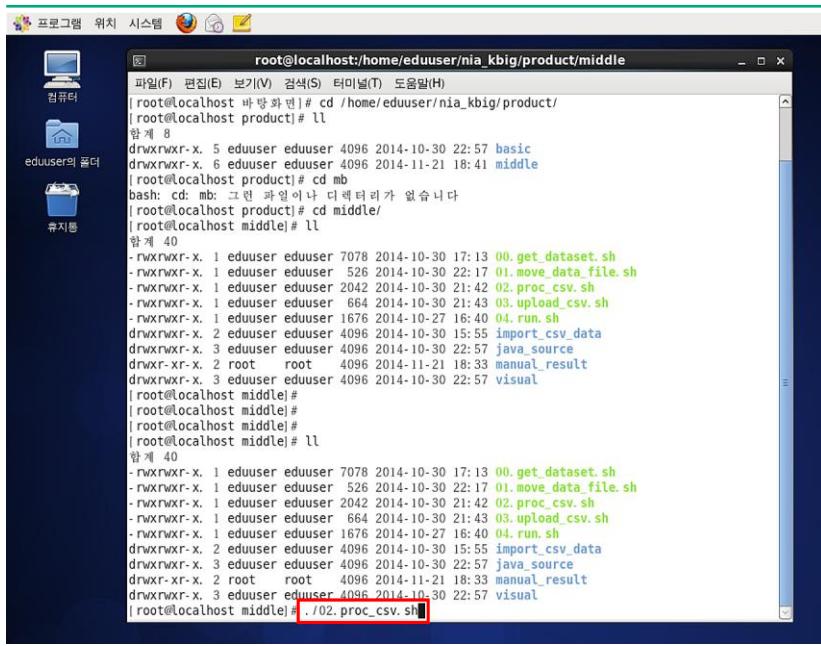
IV. 저장

V. 분석

VI. 시각화

III. 가공

▶ 원시데이터에서 분석 대상 데이터 가공

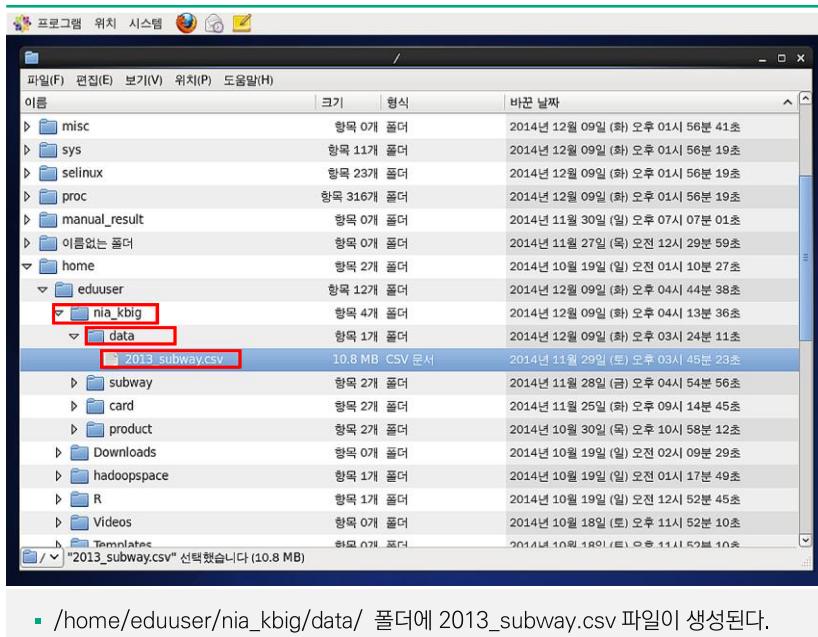


The screenshot shows a terminal window titled 'root@localhost:/home/eduuser/nia_kbig/product/middle'. The window displays a file listing with the command 'll'. A red box highlights the command './02.proc_csv.sh' which was just entered.

```
root@localhost:/home/eduuser/nia_kbig/product/middle# ll
total 12
drwxrwxr-x. 5 eduuser eduuser 4096 2014-10-30 22:57 basic
drwxrwxr-x. 6 eduuser eduuser 4096 2014-11-21 18:41 middle
drwxrwxr-x. 1 eduuser eduuser 2042 2014-10-30 21:42 02.proc_csv.sh
drwxrwxr-x. 1 eduuser eduuser 664 2014-10-30 21:43 03.upload_csv.sh
drwxrwxr-x. 1 eduuser eduuser 1676 2014-10-27 16:40 04.run.sh
drwxrwxr-x. 2 eduuser eduuser 4096 2014-10-30 15:55 import_csv_data
drwxrwxr-x. 3 eduuser eduuser 4096 2014-10-30 22:57 java_source
drwxr-xr-x. 2 root root 4096 2014-11-21 18:33 manual_result
drwxrwxr-x. 3 eduuser eduuser 4096 2014-10-30 22:57 visual
[root@localhost middle]#
[root@localhost middle]#
[root@localhost middle]#
[root@localhost middle]# ll
total 12
drwxrwxr-x. 5 eduuser eduuser 4096 2014-10-30 17:13 00.get_dataset.sh
drwxrwxr-x. 1 eduuser eduuser 526 2014-10-30 22:17 01.move_data_file.sh
drwxrwxr-x. 1 eduuser eduuser 2042 2014-10-30 21:42 02.proc_csv.sh
drwxrwxr-x. 1 eduuser eduuser 664 2014-10-30 21:43 03.upload_csv.sh
drwxrwxr-x. 1 eduuser eduuser 1676 2014-10-27 16:40 04.run.sh
drwxrwxr-x. 2 eduuser eduuser 4096 2014-10-30 15:55 import_csv_data
drwxrwxr-x. 3 eduuser eduuser 4096 2014-10-30 22:57 java_source
drwxr-xr-x. 2 root root 4096 2014-11-21 18:33 manual_result
drwxrwxr-x. 3 eduuser eduuser 4096 2014-10-30 22:57 visual
[root@localhost middle]# ./02.proc_csv.sh
```

▪ 원시 데이터 셋에서 분석할 데이터를 가공하여 2013_subway.csv 파일을 생성한다.
./02.proc_csv.sh 입력 후 엔터

▶ 가공데이터 생성 파일 디렉토리



- /home/eduuser/nia_kbig/data/ 폴더에 2013_subway.csv 파일이 생성된다.



IV 저 장

개요	85
가공 데이터 하둡 파일시스템 업로드	86
가공 데이터 하둡 파일시스템 저장	87
하둡 파일시스템 파일 조회	88
하둡 명령어로 파일 조회	89

> 개요

하둡 파일 시스템의 하둡 맵리듀스를 이용하기 위해 분석에 필요한 자료들을 하둡 파일 시스템에 업로드 시킨다. 분석에 사용될 데이터는 2013년도 역별 승하차 정보와 2013년도 9월~10월 뉴스 데이터를 사용한다. 하둡의 put 명령어를 사용하여 하둡 파일시스템의 /user/bigdata/ 폴더로 자료를 업로드한다. 하둡 파일 시스템에 저장 시에는 한글이 있는 경우 깨지기 때문에 반드시 utf-8 포맷으로 형식으로 업로드를 해야 맵리듀스 분석 실행 시 오류가 발생하지 않는다.

> 저장 방법

- 2013년치 지하철 승하차 데이터 파일(2013_subway.csv)과 2013년도 뉴스 데이터 파일(news_2013_9_10.json) 파일을 하둡에 업로드한다.
- 하둡 커맨드를 이용해서 가공된 데이터를 하둡 파일 시스템에 업로드한다.

> 가공데이터 하둡 파일시스템 업로드(03.upload_csv.sh)

> 하둡 파일시스템에 업로드 스크립트

- 원시데이터에서 가공된 원시데이터에서 가공된 2013_subway.csv, news_2013_9_10.json 파일을 하둡시스템에 업로드한다.

