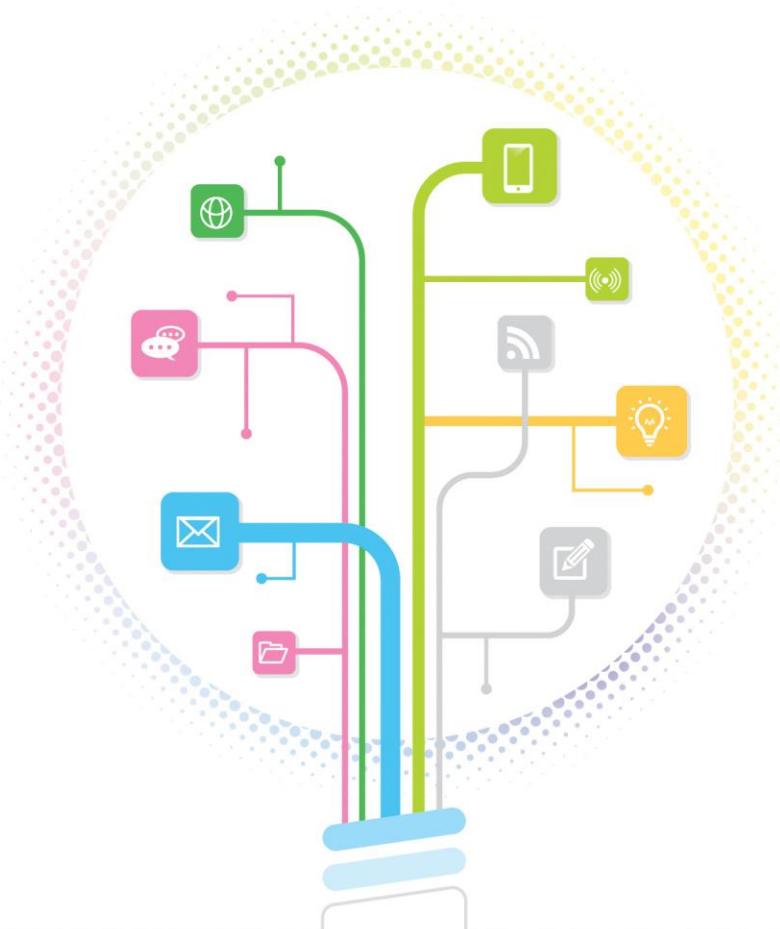


데이터 분석 콘텐츠 활용 매뉴얼



미래창조과학부



한국정보화진흥원



KBIG
빅데이터
전략센터

CONTENTS

Beginning Level 초급과정

I 개요

개요	9
----	---

II 수집

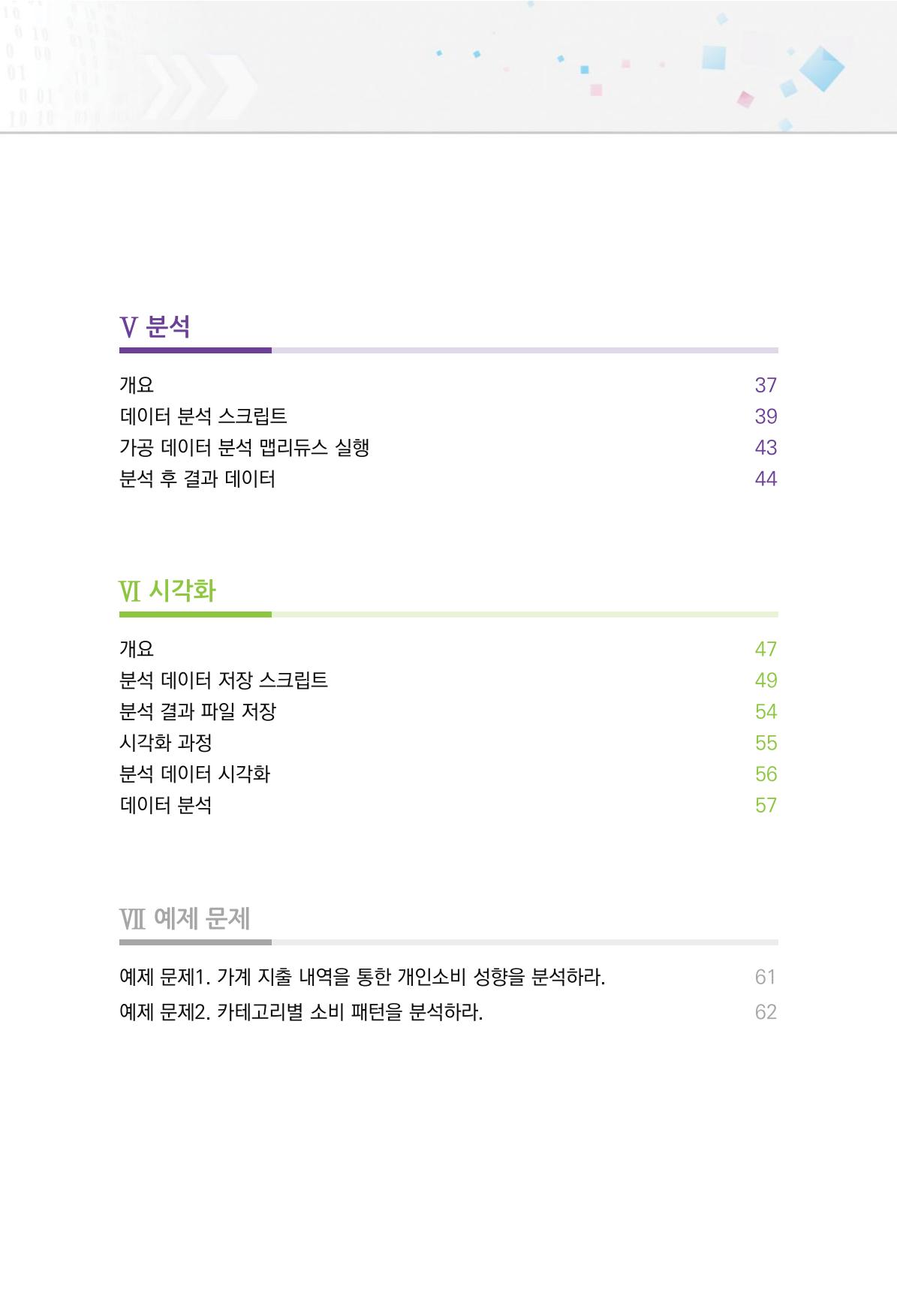
개요	13
교육용 데이터 샘플	14
데이터 수집	15
데이터 작업 영역 이동 스크립트	18

III 가공

개요	23
데이터 가공 스크립트	24

IV 저장

개요	29
가공 데이터 저장	30
MongoDB 저장 데이터 조회	32



V 분석

개요	37
데이터 분석 스크립트	39
가공 데이터 분석 맵리듀스 실행	43
분석 후 결과 데이터	44

VI 시각화

개요	47
분석 데이터 저장 스크립트	49
분석 결과 파일 저장	54
시각화 과정	55
분석 데이터 시각화	56
데이터 분석	57

VII 예제 문제

예제 문제1. 가계 지출 내역을 통한 개인소비 성향을 분석하라.	61
예제 문제2. 카테고리별 소비 패턴을 분석하라.	62

CONTENTS

Intermediate Level 

I 개요

개요	67
----	----

II 수집

개요	71
교육용 데이터 샘플	72
데이터 수집	74
데이터 작업 영역 이동 스크립트	77

III 가공

개요	81
데이터 가공 스크립트	83

IV 저장

개요	89
가공 데이터 하둡 파일시스템 업로드	90
가공 데이터 하둡 파일시스템 저장	91
하둡 파일 시스템 파일 조회	92
하둡 명령어로 파일 조회	93

V 분석

개요	97
데이터 분석 스크립트	99
데이터 분석 맵리듀스 실행	101
분석 데이터 파일 조회	102
분석 후 결과 데이터	103

VI 시각화

개요	107
분석 데이터 저장 방법 1	108
분석 데이터 저장 방법 2	110
시각화 과정	114
분석 데이터 시각화	115
데이터 분석	116

VII 예제 문제

예제 문제1. 카테고리 소비 지출 패턴과 강우량과의 비교 분석하라.	119
예제 문제2. 사용자별, 분류별, 일자별 가계 지출 패턴을 분석하라.	120



소비 

Beginning Level

초급과정







I 개요

개요

9

8

I

개요

> 개요

개인별 소비지출 정보인 소비(가계부 정보) 데이터를 바탕으로 개인이 분야별 지출한 비용 중 2013년 데이터 정보를 추출하여 개인의 소비 패턴을 일자별, 카테고리 분류 항목별로 그룹화하여 소비패턴을 알아 보고자 한다. 분석에 사용할 데이터는 임의 사용자를 선정하여 2013년도 데이터만을 가공하여 사용하며, 시계열 분석을 통하여 2013년도 월별 소비 형태의 패턴 분석을 하고자 한다. 이러한 방법으로 개인의 소비 패턴 분석을 통해 개인의 소비지출에 대한 소비 성향 분석과 소비 지출에 대한 예측을 통하여 소비 지출을 줄일 수 있는 참고 자료로 활용한다.