03.upload_csv.sh (가공데이터를 하둡 파일시스템으로 업로드)

```

01.#!/bin/bash
02. # 지하철 승하차 정보 CSV 파일 지정
03. SUBWAY_FILE='/home/eduuser/nia_kbig/data/2013_subway.csv'
04. # 2013년도 뉴스기사 파일 지정
05. NEWS_FILE='/home/eduuser/nia_kbig/data/news_2013_9_10.json'
06. # 하둡의 지하철 승하차 정보 저장 위치
07. HDFS_SUBWAY=/user/bigdata/2013_subway.csv
08. # 하둡의 뉴스기사 저장 위치
09. HDFS_NEWS=/user/bigdata/news_2013_9_10.json
10. # make directory on hadoop
11. hadoop fs -mkdir -p /user/bigdata
12. # upload target file to HDFS
13. hadoop fs -put $SUBWAY_FILE $HDFS_SUBWAY
14. hadoop fs -put $NEWS_FILE $HDFS_NEWS
15.

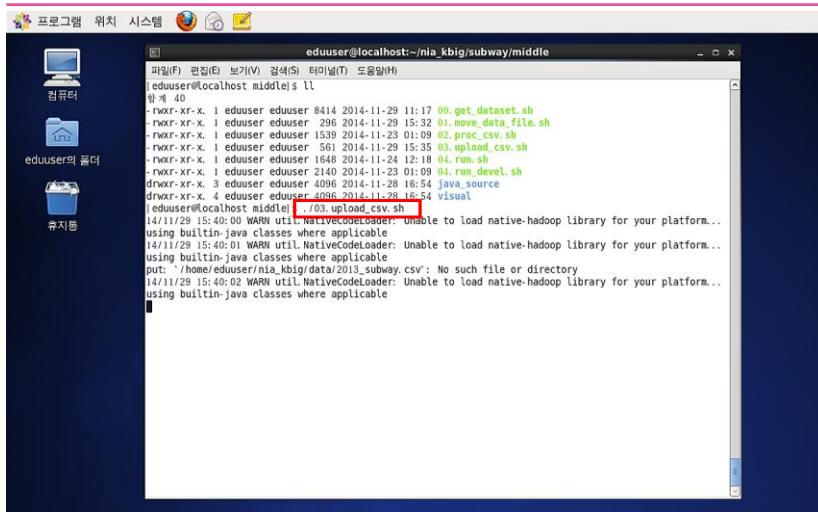
```



- 가공 데이터 하둡 파일시스템 업로드 스크립트 소스(03.upload_csv.sh)
- 라인 03 : 가공할 데이터 파일(2013_subway.csv)을 설정하는 라인이다.
- 라인 05 : 2013년도 뉴스기사 파일(news_2013_9_10.json)을 설정하는 라인이다.
- 라인 07 : 하둡 파일시스템에 가공 승하차데이터 저장 위치를 지정하는 라인이다.
- 라인 09 : 하둡 파일시스템에 뉴스기사 데이터 저장 위치를 지정하는 라인이다.
- 라인 11 : 하둡 파일시스템에 /user/bigdata 폴더를 생성하는 라인이다.
- 라인 13 : 하둡 파일시스템에 가공 승하차 데이터 파일을 업로드하는 라인이다.
- 라인 14 : 하둡 파일시스템에 뉴스 데이터 파일을 업로드하는 라인이다.

> 가공데이터 하둡 파일시스템 저장

> 가공데이터 하둡 파일시스템 저장



The screenshot shows a terminal window titled 'eduuser@localhost:~/nia_kbig/subway/middle'. The window displays a command-line session with several HDFS file operations:

```
eduuser@localhost:~/nia_kbig/subway/middle]$ ll
계 40
-rwxr-Xr-X 1 eduuser eduuser 8414 2014-11-29 11:17 00.get_dataset.sh
-rwxr-Xr-X 1 eduuser eduuser 298 2014-11-29 15:39 01.download_data_file.sh
-rwxr-Xr-X 1 eduuser eduuser 1539 2014-11-23 01:09 02.proc_csv.sh
-rwxr-Xr-X 1 eduuser eduuser 561 2014-11-29 15:35 03.upload_csv.sh
-rwxr-Xr-X 1 eduuser eduuser 1648 2014-11-24 12:18 04.rmn.sh
-rwxr-Xr-X 1 eduuser eduuser 2140 2014-11-23 01:09 04.rmn-devel.sh
drwxr-Xr-X 3 eduuser eduuser 4096 2014-11-28 16:54 java_source
drwxr-Xr-X 4 eduuser eduuser 4096 2014-11-28 16:54 visual

[eduuser@localhost middle] ./03.upload_csv.sh
```

The command `./03.upload_csv.sh` is highlighted with a red box.

- 가공한 2013_subway.csv, news_2013_9_10.json 파일을 하둡 파일시스템에 업로드 한다. `./03.upload_csv.sh` 입력 후 엔터

> 하둡 파일시스템 파일 조회

> 하둡 파일시스템 접속

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
2013_subway.csv	file	10.76 MB	1	128 MB	2014-12-08 11:57	rw-r--r--	eduuser	supergroup
news_2013.json	file	2.19 GB	1	128 MB	2014-12-08 11:58	rw-r--r--	eduuser	supergroup

Go back to DFS home

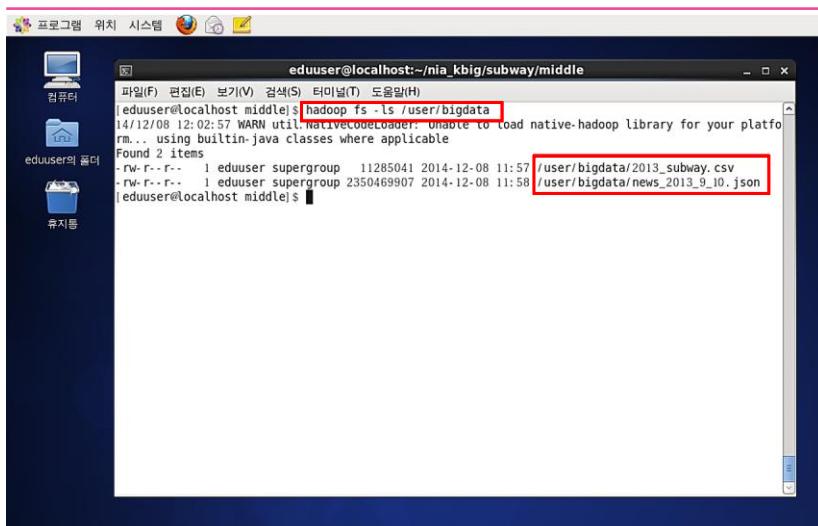
Local logs

[Log directory](#)

- 파일어포스 브라우저를 클릭하여 오픈 한 후 주소창에 <http://localhost:50070> 입력 후 엔터 치면 하둡 파일 시스템에 접속할 수 있다.
- Browse the filesystem 링크를 클릭하고 user/bigdata/ 폴더를 클릭하면 업로드한 가공 데이터 목록을 볼 수 있다.

> 하둡 명령어로 파일 조회

> 하둡 파일시스템 조회



The screenshot shows a Windows desktop environment with a terminal window open. The terminal window title is "eduuser@localhost:~/nia_kbig/subway/middle". The command entered is "hadoop fs -ls /user/bigdata". The output shows two files: "2013_subway.csv" and "news_2013_9_10.json". A red box highlights the command and the file names.

```
eduuser@localhost middle] $ hadoop fs -ls /user/bigdata
14/12/08 12:02:57 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r-- 1 eduuser supergroup 11285041 2014-12-08 11:57 /user/bigdata/2013_subway.csv
-rw-r--r-- 1 eduuser supergroup 2350469907 2014-12-08 11:58 /user/bigdata/news_2013_9_10.json
[eduuser@localhost middle]$
```

- 터미널 창에서 **hadoop fs -ls /user/bigdata** 입력 후 엔터를 치면 하둡 파일시스템에 올라간 파일을 조회할 수 있다.
- 업로드한 2013_subway.csv , news_2013_9_10.csv 파일 목록을 확인할 수 있다.

I. 개요

II. 수집

III. 가공

IV. 저장

V. 분석

VI. 시각화

W





V 분석

개요	93
데이터 분석 스크립트	96
데이터 분석 맵리द스 실행	98
분석 데이터 파일 조회	99
분석 후 결과 데이터	100

V

분석

> 개요

교통 데이터의 분석은 자바 스크립트와 하둡 파일 시스템의 맵리듀스 기능을 활용하여 D3 챕트와 R Studio에서 시계열 분석을 할 수 있는 형태로 분석 결과 데이터를 만든다. 저장 단계에서 하둡 파일 시스템에 업로드한 2013년 역별 승하차 정보 데이터에서 가을 문화축제 기간 (2013.10.24~2013.11.02)의 승하차 정보를 추출하고, 같은 기간의 뉴스 데이터 중 가을 문화 축제, 지하철 축제와 관련한 뉴스의 키워드 추출하여 시계열 분석에 필요한 데이터 파일로 저장한다. 이러한 방법으로 지하철 승하차률과 가을 문화축제와의 패턴분석을 통하여 승하차률에 영향력을 미치는지 연관성 분석을 통해 검증해 본다.

> 분석 방법

- 자바로 하둡 파일 시스템에 올라가 있는 2013_subway.csv을 대상으로 맵리듀스 작업을 통해서 축제 기간 역별 승하차 총인원 합계 데이터를 뽑아낸다.
- 맵리듀스 결과 파일을 로드하여 축제 기간 중 총 하차 인원이 가장 많이 증가한 역 순으로 리스트를 만든다.
- 맵리듀스 작업을 통해서 뉴스 데이터에서 1위 역/가을 문화축제에 대한 키워드를 추출하여 상위 10을 찾아본다.

➤ 검색 키워드 사전

- 검색 키워드 사전 정보를 가을문화 축제와 관련 있는 키워드를 구축하여 뉴스 데이터에서 키워드를 추출하여 집계하여 랭킹정보를 추출한다.

검색 키워드 사전			
체험	놀이	가을	문화
축제	대중	음악	콘서트
음악회	전시회	강의	웃음
연주회	연주	작품	미술
힐링	만들기	어쿠스틱	지하철

➤ 가공 데이터 샘플

- 2013년 지하철 승하차 정보

역코드	역명	일자	on_tot	on_05	on_06	on_24	off_11	off_22	off_23	off_24
2511	방화	20130101	4,403	87	145	274	338	338	338	338
2511	방화	20130102	8,467	177	580	383	397	397	397	397
2511	방화	20130103	8,165	160	517	412	375	375	375	375
2511	방화	20130104	8,656	168	526	477	425	425	425	425
2511	방화	20130105	6,236	123	201	425	538	538	538	538

V. 분석

-2013년 뉴스데이터

일자	제목	기사내용
20131025	지하철 5~8호선에서 가을문화축제 '풍성'	5678호선을 운영하는 서울도시철도공사(사장 김기춘)는 10월 24일(목)부터 11월 2일(토)까지 「2013년도 가을문화축제」를 개최한다. 가을문화축제는 매년 10월 5~8호선 지하철역에서 열리는 시민 참여형 문화행사로 각종 공연과 전시회, 체험 프로그램 등을 지역 주민들이 직접 진행하는 방식으로 이뤄진다. 올해로 18회째를 맞아 지하철역이 시민과 소통하는 열린 공간으로서 시민들에게 즐거움을 선사하는 한편 보다 가깝고 친근한 인상으로 다가가는데 중점을 두었다. 축제기간 동안 5~8호선 지하철역에서는 악기연주와 노래, 춤, 전시회 등 약 320여회의 다채로운 문화행사가 연일 이어진다. 통기타와 색소폰은 물론 오카리나와 플루트, 하모니카 연주 등 다양한 장르의 음악공연은 가을의 정취를 더하고, 그림과 사진, 만화, 화폐, 공예작품에 이르기 까지 특색 있는 전시는 색다른 볼거리를 제공 한다.
20131025	도시철도공사, 5~8호선 역사내 가을문화축제	서울도시철도공사가 5~8호선 역사에서 시민들이 직접 만드는 가을문화축제를 열었다. 25일 서울도시철도공사는 다음달 2일까지 「2013년도 가을문화축제」를 개최한다고 밝혔다. 가을문화축제는 매년 10월 5~8호선 지하철역에서 열리는 시민 참여형 문화행사다. 각종 공연과 전시회, 체험 프로그램 등을 지역 주민들이 직접 진행하는 방식으로 이뤄진다. 축제기간 동안 악기연주와 노래, 춤, 전시회 등 약 320여회의 다채로운 문화행사가 이어진다. 통기타와 색소폰은 물론 오카리나와 플루트, 하모니카 연주 등 다양한 장르의 음악공연은 가을의 정취를 더할 예정이다. 그림과 사진, 만화, 화폐, 공예작품에 이르기까지 특색 있는 전시는 색다른 볼거리를 제공한다.
20131026	서울 지하철역에서 즐기는 풍성한 문화행사	서울 지하철 5·6·7·8호선 역사를 찾으면 시민 참여형의 다채로운 문화행사들을 즐길 수 있다. 지하철 5~8호선을 운영하는 서울도시철도공사는 다음달 2일까지 「2013년도 가을문화축제」를 연다고 26일 밝혔다. 축제는 올해로 18회째로, 모두 37개 지하철역에서 악기연주와 노래, 춤, 전시회, 체험교육 등 320여회의 문화행사가 연일 이어진다. 철교놀이를 이용해 역사 벽면을 꾸민 5호선 김포공항역에서는 26~27일 전통 철교놀이 체험전을 한다. 북한산과 가까운 6호선 독바위역에서는 북한산 둘레길 소개와 안전한 산행을 위한 심폐소생술 체험교육이 26일 진행된다.

> 데이터 분석 스크립트(04.run.sh)

> 가공 데이터 분석 실행 스크립트

- 맵리듀스를 처리하는 프로그램은 subway.java에 구현되어 있다.
- 자바 프로그램을 컴파일하여 subway.jar 파일로 만든 후 yarn 커맨드를 이용해서 subway.jar 파일로 맵리듀스작업을 수행한다.
- 분석 결과는 하둡 파일시스템의 지정한 디렉토리에 저장한다.

04.run.sh (맵리듀스 실행)

```
01.#!/bin/bash  
02. # 현재 위치를 지정한다.  
03. CURRENT_DIR=/home/eduuser/nia_kbig/subway/middle  
04. # 컴파일하여 생성할 프로그램(jar) 경로를 지정한다.  
05. TARGET_JAR=$CURRENT_DIR/subway.jar  
06. # 소스파일 디렉토리를 지정한다.  
07. TRAGET_SOURCE_DIR=$CURRENT_DIR/java_source  
08. # 컴파일할 소스를 지정한다.  
09. TARGET_SOURCE=com/nia/hadoop/*.java  
10. # jar를 생성하는데 필요한 class 파일을 지정한다.  
11. TARGET_CLASSES=com/nia/hadoop/*.class  
12. # 실행시킬 클래스 명 지정  
13. EXE_CLASS=com.nia.hadoop.subway  
14. # Hadoop상에 존재하는 지하철 승하차 파일을 지정한다.  
15. INPUT_SUBWAY_DATA=/user/bigdata/2013_subway.csv  
16. # Hadoop상에 존재하는 2013년 뉴스 파일을 지정한다.  
17. INPUT_NEWS_DATA=/user/bigdata/news_2013_9_10.json  
18. # MapReduce로 처리한 결과 데이터파일을 생성할 디렉토리를 지정한다.  
19. OUTPUT_DIR=/user/bigdata/subway/out/  
20. # 소스 디렉토리로 이동한다.  
21. cd $TRAGET_SOURCE_DIR  
22. #컴파일에 필요한 hadoop 라이브러리 패스와 함께 source를 컴파일한다.
```



- 데이터 분석 스크립트 소스(04.run.sh)
- 라인 03 : 현재 소스가 있는 위치 디렉토리를 지정하는 라인이다.
- 라인 05 : 컴파일하여 실행 프로그램을 jar를 지정하는 라인이다.
- 라인 07 : 자바소스가 있는 위치를 지정하는 라인이다.
- 라인 09~11 : 컴파일 소스가 있는 위치 및 컴파일을 설정하는 라인이다.
- 라인 13~17 : 하둡 파일시스템에 올려져 있는 가공데이터와 뉴스데이터를 지정하는 라인이다.
- 라인 19 : 분석 실행 결과 데이터를 /user/bigdata/subway/out으로 지정하는 라인이다.