> 활용 데이터

- **card_trade.csv** : 2013년 카드 사용 정보 데이터
- **code.csv** : 코드 정보 데이터

> 선행학습

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

▶ 요구사항

- 1인 사용자(식별코드: 20140209004745257016)의 2013년 지출 정보를 추출하여 3개 분야인 마트/쇼핑(CA), 외식/부식(CB), 커피/간식(CC) 분야에 대한 지출 변화를 시각화하여 소비 패턴을 분석하라.

▶ 분석 절차

- 수집된 소비 가계부 정보 데이터를 로드한다.
- 제공 되어진 식별코드가 “20140209004745257016”에 해당하는 사용자의 2013년도 지출 정보를 시계열 분석에 용이한 데이터 형태로 변화하기 위해 추출하여 가공된 데이터를 CSV 파일로 저장한다.
- 식별코드(20140209004745257016)에 해당하는 사용자의 분야별 지출 비용을 시계열 분석의 패턴 분석에 용이한 형태로 일자 별로 그룹화하여 합산한다.
- 시계열 분석의 패턴 분석에 용이한 형태로 지출 분야별, 일자 별 합산된 데이터를 MongoDB 컬렉션에 저장한다.
- MongoDB에 분석된 데이터를 엑셀 형식(CSV)나 D3 차트 형식(JSON)으로 데이터를 저장한다.
- 일자 별 분야 별 지출 정보의 패턴을 분석하기 위해 시계열 분석을 사용한다.
- CSV, JSON 파일로 저장된 데이터를 불러와서 엑셀이나 D3 차트로 꺾은선 그래프를 사용하여 시각화한다.
- 시각화를 통해 개인의 2013년도 월별 소비 형태를 분야별로 패턴 분석한다.



1

2

II 수집

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



수집

▶ 개요

소비 데이터는 국내 XX카드사에서 제공받은 2012~2013년 개인별 소비 지출 정보인 소비(가계부 정보) 데이터를 수집하여 분석 목적을 달성할 수 있는 한도 내에서 개인 정보, 카드 정보 등을 비식별화 처리를 통해 분석에 용이하게 편집하여 제공한다. 소비 가계부 정보 데이터는 개인이 카드 사용 시 지출되는 비용을 문자로 전송 받은 데이터를 스마트 폰 가계부 앱을 통해서 수집되는 데이터이다. 가계부 프로그램을 통해서 개인이 가계부를 별도로 작성할 수 있으면 작성된 데이터를 엑셀로 다운로드해서 분석 데이터 셋 자료로 활용할 수 있게 편집하여 제공한다.

▶ 수집 방법

- **데이터 제공** : 소비 가계부 정보 데이터는 국내 XX카드사에서 제공하는 데이터를 OpenAPI, 자료수집기(Crawler)를 통하여 데이터를 수집하였고, 실습용 자료는 빅데이터 분석 활용센터에 접속하여 유통 데이터 셋을 다운로드할 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

▶ 가계부 데이터(card_trade.csv)

거래정보	사용자식별 코드	카테고리	문자수신 일자	결제타입	거래타입	결제 코드	사용처	지출 비용	거래일시
2013121017 5053101518	2013121017 5053253232	CZ9 999	2013111 4174310	PA 12	TT 01	BC140	밝은세상이비	57,900	2013103 1173000
2013121017 5053101519	2013121017 5053253232	CA9 999	2013111 4174251	PA 03	TT 01	BC141	(주)농협유통 양재하나로클	215,510	2013102 6185900
2013121017 5054101520	2013121017 5053253232	CA9 999	2013111 4174231	PA 03	TT 01	BC141	임광마트	11,700	2013102 4195600
2013121017 5054101521	2013121017 5053253232	CB9 999	2013111 4174210	PA 03	TT 01	BC141	김경자소문난 대구왕볼점	48,000	2013102 4194700
2013121017 5054101522	2013121017 5053253232	CA9 999	2013111 4174151	PA 03	TT 01	BC141	씨엔에스유통 (주)삼익주유	49,000	2013102 3065700
2013121017 5054101523	2013121017 5053253232	CE0 011	2013111 4174129	PA 03	TT 01	BC141	메디팜화자 약국	19,000	2013102 1195800
2013121017 5054101524	2013121017 5053253232	CA0 081	2013111 4174113	PA 12	TT 01	BC140	씨유양재알뜰 점	14,000	2013102 1132300
2013121017 5054101525	2013121017 5053253232	CI9 999	2013111 4174047	PA 03	TT 01	BC141	솔대어린이집 [아이사랑]	253,000	2013102 1072800
2013121017 5054101526	2013121017 5053253232	CF9 999	2013111 4174022	PA 03	TT 01	BC141	리헤어겔러리	10,000	2013102 0183500
2013121017 5054101527	2013121017 5053253232	CA0 040	2013111 4174001	PA 03	TT 01	BC141	대보유통(주)/ 화성하주유소	80,000	2013101 8025000
2013121017 5054101528	2013121017 5053253232	CA9 999	2013111 4173857	PA 03	TT 01	BC141	씨엔에스유통 (주)삼익주유	48,000	201310 4060700
2013121017 5054101529	2013121017 5053253232	CZ9 999	2013111 4173830	PA 03	TT 01	BC141	OK25	4,600	201310 2090100
2013121017 5054101530	2013121017 5053253232	CD9 999	2013111 4173808	PA 03	TT 01	BC141	점풀린파크 (복수원)	4,000	201310 3152700

▶ 코드설명 데이터(code.csv)

코드	코드설명	코드	코드설명	코드	코드설명
CA0000	마트/쇼핑	CF0000	미용/뷰티	CK0000	외환/해외
CB0000	외식/부식	CG0000	교통/주유	CL0000	기타
CC0000	커피/간식	CH0000	주거/생활	CZ9999	미지정
CD0000	레저/문화	CI0000	교육/학원		
CE0000	건강/의료	CJ0000	보험/세금		

I. 개요

II. 수집

III. 기공

IV. 저장

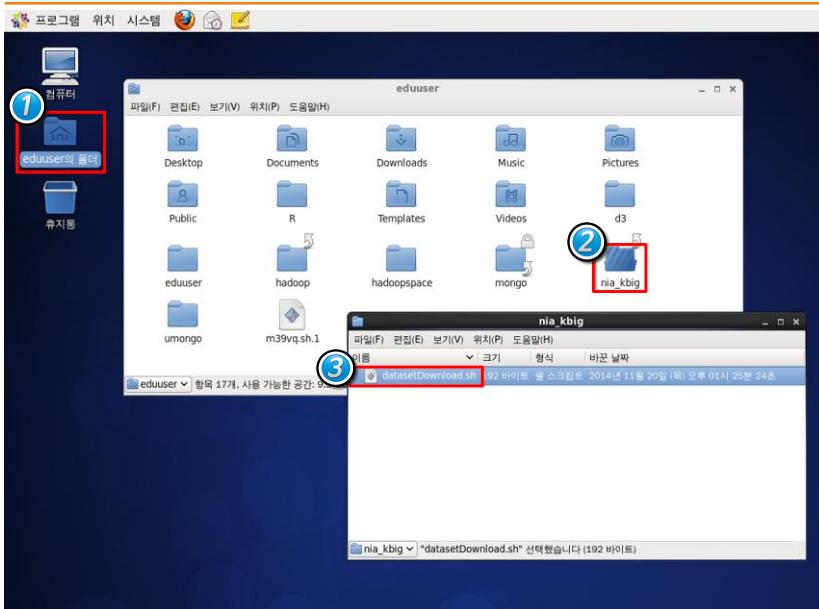
V. 분석

VI. 시각화

▶ 데이터 수집(datasetDownload.sh)

- 데이터 저장소에서 서버 로컬로 소비 가계부 데이터 셋을 복사해 온다.
 - **card_trade.csv** : 2013년 카드 사용 정보 데이터
 - **code.csv** : 코드 정보 데이터

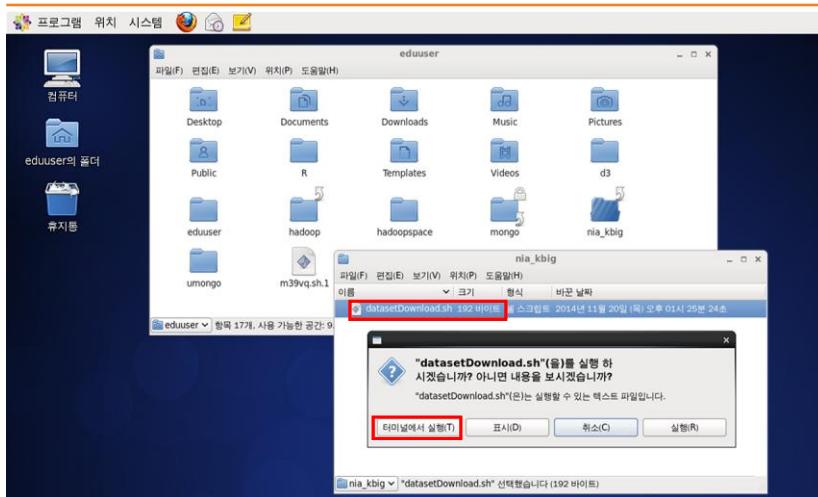
▶ 실습코드 디렉토리로 이동



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

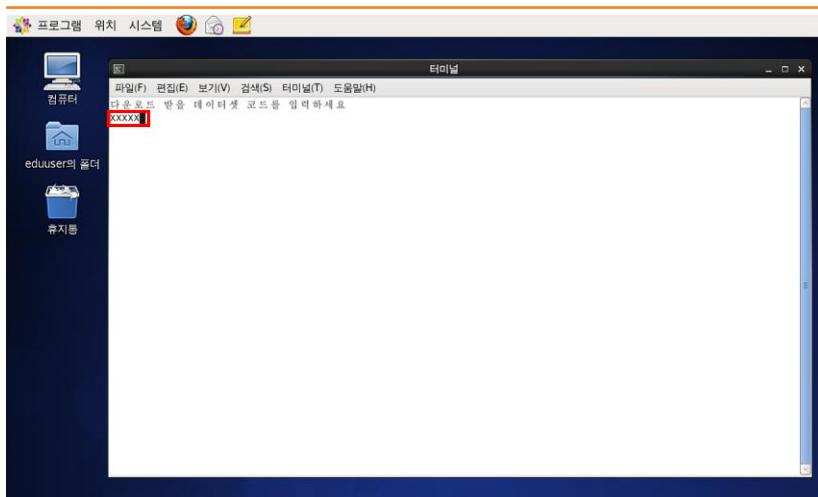
▶ 레파지토리에서 데이터 수집

datasetDownload.sh (원시데이터로 컬서버로 복사)



- '터미널에서 실행' 버튼을 클릭한다.

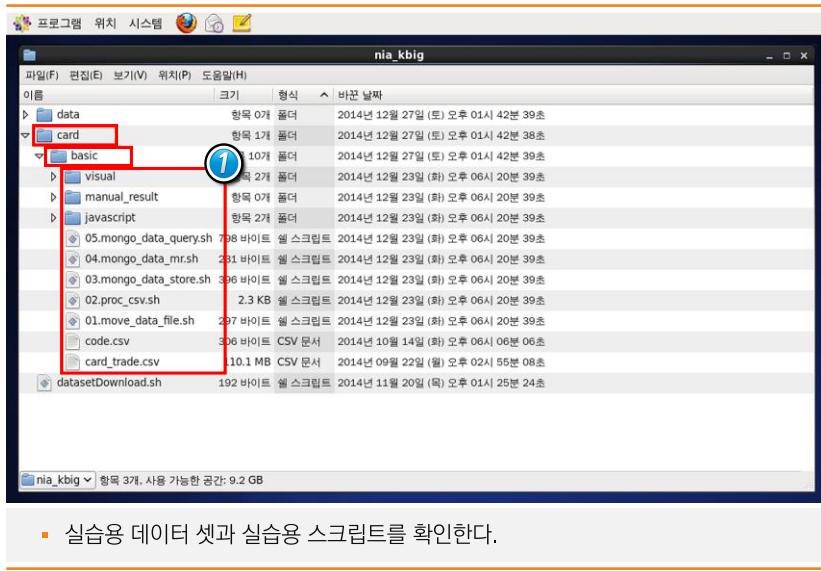
▶ 데이터셋 코드 입력



- 다운로드 받은 데이터셋 코드를 입력 후 엔터

II. 수집

▶ 데이터셋과 실습용 쉘 스크립트



▶ 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

▪ code.csv : 카테고리 분류 코드 데이터

▪ card_trade.csv : 소비 가계부 데이터



- 데이터 작업 영역 이동 스크립트 소스(01.move_data_file.sh)
- 라인 03~04 : 다운로드한 원시데이터 card_trade.csv, code.csv 파일을 설정하는 라인이다.
- 라인 06 : 작업 폴더를 설정하는 라인이다.
- 라인 07~08 : 작업 폴더로 다운로드한 원시데이터를 이동하는 라인이다.

▶ 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)

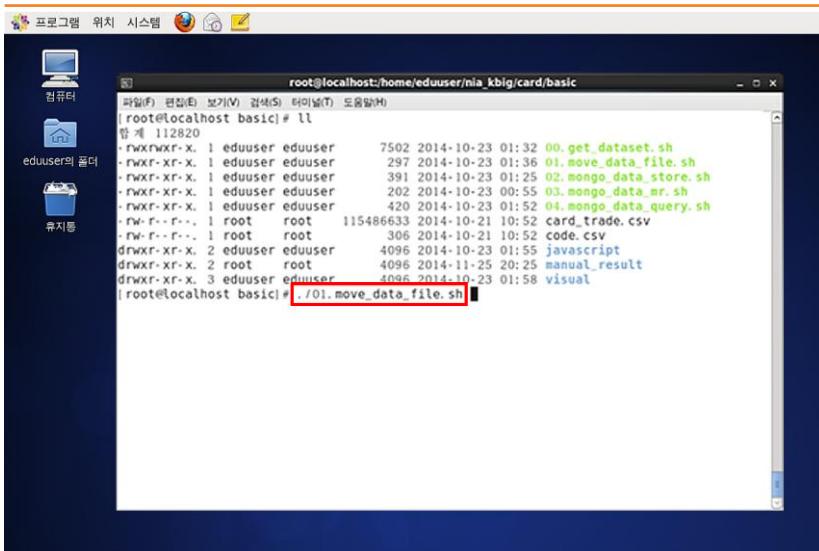
```

01.#!/bin/bash
02. # 복사 대상 파일 정의
03. TARGET_PRODUCT_PRICE=/home/eduuser/nia_kbig/card/basic/card_trade.csv
04. TARGET_CODE=/home/eduuser/nia_kbig/card/basic/code.csv
05. # 작업 디렉토리 정의
06. LOCAL_DIR=/home/eduuser/nia_kbig/data/
07. mv $TARGET_PRODUCT_PRICE $LOCAL_DIR
08. mv $TARGET_CODE $LOCAL_DIR

```

II. 수집

➤ 수집 데이터셋 작업 영역 폴더 이동



The screenshot shows a terminal window titled 'root@localhost basic' with the command 'ls' run. The output lists several files and their details:

```
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(M)
합계 112820
-rwxrwxr-x 1 eduuser eduuser      7502 2014-10-23 01:32 00.get_dataset.sh
-rwxr-xr-x 1 eduuser eduuser      297 2014-10-23 01:36 01.move_data_file.sh
-rwxr-xr-x 1 eduuser eduuser      391 2014-10-23 01:25 02.mongo_data_store.sh
-rwxr-xr-x 1 eduuser eduuser      202 2014-10-23 00:55 03.mongo_data_mr.sh
-rwxr-xr-x 1 eduuser eduuser      420 2014-10-23 01:52 04.mongo_data_query.sh
-rw-r--r-- 1 root   root       115486633 2014-10-21 10:52 card.trade.csv
-rw-r--r-- 1 root   root        306 2014-10-21 10:52 code.csv
drwxr-xr-x 2 eduuser eduuser     4096 2014-10-23 01:55 javascript
drwxr-xr-x 2 root   root     4096 2014-11-25 20:25 manual_result
drwxr-xr-x 3 eduuser eduuser     4096 2014-10-23 01:58 visual
```

In the bottom right corner of the terminal window, there is a red rectangular box highlighting the command `./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

작업영역 폴더에 복사한 소비 데이터(card_trade.csv)의 가공은 전처리 단계에서 수집된 데이터에서 사용자 1인 대상자를 선정하여 소비 지출 정보 데이터 중 2013년도 데이터만을 추출하여 시계열 분석의 패턴 분석에 유용한 객체 형태로 변환하도록 한다.