I. 개요

II. 수집

III. 가공

IV. 저장

V. 분석

VI. 시각화

V. 분석

```
javac -classpath /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-core-2.2.0.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar:/usr/local/hadoop/share/hadoop/common/hadoop-common-2.2.0.jar $TARGET_SOURCE  
# 컴파일한 *.class 파일을 jar로 압축한다.  
jar cf $TARGET_JAR $TARGET_CLASSES  
# yarn 커맨드로 Hadoop에서 TARGET_JAR 프로그램을 돌려서 Map/Reduce를 실행한다.  
yarn jar $TARGET_JAR $EXE_CLASS $INPUT_SUBWAY_DATA $INPUT_NEWS_DATA  
$OUTPUT_DIR  
# 작업 수행이 완료되었다면 소스 디렉토리에서 나온다.  
cd ..
```



부연설명

- 데이터 분석 스크립트 소스(04.run.sh)
- 라인 23 : javac로 자바 컴파일을 실행을 하는 라인이다.
- 라인 25 : 컴파일한 자바 class 파일을 jar로 압축하는 라인이다.
- 라인 27 : yarn 명령어를 이용하여 하둡의 맵리듀스를 실행하는 라인이다.
`yarn jar jar압록파일명 가공승하차데이터 뉴스데이터 분석결과저장위치`

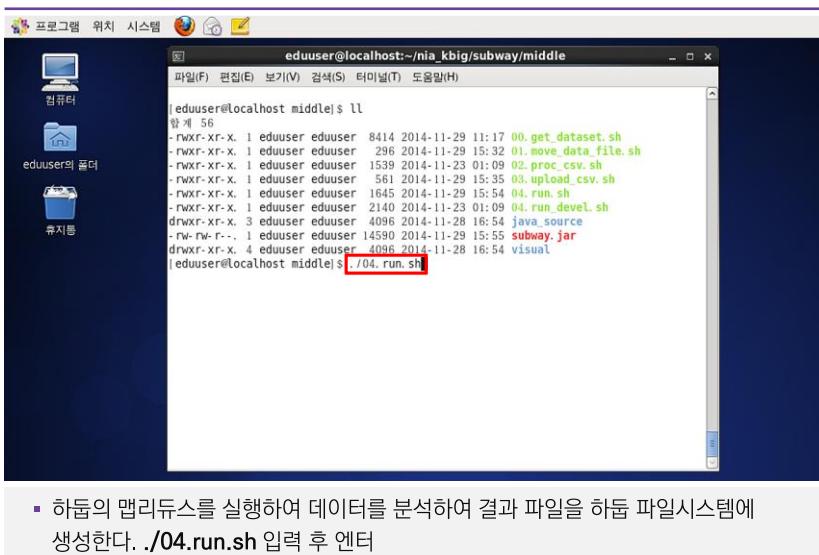
Tip ↪

데이터 분석 스크립트(04.run.sh) 실행시, 맵리듀스 분석 실행 중 멈춤 현상 해결 방법

- Ctrl+C 를 눌러 스크립트 실행 종료.
- 하둡 종료 : 터미널 입력창에 stop-all.sh 입력 후 엔터.
- 하둡 재실행 : 터미널 입력창에 start-all.sh 입력 후 엔터.
- 하둡 실행 상태 확인 : 터미널 입력창에 jps 입력 후 엔터.
(목록 중에 NodeManager가 존재하는지 확인한다.)
- 데이터 분석 스크립트(04.run.sh) 재실행 : 터미널 입력창에 ./04.run.sh 입력 후 엔터.

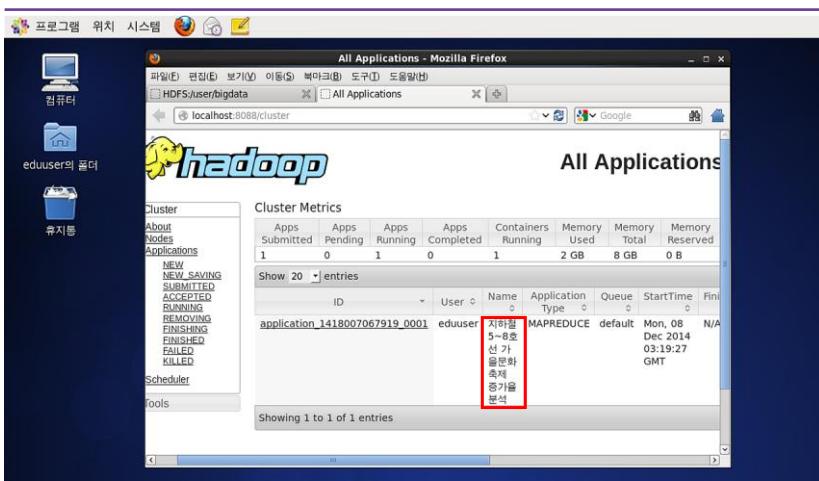
> 데이터 분석 맵리듀스 실행

> 분석 맵리듀스 실행



- 하둡의 맵리듀스를 실행하여 데이터를 분석하여 결과 파일을 하둡 파일시스템에 생성한다. ./04.run.sh 입력 후 엔터

> 맵리듀스 실행 현황 조회



- 파이어폭스 브라우저를 클릭한 후 주소 입력창에 <http://localhost:8088>을 입력 후 엔터를 치면 맵리듀스 진행과정을 볼 수 있다.

> 분석 데이터 파일 조회

> 맵리듀스 분석 결과 파일 조회

Contents of directory /user/bigdata/subway/out/result1

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
_SUCCESS	file	0 B	1	128 MB	2014-11-29 15:56	rw-r--r--	eduuser	supergroup
part-r-00000	file	186 B	1	128 MB	2014-11-29 15:56	rw-r--r--	eduuser	supergroup
station_growth.csv	file	215 B	1	128 MB	2014-11-29 15:56	rw-r--r--	eduuser	supergroup
station_growth.js	file	768 B	1	128 MB	2014-11-29 15:56	rw-r--r--	eduuser	supergroup

- 지하철 승하차 인원 분석 결과 저장 위치는 하둡 파일시스템 안에 /user/bigdata/subway/out/result1에 station_growth.js, station_growth.csv의 2개의 파일로 데이터가 저장된다.

Contents of directory /user/bigdata/subway/out/result2

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
_SUCCESS	file	0 B	1	128 MB	2015-01-19 12:32	rw-r--r--	eduuser	supergroup
keyword_rank.csv	file	158 B	1	128 MB	2015-01-19 12:32	rw-r--r--	eduuser	supergroup
keyword_rank.js	file	652 B	1	128 MB	2015-01-19 12:32	rw-r--r--	eduuser	supergroup
part-r-00000	file	212 B	1	128 MB	2015-01-19 12:32	rw-r--r--	eduuser	supergroup

- 뉴스 키워드 분석 결과 저장 위치는 하둡 파일시스템 안에 /user/bigdata/subway/out/result2에 keyword_rank.js, keyword_rank.csv의 2개의 파일로 데이터가 저장된다.

> 분석 후 결과 데이터

> station_growth

```
01. var data =  
02. [  
03. "ranking": "1",  
04. "growth": "15.97",  
05. "festival_count": "233399",  
06. "week_ago_count": "201258" },  
07. {"stat_name": "독바위", "data": {  
08. "ranking": "2", "festival_count": "7377", "growth": "10.916",  
09. "week_ago_count": "6651" },  
10. {"stat_name": "영등포구청(5)", "data": {  
11. "ranking": "3", "growth": "8.232", "festival_count": "45594",  
12. "week_ago_count": "42126" },  
13. {"stat_name": "신금호", "data": {  
14. "ranking": "4", "growth": "3.085", "festival_count": "13333",  
15. "week_ago_count": "12934" },  
16. {"stat_name": "망원", "data": {  
17. "ranking": "5", "growth": "1.936", "festival_count": "75872",  
18. "week_ago_count": "74431" } }  
19.  
20. ];
```

I. 개요

II. 수집

III. 가공

IV. 저장

> keyword_rank

```
01. var keyword_data =  
02. [  
03. { "keyword": "축제", "ranking": 1, "count": 142 },  
04. { "keyword": "문화", "ranking": 2, "count": 107 },  
05. { "keyword": "가을", "ranking": 3, "count": 70 },  
06. { "keyword": "체험", "ranking": 4, "count": 27 },  
07. { "keyword": "개최", "ranking": 5, "count": 17 },  
08. { "keyword": "놀이", "ranking": 6, "count": 16 },  
09. { "keyword": "만들기", "ranking": 7, "count": 15 },  
10. { "keyword": "음악", "ranking": 8, "count": 12 },  
11. { "keyword": "작품", "ranking": 9, "count": 10 },  
12. { "keyword": "연주", "ranking": 10, "count": 8 }  
13.  
14. ];
```

V. 분석

VI. 시각화



1

2



VI 시각화

개요	103
분석 데이터 저장 방법	104
시각화 과정	107
분석 데이터 시각화	108
데이터 분석	111

VI

시각화

> 개요

교통 데이터(지하철 승하차)와 뉴스 데이터를 하둡 맵리듀스로 분석하는 과정에서 데이터를 시각화하기 위해 CSV, JSON 형태의 결과 데이터로 저장해야 한다. 역 분석 결과 데이터와 뉴스 데이터의 가을 문화축제와 관련된 키워드 랭킹 데이터를 시각화하기 위하여 D3 차트 라이브러리를 활용한다. 이 결과 지하철 승하차 증감률 데이터는 막대그래프 형태, 키워드 분석 결과는 파이 차트 형태로 시각화하여 키워드 랭킹에 따른 승하차 변화에 대한 패턴을 비교 분석할 수 있다.

> 시각화 방법 및 활용기술

- 축제 기간 중 총 승하차 인원이 가장 많이 증가한 역 순으로 리스트와 증가율 1위 역과 가을 문화축제에 대한 뉴스 데이터 키워드 상위 10을 시각화에 사용하기 위해서 준비한다.
- 하둡의 맵리듀스 결과 출력된 결과 파일을 다운로드한다.
- 승하차 증가율 분석 결과 파일은 /user/bigdata/subway/out/result1/station_growth.js, station_growth.csv이고, 뉴스 키워드 분석 결과 파일은 /user/bigdata/subway/out/result1/keyword_rank.js, keyword_rank.csv이다.
- 분석 결과 파일 4개를 /home/eduuser/nia_kbig/subway/middle/visual/js 폴더로 복사한다.
- 분석 결과 파일 station_growth.csv, keyword_rank.csv를 /home/eduuser/nia_kbig/data/ 폴더로 복사한다.
- R Studio를 실행하여 R분석 스크립트인 subway.r 파일을 로드 후 실행한다.

▶ 분석 데이터 저장 방법 (05.hadoop_filecopy.sh)

▶ 데이터 저장(05.hadoop_filecopy.sh)

- 변환된 데이터를 저장하기 위해서 아래와 같이 저장 스크립트를 실행한다.
- 하둡 파일시스템에서 /home/eduuser/nia_kbig/subway/middle/visual/js/ 폴더로 파일을 가져온다.

05.hadoop_filecopy.sh (하둡파일시스템에서 서버로컬로 파일 복사)

```

01. # 하둡파일 시스템에서 로컬 파일로 다운로드 한다.
02. #역 데이터 json 파일 저장
03. $ hadoop fs -get /user/bigdata/ subway/out/result1/station_growth.js /home/
    ↪ /eduuser/nia_kbig/subway/middle/visual/js/station_growth.js
04. #역 데이터 csv 파일 저장
05. $ hadoop fs -get /user/bigdata/ subway/out/result1/station_growth.csv /hom
    ↪ e/eduuser/nia_kbig/subway/middle/visual/js/station_growth.csv
06. #키워드 json 파일 저장
07. $ hadoop fs -get /user/bigdata/ subway/out/result2/keyword_rank.js
    ↪ /home/eduuser/nia_kbig/subway/middle/visual/js/keyword_rank.js
08. #키워드 csv 파일 저장
09. $ hadoop fs -get /user/bigdata/ subway/out/result2/keyword_rank.csv /hom
    ↪ e/eduuser/nia_kbig/subway/middle/visual/js/keyword_rank.csv

```

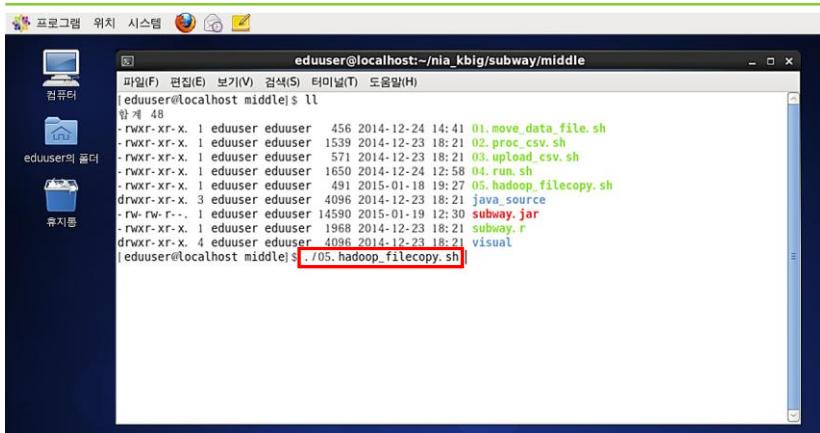


• 분석 데이터 저장 스크립트 소스(05.hadoop_filecopy.sh)

• 라인 03~09 :

- 하둡 파일시스템 저장되어 있는 승하차 증감률이 큰 역 분석 결과 데이터와 키워드 랭킹 순위 데이터를 /home/eduuser/nia_kbig/subway/middle/visual/js/ 폴더로 저장하는 라인이다.
- hadoop fs -get 하둡 파일시스템 분석데이터 경로 저장할 서버 폴더 경로