> 가공 방법

- 분석 대상은 소비 가계부 데이터(card_trade.csv) 파일에서 식별코드가 “20140209004745257016”인 사용자의 2013년도 소비 지출 데이터만 추출하여 시계열 분석의 패턴분석에 유용한 객체 형태로 변환하여 “2013_card_trade.csv” 파일을 생성한다.

> 데이터셋

거래정보	사용자식별 코드	카테고리	결제코드	사용처	지출비용	거래일시
2013121017 5053101518	2013121017 5053253232	CZ9999	BC140	밝은세상이비	57,900	20131031173000
2013121017 5053101519	2013121017 5053253232	CA9999	BC141	(주)농협유통 양재하나로클	215,510	20131026185900
2013121017 5054101520	2013121017 5053253232	CA9999	BC141	임광마트	11,700	20131024195600
2013121017 5054101521	2013121017 5053253232	CB9999	BC141	김경자소문난 대구왕뽈찜	48,000	20131024194700
2013121017 5054101522	2013121017 5053253232	CA9999	BC141	씨앤텍스유통 (주)삼익주유	49,000	20131023065700

> 데이터 가공 스크립트 (02.proc_csv.sh)

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

02.proc_csv.sh (원시데이터에서 분석할 대상을 추출하여 저장)

```

01.#!/bin/bash
02. # 입력 CSV 파일 지정
03. INPUT_FILE='/home/eduuser/nia_kbig/data/card_trade.csv'
04. # 출력결과 CSV 파일 지정
05. OUTPUT_FILE='/home/eduuser/nia_kbig/data/2013_card_trade.csv'
06. # 분석할 연도 설정
07. TARGET_YEAR='2013'
08. # HEADER컬럼 출력
echo "u_trade_no,user_sid,category_type,corp_id,sms_receive_dt,pay_type
    ↪ ,trade_type,pay_cd,pay_account,trade_site_nm,trade_site_id,trade_mo
    ↪ ney,trade_dt,quota_month,balance_money,point_type,add_point,use_p
    ↪ oint,sms_org,car_fill_yn,online_site_yn,online_site_id,company_card_y
    ↪ n,inarea_yn,foreign_amount,foregin_unit,auto_pay_seq,parse_seq,callb
    ↪ ack_num,category_nm,memo,pay_nm,result_parse,reg_dt,trade_stat,
    ↪ use_yn,reg_type,sms_send_id,ref_u_trade_no,mod_dt,user_nm,check
    ↪ _money,card_add_money,notapply_check_yn,notapply_check_dt,app_yn,
    ↪ autopay_type,autopay_remaind_cnt,join_type,trade_yn" > $OUTPUT_FILE
10. #','를 구분자로 해서 파일을 읽어들인다.
11. IFS=':'
```



- 데이터 가공 스크립트 소스(02.proc_csv.sh)
- 라인 03~05 : 가공 대상인 소비가계부 데이터(card_trade.csv) 지정하고, 가공데이터를 2013_card_trade.csv 파일로 저장하는 라인이다.
- 라인 07 : 가공 연도 설정을 2013년도로 설정하는 라인이다.
- 라인 09 : 가공데이터 파일을 생성시 상단에 Header 정보를 추가하는 라인이다.
- 라인 11~15 : 원시데이터 파일을 1라인씩 읽어서 2013년도 해당하는 데이터만을 선택하는 라인이다.

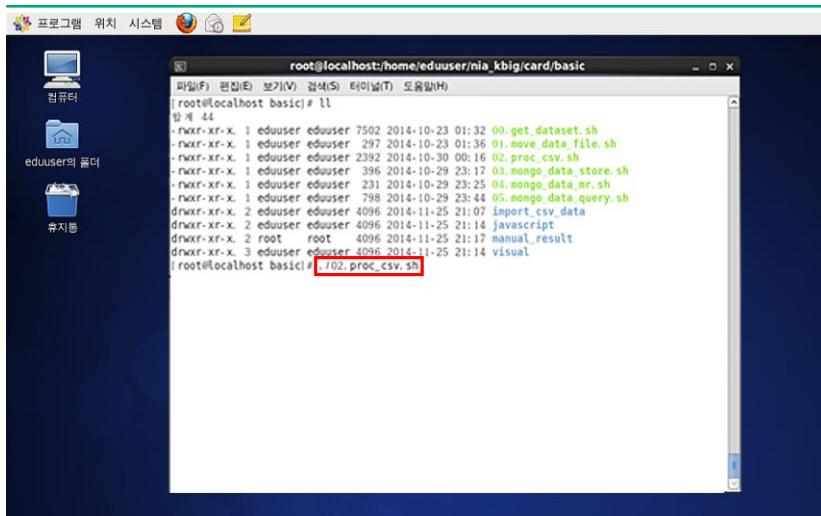
III. 가공

```
12.     while read U_TRADE_NO USER_SID CATEGORY_TYPE CORP_ID SMS_RECEI  
13.         ↳ VE_DT PAY_TYPE TRADE_TYPE PAY_CD PAY_ACCOUNT TRADE_SITE_N  
14.             M TRADE_SITE_ID TRADE MONEY TRADE_DT QUOTA_MONTH BALAN  
15.             CE_MONEYPOINT_TYPE ADD_POINT USE_POINT SMS_ORG CAR_FILL_  
16.             YN ONLINE_SITE_YN ONLINE_SITE_ID COMPANY_CARD_YN INAREA_Y  
17.             N FOREIGN_AMOUNT FOREIGN_UNIT AUTO_PAY_SEQPARSE_SEQ CAL  
18.             LBACK_NUM CATEGORY_NM MEMO PAY_NM RESULT_PARSE REG_DT  
19.             TRADE_STAT USE_YNREG_TYPE SMS_SEND_ID REF_U_TRADE_NO MO  
20.             D_DT USER_NM CHECK_MONEYCARD_ADD_MONEY NOTAPPLY_CHEC  
K_YN NOTAPPLY_CHECK_DTAPP_YN AUTOPAY_TYPE AUTOPAY_REMAI  
ND_CNTJOIN_TYPE TRADE_YN  
  
13.     do  
14.         # TRADE_DT가 TARGET_YEAR로 시작하는 년도인지 체크한다.  
15.         if [[ $TRADE_DT == ${TARGET_YEAR}* ]]; then  
16.             # 해당년도의 데이터만을 CSV로 출력한다.  
17.             echo "$U_TRADE_NO,$USER_SID,$CATEGORY_TYPE,$CORP_ID,$SM  
S_RECEIVE_DT,$PAY_TYPE,$TRADE_TYPE,$PAY_CD,$PAY_ACCO  
UNT,$TRADE_SITE_NM,$TRADE_SITE_ID,$TRADE_MONEY,$TRA  
DE_DT,$QUOTA_MONTH,$BALANCE_MONEY,$POINT_TYPE,$AD  
D_POINT,$USE_POINT,$SMS_ORG,$CAR_FILL_YN,$ONLINE_SITE  
_YN,$ONLINE_SITE_ID,$COMPANY_CARD_YN,$INAREA_YN,$FOR  
EIGN_AMOUNT,$FOREGIN_UNIT,$AUTO_PAY_SEQ,$PARSE_SEQ,  
$CALLBACK_NUM,$CATEGORY_NM,$MEMO,$PAY_NM,$RESULT  
_PARSE,$REG_DT,$TRADE_STAT,$USE_YN,$REG_TYPE,$SMS_SE  
ND_ID,$REF_U_TRADE_NO,$MOD_DT,$USER_NM,$CHECK_MON  
EY,$CARD_ADD_MONEY,$NOTAPPLY_CHECK_YN,$NOTAPPLY_C  
HECK_DT,$APP_YN,$AUTOPAY_TYPE,$AUTOPAY_REMAIND_CNT  
,${JOIN_TYPE},$TRADE_YN" >> $OUTPUT_FILE  
18.         fi  
19.     done < $INPUT_FILE  
20.
```