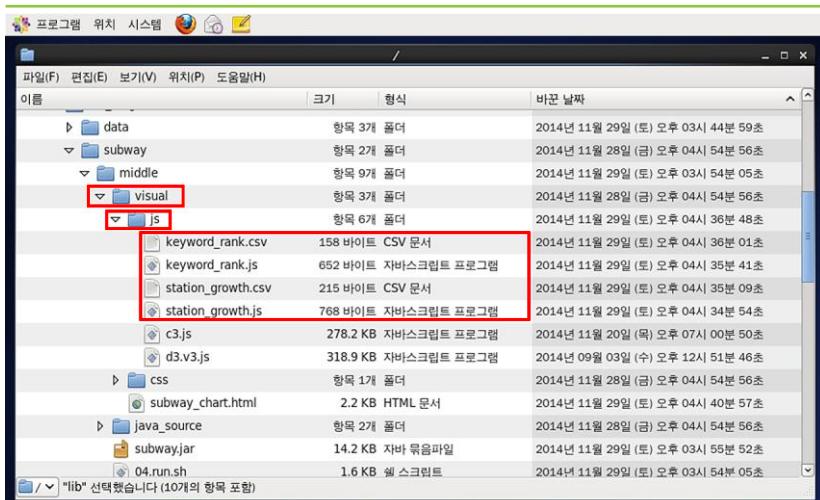
▶ 데이터 저장 스크립트 실행



```
eduuser@localhost:~/nia_kbig/subway/middle
[eduuser@localhost middle]$ ll
파일(F) 편집(E) 보기(V) 검색(S) 타이널(T) 도움말(H)
[eduuser@localhost middle]$ ll
합계 48
-rwxr-xr-x. 1 eduuser eduuser 456 2014-12-24 14:41 01.move_data_file.sh
-rwxr-xr-x. 1 eduuser eduuser 1530 2014-12-23 18:21 02.proc_csv.sh
-rwxr-xr-x. 1 eduuser eduuser 571 2014-12-23 18:21 03.upload_csv.sh
-rwxr-xr-x. 1 eduuser eduuser 1650 2014-12-24 12:58 04.run.sh
-rwxr-xr-x. 1 eduuser eduuser 491 2015-01-18 19:27 05.hadoop_filecopy.sh
drwxr-xr-x. 3 eduuser eduuser 4096 2014-12-23 18:21 java_source
-rw-rw-r--. 1 eduuser eduuser 14598 2015-01-19 12:30 subway.jar
-rwxr-xr-x. 1 eduuser eduuser 1968 2014-12-23 18:21 subway.r
drwxr-xr-x. 4 eduuser eduuser 4096 2014-12-23 18:21 visual
[eduuser@localhost middle]$ ./05.hadoop_filecopy.sh
```

- 하둡 파일 시스템에서 맵리듀스 분석 데이터를 로컬 서버 폴더 (/home/eduuser/nia_kbig/subway/middle/visual/js)로 데이터를 저장한다.
- ./05.hadoop_filecopy.sh 입력 후 엔터

▶ D3 시각화 분석 결과 데이터 저장

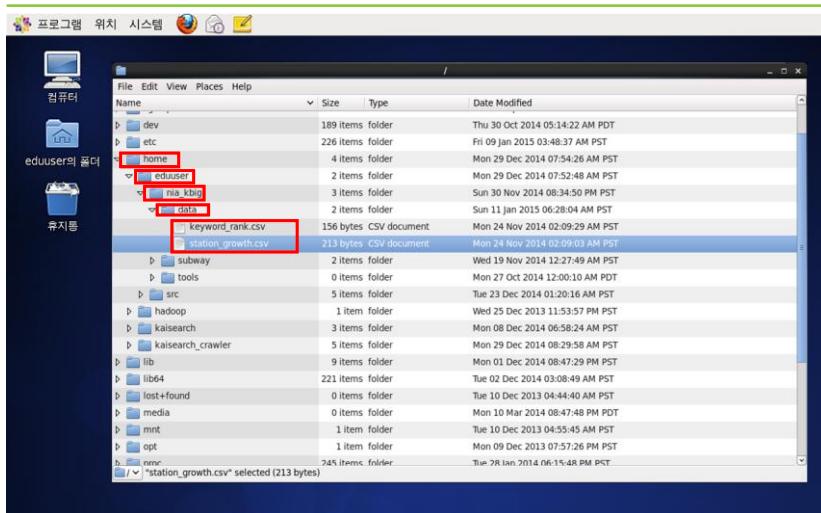


이름	크기	형식	비교 날짜
data	항목 3개 폴더		2014년 11월 29일 (토) 오후 03시 44분 59초
subway	항목 274 폴더		2014년 11월 28일 (금) 오후 04시 54분 56초
middle	항목 9개 폴더		2014년 11월 29일 (토) 오후 03시 54분 05초
visual	항목 3개 폴더		2014년 11월 28일 (금) 오후 04시 54분 56초
js	항목 674 폴더		2014년 11월 29일 (토) 오후 04시 36분 48초
keyword_rank.csv	158 바이트 CSV 문서		2014년 11월 29일 (토) 오후 04시 36분 01초
keyword_rank.js	652 바이트 자바스크립트 프로그램		2014년 11월 29일 (토) 오후 04시 35분 41초
station_growth.csv	215 바이트 CSV 문서		2014년 11월 29일 (토) 오후 04시 35분 09초
station_growth.js	768 바이트 자바스크립트 프로그램		2014년 11월 29일 (토) 오후 04시 34분 54초
c3.js	278.2 KB 자바스크립트 프로그램		2014년 11월 20일 (목) 오후 07시 00분 50초
d3.v3.js	318.9 KB 자바스크립트 프로그램		2014년 09월 03일 (수) 오후 12시 51분 46초
css	항목 1개 폴더		2014년 11월 28일 (금) 오후 04시 54분 56초
subway_chart.html	2.2 KB HTML 문서		2014년 11월 29일 (토) 오후 04시 40분 57초
java_source	항목 2개 폴더		2014년 11월 28일 (금) 오후 04시 54분 56초
subway.jar	14.2 KB 자바 런타임 파일		2014년 11월 29일 (토) 오후 03시 55분 52초
04.run.sh	1.6 KB 쉘 스크립트		2014년 11월 29일 (토) 오후 03시 54분 05초

"/lib" 선택했습니다 (10개의 항목 포함)

- visual/js 폴더에 담색기로 보면 keyword_rank.csv, keyword_rank.js, station_growth.csv, station_growth.js 의 4개 파일이 복사되어 있다.
- 시각화에 사용되는 파일은 keyword_rank.js, station_growth.js 2개 파일이 D3 차트에서 사용된다.

▶ R 시각화 분석 결과 데이터 저장



- 하둡 파일시스템에서 복사한 데이터 위치 폴더(/home/eduuser/nia_kbig/subway/middle/visual/js/)로 이동하면 keyword_rank.csv, station_growth.csv 파일이 위치해 있다.
- R 시각화에 사용되는 파일은 keyword_rank.csv, station_growth.csv 2개 파일을 /home/eduuser/nia_kbig/data/ 폴더로 복사한다.
- R Studio에서 2개 파일을 로드 후 분석을 실행한다.

> 시각화 과정

> 시각화 절차



> 시각화 과정

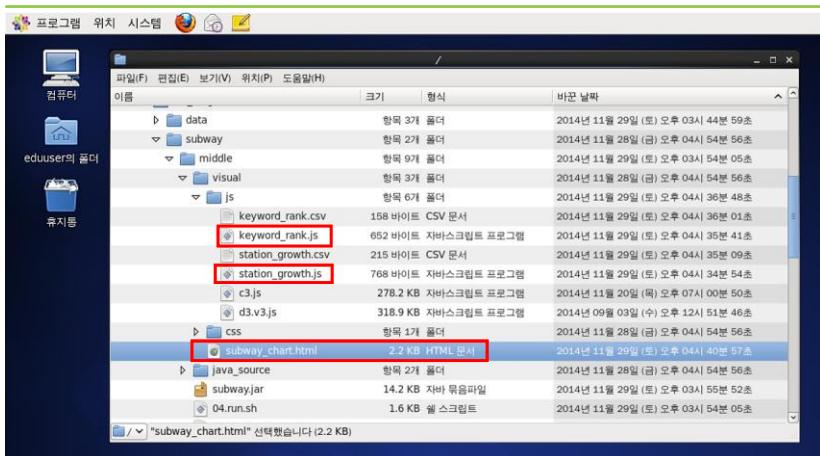
- d3.v3.js 라이브러리 파일을 제공 사이트에서 다운로드하여 저장한다.
- 시각화할 HTML 페이지를 생성한다.
- D3 Chart 라이브러리 모듈을 페이지에 삽입한다.
- 승하차 증가 차트와 키워드 분석 차트를 연계한다.
- D3 Chart Data를 읽어 오는 부분(station_growth.js, keyword_rank.js)에 결과 데이터를 삽입한다.
- X축과 Y축의 값을 지정한다.
- html 페이지를 웹 브라우저에서 실행한다.