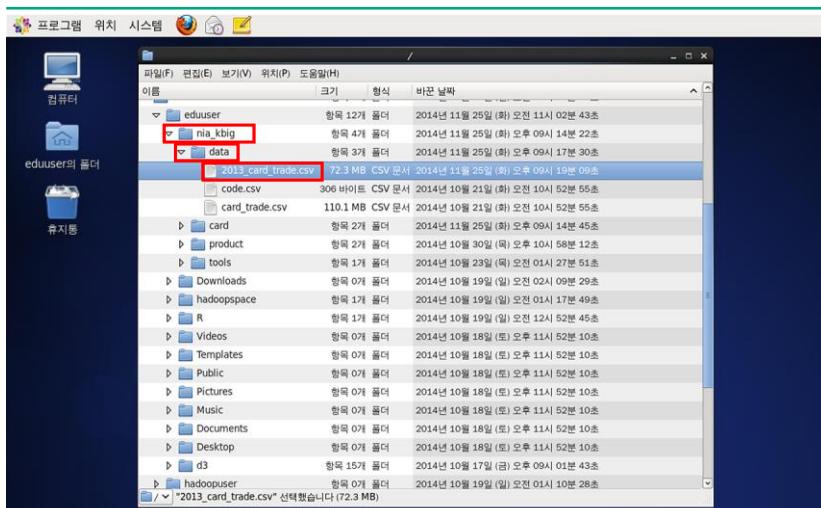
- 24페이지 데이터 가공 스크립트 소스(02.proc_csv.sh)
- 라인 17~19 : 2013년도 소비가계부 데이터에 해당되는 데이터를 파일로 저장한다.

▶ 원시데이터에서 분석 대상 데이터 가공



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

▶ 가공 데이터 작업 영역 폴더에 생성



- /home/eduuser/nia_kbig/data 폴더에 2013_card_trade.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 데이터를 MongoDB에 입력한다. MongoDB에서 저장된 데이터를 가지고 분석을 지원하는 맵리듀스 기능을 제공한다.

> 저장 방법

- 2013년도 가계부 데이터파일(2013_card_trade.csv)을 분석하기 위해 MongoDB에 Import 처리한다.
- MongoDB에서 제공하는 CSV Import 툴인 mongoimport을 사용하여 MongoDB에 원시데이터를 Import 한다.
- MongoDB에 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 import 할 CSV 파일을 mongoimport 커맨드로 저장 처리를 한다.

03.mongo_data_store.sh (가공데이터를 MongoDB에 저장)

```

01.#!/bin/bash
02. # Import 파일 위치 경로
03.LOCAL_TARGET=/home/eduuser/nia_kbig/data/2013_card_trade.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=card
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 -he
17.             ↛ derline

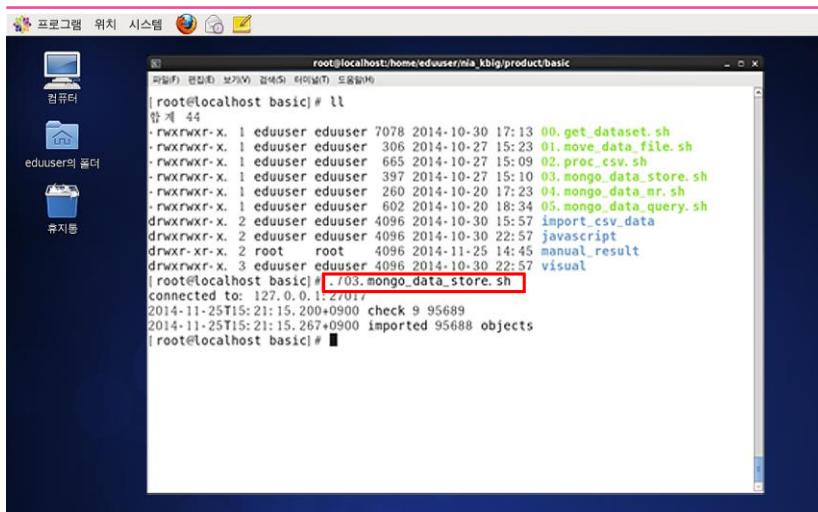
```



- 가공 데이터 저장 스크립트 소스(03.mongo_data_store.sh)
- 라인 03 : MongoDB에 입력할 가공데이터(2013_card_trade.csv)를 지정하는 라인이다.
- 라인 05~06 : 로컬 서버에 있는 MongoDB에 접속을 설정하는 라인이다.
- 라인 08~12 : bigdata 데이터베이스를 정의하고 컬렉션으로 card를 지정한다. MongoDB에 입력되는 데이터 파일 형식이 csv 파일로 지정하는 라인이다.
- 라인 14~16 : mongoimport 명령어에 접속호스트, 포트, 데이터베이스, 컬렉션, 타입을 지정하고, 가공된 데이터(2013_card_trade.csv)를 MongoDB에 import 처리하는 라인이다.

IV. 저장

▶ 가공 데이터 MongoDB에 저장

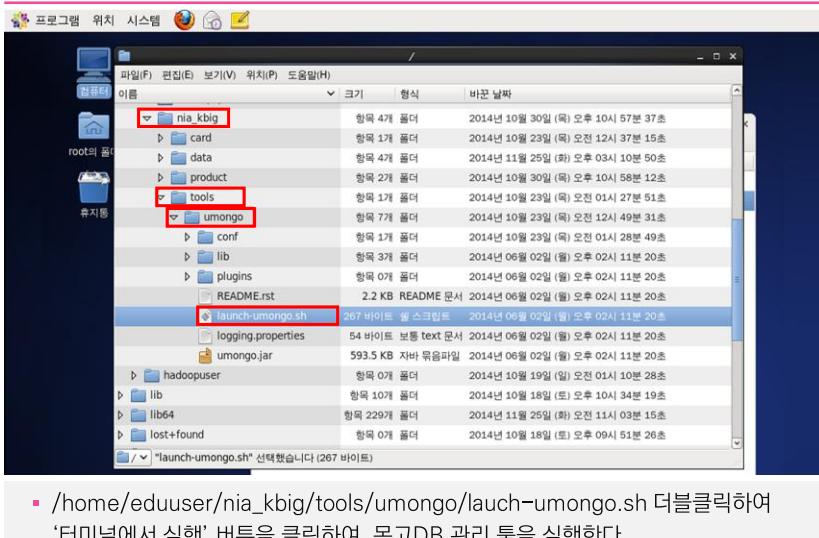


```
root@localhost basic]# ll
합계 44
-rwxrwxr-x. 1 eduuser eduuser 7078 2014-10-30 17:13 00.get_dataset.sh
-rwxrwxr-x. 1 eduuser eduuser 306 2014-10-27 15:23 01.move_data_file.sh
-rwxrwxr-x. 1 eduuser eduuser 665 2014-10-27 15:09 02.proc_csv.sh
-rwxrwxr-x. 1 eduuser eduuser 397 2014-10-27 15:10 03.mongo_data_store.sh
-rwxrwxr-x. 1 eduuser eduuser 260 2014-10-20 17:23 04.mongo_data_mr.sh
-rwxrwxr-x. 1 eduuser eduuser 602 2014-10-20 18:34 05.mongo_data_query.sh
drwxrwxr-x. 2 eduuser eduuser 4096 2014-10-30 15:57 import_csv_data
drwxrwxr-x. 2 eduuser eduuser 4096 2014-10-30 22:57 javascript
drwxr-xr-x. 2 root root 4096 2014-11-25 14:45 manual_result
drwxrwxr-x. 3 eduuser eduuser 4096 2014-10-30 22:57 visual
root@localhost basic]# ./03.mongo_data_store.sh
connected to: 127.0.0.1:27017
2014-11-25T15:21:15.200+0900 check 9 95689
2014-11-25T15:21:15.267+0900 imported 95688 objects
root@localhost basic]#
```