> D3 Chart 모듈 삽입과 결과 데이터 삽입

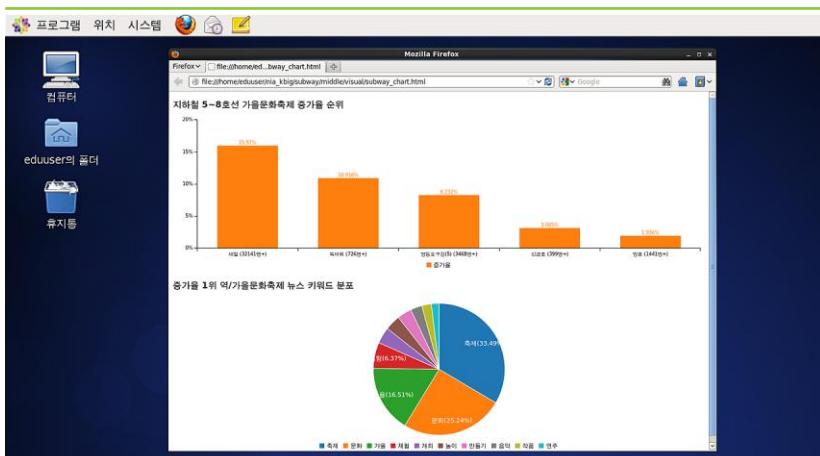
```
<!-- d3 모듈을 불러온다 -->
<script src="js/d3.v3.js"></script>
<!--d3 Chart data 연계 -->
<script src="js/station_growth.js"></script>
<script src="js/keyword_rank.js"></script>
```

> 분석 데이터 시작화

> D3 차트 시작화



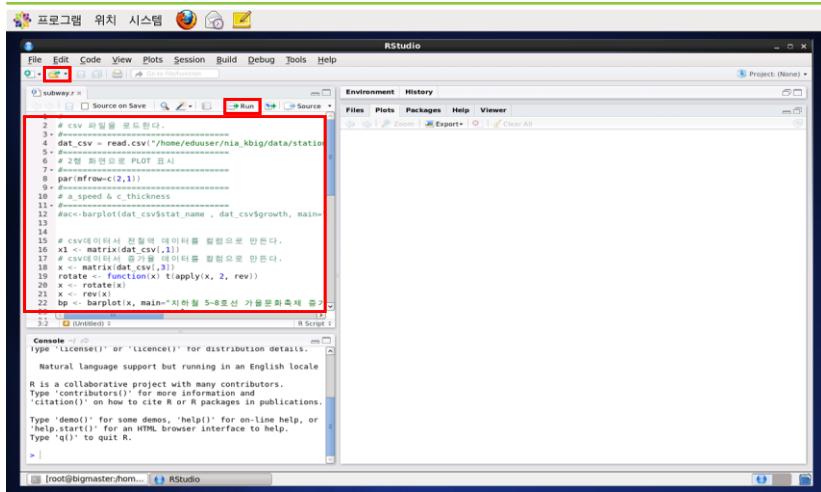
- keyword_rank.js, station_growth.js 파일 2개 복사가 완료되면 바로 하단에 위치에 있는 subway_chart.html 파일을 더블 클릭하여 ‘표시’ 버튼을 클릭하면 브라우저로 파일이 열린다.



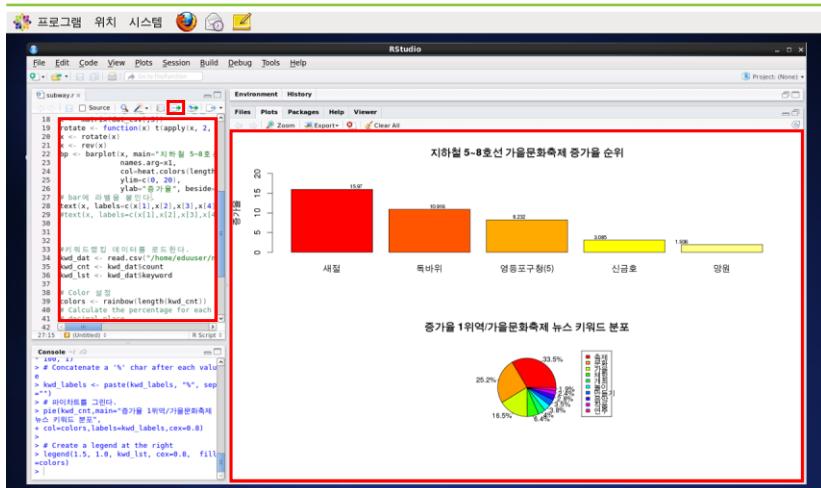
- 지하철 5~8호선의 가을 문화축제 기간 중 승하차 증감률이 가장 큰 역을 상위 5개 역을 출력한다. 1위 세종역과 관련하여 뉴스 키워드 분석 결과를 파이 차트로 출력하여 키워드 랭킹을 확인할 수 있다.

VI. 시각화

> R 시작화



- R Studio 실행 후 /home/eduuser/nia_kbig/subway/middle/ 폴더에 있는 subway.r 스크립트를 불러온다.



- subway.r 스크립트를 실행을 하면 지하철 승하차 증감율이 가장 큰 상위 5개역이 막대그래프로 출력이 되고 관련 뉴스 키워드 분포가 파이차트로 출력이 된다.