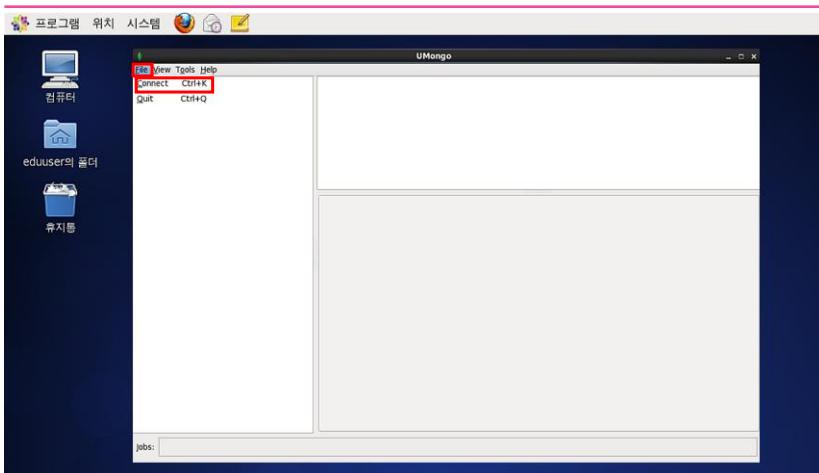
- 원시 데이터 셋을 분석하기 위해 MongoDB에 원시데이터를 저장한다.
./03.mongo_data_store.sh 를 입력 후 엔터

> MongoDB 저장 데이터 조회

> umongo 툴 실행

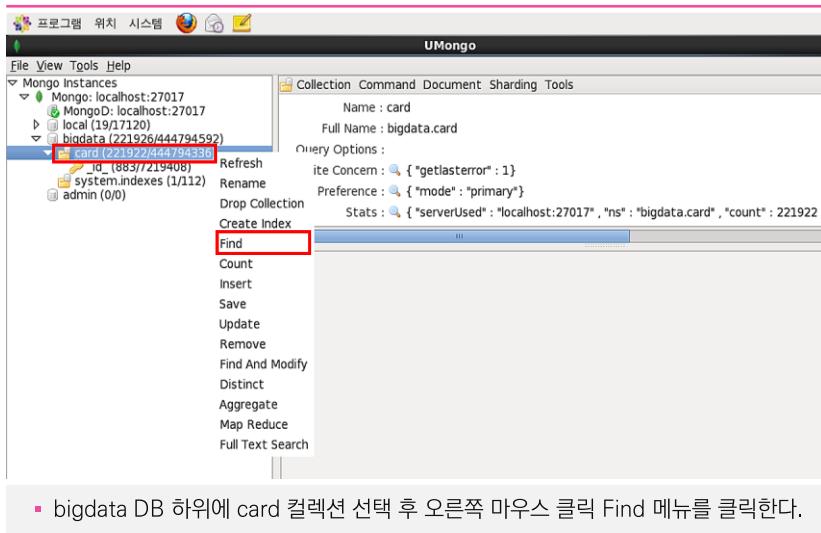


> MongoDB 접속

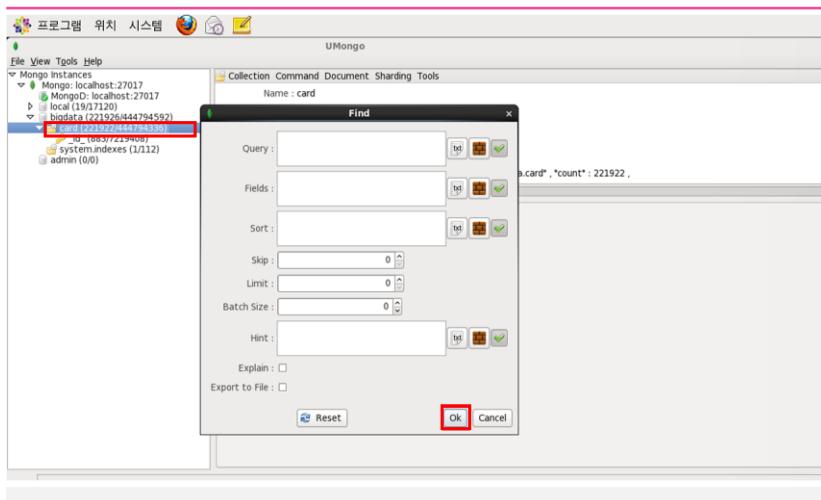


IV. 저장

▶ card 컬렉션 선택



▶ card 컬렉션 Find



> card 컬렉션 목록 선택

The screenshot shows the UMongo interface with the following details:

- Mongo Instances:** Mongo: localhost:27017, MongoD: localhost:27017, local(192.168.0.10:27017), bigdata(27017/44474592), _id (86377219408)
- Collection:** card
- Document:** bigdata.card
- Sharding:** None
- Tools:** None
- Name:** card
- Full Name:** bigdata.card
- Query Options:** Write Concern: { "getlasterror": 1}, Read Preference: { "mode": "primary"}
- Stats:** { "serverUsed": "localhost:27017", "ns": "bigdata.card", "count": 221922, "size": 1032000000}
- Result:** View Tools [td] [p]
- Document Details:**

```
{
  "_id": {
    "$oid": "547520028be79051deb8ccf7"
  },
  "u_trade_no": "20131211152750101587",
  "u_user_id": {
    "$oid": "547520028be79051deb8cc00"
  },
  "u_trade_no": "20131211152750101588",
  "u_u_id": {
    "$oid": "547520028be79051deb8cc01"
  },
  "u_trade_no": "20131211152750101589",
  "u_u_id": {
    "$oid": "547520028be79051deb8cc17"
  },
  "u_trade_no": "20131212091610101977"
}
{
  "_id": {
    "$oid": "547520028be79051deb8ccf7"
  },
  "u_trade_no": "20131212091610101977",
  "user_id": "20131212091604253233",
  "category_type": "C29999",
  "corp_id": "",
  "sms_receive_dt": 20131018154258,
  "pay_type": "PA12",
  "trade_type": "T103",
  "pay_cd": "BC140",
  "pay_account": "*****",
  "trade_site_nm": "",
  "trade_site_id": "",
  "trade_invoiceno": "1603100"
}
```

- bigdata MongoDB의 card 컬렉션에 입력된 데이터 목록이 출력이 되고 목록을 클릭하면 데이터셋 내용을 확인할 수 있다.

W





V 분석

개요	37
데이터 분석 스크립트	39
가공 데이터 분석 맵리듀스 실행	43
분석 후 결과 데이터	44

V 분석

> 개요

소비 데이터의 분석은 몽고DB에 입력된 가공 데이터에서 일자별 소비 지출 비용을 카테고리 분야별로 분류를 하고 분류된 데이터를 일자별로 합계를 계산하여 일자, 카테고리 코드, 지출 합계 항목을 몽고DB에 데이터를 입력 한 후, 2013년도 사용자의 분야별 지출 소비 패턴을 분석하기 위해 시계열 분석 방법을 적용한다.

> 분석 방법

- 사용자 식별코드가 “20140209004745257016”에 해당하는 2013년도 지출 데이터 중 3개 분야(마트/쇼핑, 외식/부식, 커피/간식)를 그룹화하여 일자별, 분야별 평균 지출 데이터를 추출한다.
 - 시계열 분석의 패턴 분석에 용이한 형태로 맵리듀스 작업을 통해서 지출 분야별, 일자 별 합산된 데이터를 몽고DB에 ‘card_mr_result’ 컬렉션에 저장을 한다.
- 일자 별, 분야 별 지출 정보의 패턴을 분석하기 위해 시계열 분석을 사용한다.

> 가공 데이터 샘플

거래정보	사용자식별 코드	카테고리	결제코드	사용처	지출비용	거래일시
20131210175 053101518	2013121017 5053253232	CZ9999	BC140	밝은세상이비	57,900	2013103 1173000
20131210175 053101519	2013121017 5053253232	CA9999	BC141	(주)농협유통 양재하나로클	215,510	2013102 6185900
20131210175 054101520	2013121017 5053253232	CA9999	BC141	임광마트	11,700	2013102 4195600
20131210175 054101521	2013121017 5053253232	CB9999	BC141	김경자소문난 대구왕뽈찜	48,000	2013102 4194700
20131210175 054101522	2013121017 5053253232	CA9999	BC141	씨엔에스유통(주) 삼익주유	49,000	2013102 3065700

➤ 데이터 분석 스크립트(04.mongo_data_mr.sh)

➤ 가공 데이터 분석 실행 스크립트

- MongoDB는 맵리듀스 스크립트를 자바스크립트로 처리하여 자바스크립트 파일로 작성한 후 mongo 커맨드로 실행시켜 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.  
07. # 소비 가계부 지출 평균 M/R 모듈  
08. EXECUTE_JS_MODULE=javascript/card_avg_price.js  
09.  
10. mongo --host $MONGO_HOST --port $MONGO_PORT $MONGO_DATABASE  
    ↛ SE $EXECUTE_JS_MODULE  
11.
```



- 가공데이터 분석 맵리듀스 실행 스크립트 소스(04.mongo_data_mr.sh)
- 라인 03~05 : 로컬 서버에 있는 MongoDB에 접속정보를 설정하는 라인이다.
- 라인 08 : javascript 폴더에 있는 분석스크립트 파일(card_avg_price.js)을 맵리듀스 분석 파일로 지정하는 라인이다.
- 라인 10 : mongo 명령어에 접속호스트, 포트, 데이터베이스, 분석스크립트파일을 지정하고 가공된 데이터를 MongoDB에서 맵리듀스 분석 작업을 수행하는 라인이다.

➤ 맵리듀스 분석 스크립트(card_avg_price.js)

```
01. /**
02. * [card] 콜렉션에서 통계대상 (user_sid : "20140209004745257016")이 사용한 내역을
03. * '카테고리'별로 일자별 총금액을 구한다.
04. */
05.
06. // MapReduce 대상 콜렉션을 설정한다.
07. var card = db.card;
08.
09. // Map함수를 선언한다.
10. var map = function() {
11.   // 통계대상 "20140209004745257016" 이 아닌 다른이의 데이터는 처리하지
12.   ↪ 않는다.
13.   var user_sid = this.user_sid;
14.   if (user_sid != "20140209004745257016"){
15.     return;
16.   }
17.   var date = this.trade_dt + "";
18.   // 일자검증(테스트 데이터가 있다면 제거한다)
19.   if (date.length < 8) {
20.     return;
21.   }
22.   // '일자(YYYYMMDD):카테고리코드의앞2자'를 키로 만든다.
23.   date = date.substring(0, 8); // YYYYMMDD
24.   // 2013년 데이터가 아니면 버린다.
25.   if (date < "20130101" || date >= "20140101") {
26.     return;
27.   }
28.
```



- 맵리듀스 분석 스크립트 소스(card_avg_price.js)
- 라인 07 : 몽고DB의 card 컬렉션을 설정하는 라인이다.
- 라인 10 : map 함수를 선언하는 라인이다.
- 라인 12~15 : 사용자 아이디가 '20140209004745257016'인 사용자의 데이터만을 설정하는 라인이다.

I. 개요

II. 수집

III. 가공

IV. 저장

V. 분석

VI. 시각화

V. 분석

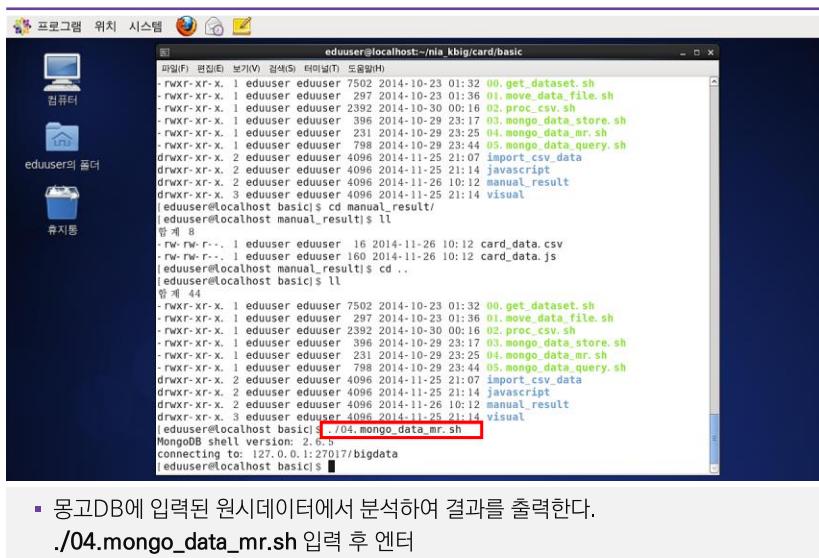
```
29.     var kind = this.category_type.substring(0,2);
30.     var key = date + ":" + kind;
31.     //console.log(key);
32.     if ( this.trade_money == null || this.trade_money == "") {
33.         return;
34.     }
35.     var curDoc = new Object;
36.     curDoc.date = date;
37.     curDoc.kind = kind;
38.     curDoc.totalPrice = this.trade_money;
39.
40.     emit(key, curDoc);
41. };
42.
43. // Reduce함수를 선언한다.
44. var reduce = function(key, products) {
45.     // map에서 생성한 도큐먼트와 동일한 JSON 형태로 만들어주어야 한다.
46.     var reduced = {date: "", kind:"", totalPrice : 0 };
47.     // 동일한 키를 가지는 아이템들을 일순하면서 가격의 총합을 구한다.
48.     products.forEach(function(product) {
49.         reduced.date = product.date;
50.         reduced.kind = product.kind;
51.         reduced.totalPrice += product.totalPrice;
52.     });
53.     return reduced;
54. };
55.
56. // 콜렉션에 M/R 작업을 건다.
57. card.mapReduce(
58.     map,
59.     reduce,
60.     { out: 'card_mr_result'}, // out에 지정한 'card_mr_result' 콜렉션에 결과
61.     // 데이터가 저장된다.
62.     function(err, coll) {
63.         coll.find().toArray(function(err, arr) {
64.             console.log(arr);
```



- 40페이지 맵리듀스 분석 스크립트 소스(card_avg_price.js)
- 라인 29~30 : 카테고리 앞에 2자리를 단어를 자른다. 일자+카테고리 단어를 합쳐서 key 값을 생성하는 라인이다.
- 라인 35~41 : curDoc 객체를 생성하고 일자, 카테고리 2자리 코드, 총합계금액을 저장하는 라인이다.
- 라인 44~54 : map에서 생성한 객체와 동일한 구조의 json 형태를 만들어고 동일한 key 값을 가지고 있는 객체에 대해서 루프 돌면서 일자, 카테고리 코드, 총합계 금액을 저장하는 라인이다.
- 라인 57~65 : 맵리듀스 분석 작업을 수행하는 라인이다.

> 가공 데이터 분석 맵리듀스 실행

> 분석 맵리듀스 실행



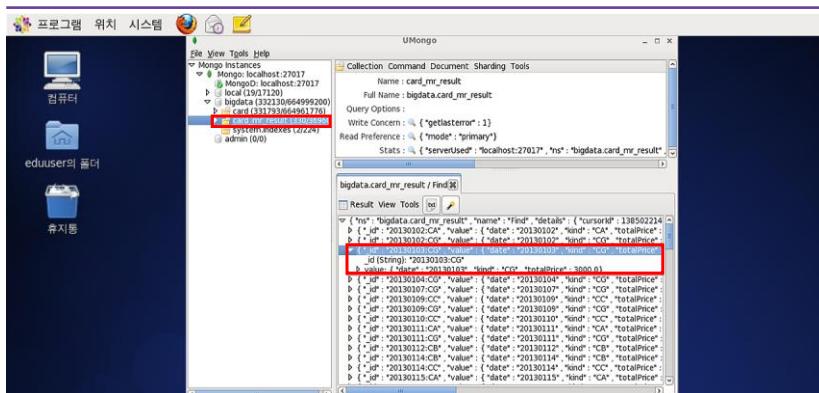
```

eduser@localhost basic]$ cd manual_result/
[eduser@localhost manual_result]$ ll
总 计 8
-rw-r--r-- 1 eduser eduser 16 2014-11-26 10:12 card_data.csv
-rw-r--r-- 1 eduser eduser 16 2014-11-26 10:12 card_data.js
[eduser@localhost manual_result]$ cd ..
[eduser@localhost basic]$ ll
总 计 44
-rw-r--r-- 1 eduser eduser 7502 2014-10-23 01:32 00.get_dataset.sh
-rw-r--r-- 1 eduser eduser 297 2014-10-23 01:36 01.move_data_file.sh
-rw-r--r-- 1 eduser eduser 2392 2014-10-30 00:16 02.proc_csv.sh
-rw-r--r-- 1 eduser eduser 396 2014-10-29 23:17 03.mongo_data_store.sh
-rw-r--r-- 1 eduser eduser 231 2014-10-29 23:25 04.mongo_data_mr.sh
-rw-r--r-- 1 eduser eduser 798 2014-10-29 23:44 05.mongo_data_query.sh
drwxr-xr-x 2 eduser eduser 4096 2014-11-25 21:07 import_csv_data
drwxr-xr-x 2 eduser eduser 4096 2014-11-25 21:14 javascript
drwxr-xr-x 2 eduser eduser 4096 2014-11-26 10:12 manual_result
drwxr-xr-x 3 eduser eduser 4096 2014-11-25 21:14 visual
[eduser@localhost basic]$ ls
MongoDB shell version: 2.6.0
connecting to: 127.0.0.1:27017/bigdata
[eduser@localhost basic]$ ./04.mongo_data_mr.sh

```

▪ MongoDB에 입력된 원시데이터에서 분석하여 결과를 출력한다.
./04.mongo_data_mr.sh 입력 후 엔터

> 분석 결과 데이터 확인



▪ digdata DB에서 오른쪽 마우스 클릭 후 Refresh 선택하면 분석 결과를 저장한 card_mr_result 컬렉션이 생성되어 있다.

▪ card_mr_result 컬렉션 선택 후 마우스 오른쪽 클릭 후 find 팝업 메뉴를 클릭하면 입력된 데이터 리스트를 확인할 수 있다.

▶ 분석 후 결과 데이터

▶ product_mr_result 데이터셋