> 2013년도 지하철 5~8호선 가을문화축제 주요행사계획

호선	역명	일정	행사내용
5호선	김포공항	10.26(토)~27(일) 13:00~19:00	전통 칠교놀이 체험전
	화곡	10.27(일) 15:00~17:00	클래식&가요&팝 오케스트라 연주회
	오목교	10.26(토) 17:00~19:00	관현악과 성악이 어우러진 오목교역 작은음악회
	영등포구청	10.28(월)~11.2(토)	산모를 위한 힐링 – 포대기, 아기용품만들기 체험 및 취업
	여의도	11.2(토) 10:30~11:00	여의도중학교 음악동아리 <동감> 재능기부공연 – "사랑합니다! 감사합니다!"
	서대문	10.28(월)~11.1(금)	현대인 맞춤형 건강상담
	청구	10.30(수) 10:00~12:00	시민과 함께하는 무료법률상담
	신금호	10.24(목) 14:00~16:00	현대인에게 꼭 필요한 정신건강검진(우울증, 스트레스, 일코올)
	왕십리	10.30(수) 17:00~18:00	하모니카와 떠나는 신나는 가을여행
	왕십리	11.2(토) 17:00~19:00	팝핀댄스공연 및 통기타연주
	군자역	10.24(목)~11.2(토)	시(詩) 전시회
	아차산	10.24(목)~11.1(금)	선희예술고등학교 재학생 23명의 미술작품 전시회
	천호	11.2(토) 15:00~16:00	수화공연 및 어르신 합창단 공연
	강동	10.30(수) 19:00~20:30	"강동드림뮤즈" 동호회 기울음악회
	굽은다리	10.24(목)~10.28(월)	만화, 애니메이션 그림전시 및 캐리커처 체험
	명일	10.24(목) 10:30~11:30	심폐소생술 응급조치요령 시민체험
	상일동	10.29(화) 15:00~17:00	관악합주 및 밴드공연 & 디자인 작품 전시회
	둔촌동	10.26(목) 16:00~18:00	가을맞이 아코디언 연주회
	거여	11.2(토) 15:00~16:00	합창과 함께하는 색소폰과 국악의 만남
6호선	새절	10.25(금)~/28(월)/30(수) 11.1(금) 18:00~19:00	웃음힐링강의 – 10월25일(박장대소), 10월28일(포복절도) 10월30일(요절복통), 11월01일(파안대소)
	망원	10.24(목)~10.26(토)	수공예품전시회 – 천연화장품/손세정제 만들기 체험
	상수	10.25(금) 18:30~19:30	2013. 마포문화재단 찾아가는 문화공연 – 5인조어쿠스틱공연
	대흥	10.24(목) 19:40~20:20	통기타와 챔버를 이용한 주민과 함께하는 작은음악회
	녹사평	10.31(목) 16:00~17:00	어르신 문화예술공연
	독바위	10.26(토) 10:00~11:00	북한산 둘레길 소개 및 안전한 산행 위한 심폐소생술교육
	월곡	10.28(월) 16:00~18:00	깊어가는 가을 월곡역 하모니카 연주회
7호선	도봉산	10.24(목)~11.2(토)	2013국제효만화공모전 당선작품 전시회
	공릉	10.26(목) 17:00~18:30	사물놀이, 아코디언, 오케스트라, 색소폰, 난타, 오카리나 등
	뚝섬유원지	10.30(수) 18:00~19:00	오선지와 기타사랑의 가을 음악회
	논현	11.1(금) 18:00~20:00	가을음악콘서트-싱어송라이터와 어쿠스틱기타의 만남
	광명사거리	10.28(월) 17:30~18:00	"행복더하기 + 나누기" – 어르신댄스(스포츠댄스, 라인댄스), 어린이합창
	부천시청	10.25(금) 19:00~20:00	부천시공무원이 꾸미는 7080노래 및 색소폰연주회
	상동	11.1(금) 19:00~20:00	통기타와 타악기(퍼쿠션)를 이용한 힘이 나는 음악회
8호선	송파	11.1(금) 14:00~16:00	색소폰 및 클라리넷 연주회 – 다양한 연령대를 위한 트롯, 가요 등 연주
	가락시장	10.26(토) 14:00~16:00	건전가요 중심의 만들린 연주회 & 누구나 함께하는 다과회 개최
	장지	10.26(토) 16:20~17:30	남녀보컬과 어쿠스틱세션을 통해 대중음악
	모란	10.26(토) 19:00~20:00	팬플룻 연주, 난타연주

I.개요

II.수집

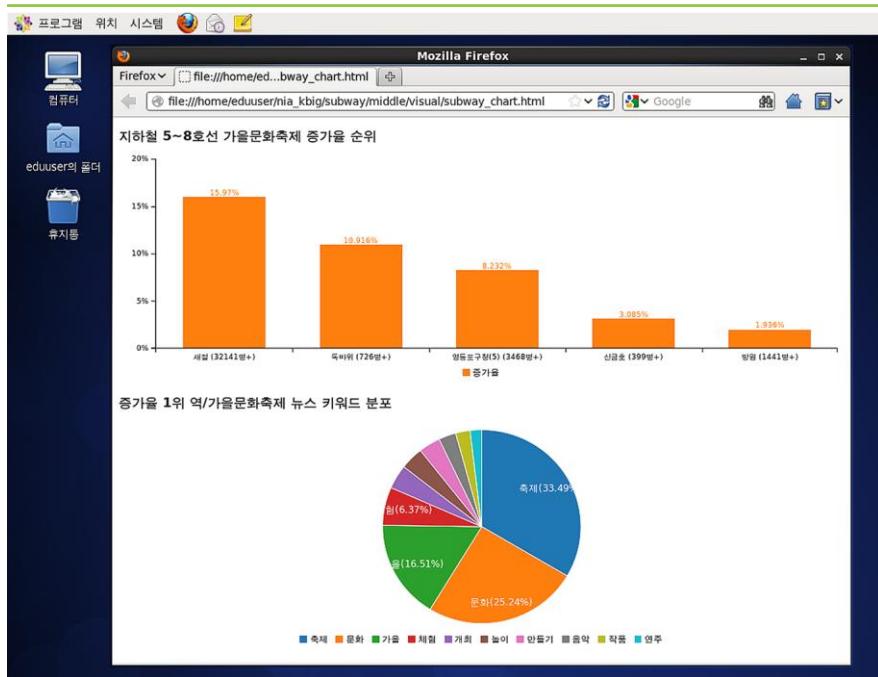
III.기공

IV.저장

V.분석

VI.시각화

> 데이터 분석



- 지하철 5~8호 가을 문화축제 기간 (2013.10.24 ~2013.11.02) 중 전주 대비 승하차율이 상승한 역은 새절(15.97%), 독바위(10.916%), 영등포구청(5)(8.232%), 신금호(3.085%), 망원(1.936%) 순으로 분석되었다.
- 주요 행사 계획표에서 새절역의 가을 문제 축제 행사는 웃음 힐링 강의가 4일 동안 진행되어 다른 역보다 행사를 많이 진행하였으며, 독바위역 행사는 북한산 둘레길 소개 및 안전한 산행을 위한 심폐소생술 체험 교육, 영등포구청(5) 역은 산모를 위한 힐링 행사였다.
- 증가율 1위 역과 가을 문화축제와 관련하여 뉴스 키워드 검색을 분석할 결과는 축제(44.49%), 문화(25.24%), 가을(16.51%), 체험(16.51%) 순으로 분석되었다.
- 지하철 가을 문화축제가 승객의 승하차에 많은 영향을 주지는 못하지만 몇 개 역에는 행사와 관련하여 소폭 증가하는 패턴을 찾아볼 수가 있다.

I. 개요

II. 수집

III. 가공

IV. 저장

V. 분석

VI. 시각화



VII 예제문제

예제 문제1

115

예제 문제2

116

예 / 제 / 문 / 제

예제 1

여의도 벚꽃축제 기간의 승하차 인원의 증가율과 소셜 데이터를 연계하여 분석하라.

- 2013년도 여의도 벚꽃축제 기간의 5호선 여의도역, 여의나루역의 승하차 인원과 소셜 분석을 하여 연관 분석하라.

- 2013년도 여의도 벚꽃축제 기간의 날짜를 파악한다.
- 2013년도 벚꽃축제 기간의 5호선 여의도역과 여의나루역의 승하차 인원을 추출한다.
- 축제 기간 여의도역과 여의나루역에 대한 일자별 승차, 하차 인원의 합계를 구한다.
- 2013년도 축제 기간의 소셜 데이터에서 키워드를 추출하여 순위를 합산한다.
- 승하차 인원 데이터와 소셜 데이터 분석 결과를 D3 차트로 시각화한다.

예제 2

2009 ~2013년도까지 호선별 수송인원 점유율의 추이를 분석하라.

- 2009 ~2013년도의 승하차 인원을 호선별(5~8) 점유율을 분석하라.

- 2009~2013년도 호선별 승하차 인원을 추출한다.
- 호선별 연도별 인원을 합산한다.
- 호선별 연도별 합산 데이터를 백분율로 환산한다.
- 각 호선별 수송률을 계산한다.
- 호선별 연도별 수송률을 차트로 시각화한다.

데이터 분석 콘텐츠 활용 매뉴얼

2014년 12월 인쇄

2015년 1월 발행

발 행 처 한국정보화진흥원 빅데이터전략센터

집 필 신신애, 김성현, 박재원, 김현태, 김지홍, 정다운,
이승하, 신은비

주 소 서울시 중구 청계천로 14

연 락 처 (02) 2131-0114

인 쇄 HNJ Printing

〈비매품〉

[데 이 터 분 석 콘 텐 츠]
활용 매뉴얼

NIA  한국정보화진흥원

(100-775) 서울시 종구 청계천로 14 한국정보화진흥원
TEL 02-2131-0114 FAX 02-2131-0109
www.nia.or.kr