```

01. [
02. {
03.   "kind": "마트/쇼핑",
04.   "Data": [
05.     {
06.       "Date": "20130102",
07.       "Value": 72710
08.     }
09.     ,
10.     {
11.       "Date": "20130111",
12.       "Value": 88760
13.     }
14.     ,
15.     {
16.       "Date": "20130115",
17.       "Value": 125000
18.     }
19.     ,
20.     {
21.       "Date": "20130116",
22.       "Value": 30700
23.     }
24.     ,
25.     {
26.       "Date": "20130125",
27.       "Value": 17500
28.     }
29.     ,
30.     {
31.       "Date": "20130127",
32.       "Value": 64540
33.     }
34.     ,
35.     {
36.       "Date": "20130128",
37.       "Value": 45000
38.     }
39.     ,
40.     {
41.       "Date": "20130129",
42.       "Value": 90000
43.     }
44.     ...[]
45.   ]
46. ];
47.
```



1

2



> 개요

몽고DB에 분석 저장된 데이터를 시각화하기 위해서 csv, json 형태의 파일로 결과 데이터를 저장해야 한다. csv형태 파일은 오픈오피스 스프레드시트에서 불러들여 차트를 생성할 수 있으며, json 형태의 파일은 D3 차트에서 시각화 할 때 사용되는 데이터 형식의 파일이다. 소비 가계부 데이터에서 분석된 데이터 중 3개 분야(마트/쇼핑, 외식/부식, 커피/간식)에 대해서 사용자의 소비 지출 정보를 분야별, 일자별 합계 금액을 꺾은선 그래프를 출력하여 2013년도의 소비 지출 패턴을 시각화 하여 검증해 본다.

> 시각화 방법 및 활용기술

- 2013년 3개 분야(마트/쇼핑, 외식/부식, 커피/간식)에 대한 지출 변화 추이 차트를 만들기 위해서 ‘card_mr_result’ 콜렉션에서 시각화에 사용할 데이터를 추출한다.
- 가공 변환된 데이터를 JSON 파일과 엑셀 CSV로 저장한다.
- 카테고리별 지출 변화 추이를 비교 분석하기 위해서 D3 Chart의 꺾은선 그래프를 활용한다.

▶ 데이터 변환



분석데이터 저장
스크립트 실행
(05.mongo_data_query.sh)



▶ 분석 데이터 저장 스크립트(05.mongo_data_query.sh)

▶ MongoDB 분석 결과 데이터 저장

- MongoDB에 저장된 분석 데이터를 파일 형식으로 저장한다.
- 터미널 커맨드 창에서 `./05.mongo_data_query.sh` 입력 후 엔터

05.mongo_data_query.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/card/basic/manual_result/card_
    ↵_data.js
06.OUTPUT_CSV_FILE=/home/eduuser/nia_kbig/card/basic/manual_result/card_
    ↵_data.csv
07.# 3개 분야(마트/쇼핑, 외식/부식, 커피/간식) M/R 결과 컬렉션에서 데이터 조회 모듈
08.EXECUTE_JS_MODULE=javascript/card_mr_result_query.js
09.
10.# M/R 결과 컬렉션에서 데이터 결과 출력 (JSON format)
11.mongo --quiet --host $MONGO_HOST --port $MONGO_PORT --eval "va
    ↵r param='json'" $MONGO_DATABASE $EXECUTE_JS_MODULE > $OUTP
        UT_JS_FILE
12.
13.# M/R 결과 컬렉션에서 데이터 결과 출력 (CSV format)
14.mongo --quiet --host $MONGO_HOST --port $MONGO_PORT --eval "va
    ↵r param='csv'" $MONGO_DATABASE $EXECUTE_JS_MODULE > $OUTP
        UT_CSV_FILE
```



- 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 지정을 한다. card_data.js, card_data.csv 2가지 형태 파일로 설정을 하는 라인이다.
- 라인 08 : 맵리듀스를 실행하는 스크립트(card_mr_result_query.js)를 지정하는 라인이다.
- 라인 11 : mongo 명령어를 사용하여 호스트, 포트, 데이터형식, 데이터베이스, 저장스크립트 설정하고 json 파일을 지정하여 데이터를 저장하는 라인이다.
- 라인 14 : mongo 명령어를 사용하여 호스트, 포트, 데이터형식, 데이터베이스, 저장스크립트 설정하고 csv 파일을 지정하여 데이터를 저장하는 라인이다.

I. 개요

II. 수집

III. 가공

IV. 저장

V. 분석

VI. 시각화

▶ 분석 결과 출력 스크립트

Javascript/card_mr_result_query.js

```

01. // 대상 콜렉션을 선정한다.
02. var card_mr_result = db.card_mr_result;
03.
04. function printResultJSONArray(targetKind , targetName) {
05.     // kind명이 'targetKind'인 종 문서의 갯수를 구한다
06.     var totalCount = card_mr_result.find({ "value.kind" : targetKind }).count();
07.     if (totalCount == 0) {
08.         print('f');
09.         print("kind" : "" + targetName + " ,");
10.         print('Data' : [] );
11.         print(')');
12.         return;
13.     }
14.     var startIdx = 0;
15.
16.     // kind명이 'targetKind'인 문서를 조회한다.
17.     card_mr_result.find({ "value.kind" : targetKind }).forEach(get_results);
18.     // 조회결과를 callback처리하여 JSON 배열 포맷을 맞춘다.
19.     function get_results (result) {
20.         if (startIdx == 0) {
21.             print('f');
22.             print("kind" : "" + targetName + " ,");
23.             print("Data" : [ );
24.         }
25.         var date = result.value.date;
26.         var totalPrice = result.value.totalPrice;
27.         print(' { ');
28.         print("Date" : "" + date + " ,");
29.         print("Value" : ' + totalPrice );
30.         print(' } ');
31.         if (startIdx < totalCount -1) {
32.             print(",");
33.         } else {
34.             print(']');
35.             print(')');
36.         }
37.         startIdx++;
38.     }

```

```

39.    }
40.
41.    function printResultCSV(targetKind , targetName) {
42.        // item에 따른 총 문서의 갯수를 구한다
43.        var totalCount = card_mr_result.find({ "value.kind" : targetKind }).count();
44.
45.        // item에 따른 문서를 조회한다.
46.        card_mr_result.find({ "value.kind" : targetKind }).forEach(get_results);
47.        // 조회결과를 callback처리하며 CSV 포맷을 맞춘다.
48.        function get_results (result) {
49.            var date = result.value.date;
50.            var totalPrice = result.value.totalPrice;
51.            print( targetName + "," + date + "," + totalPrice);
52.        }
53.    }
54.
55.    // 복수개의 부류명을 지정하여 결과 값을 JSON Array로 출력한다.
56.    function printAsJSON() {
57.        print("var data = ");
58.        print("[");
59.        printResultJSONArray("CA", "마트/쇼핑");
60.        print(",");
61.        printResultJSONArray("CB", "외식/부식");
62.        print(",");
63.        printResultJSONArray("CC", "커피/간식");
64.        /*print(",");
65.        printResultJSONArray("CD", "레저/문화");
66.        print(",");
67.        printResultJSONArray("CH", "주거/생활");*/

```



- 분석 결과 출력 스크립트 소스(javascript/card_mr_result_query.js)
- 라인 02 : 맵리듀스 분석결과를 저장할 컬렉션을 설정하는 라인이다.
- 라인 04 : json 결과물을 출력하는 함수를 선언하는 라인이다.
- 라인 06~14 : 총 개수가 없을 경우 카테고리명과 null data 정보를 출력하는 라인이다. 라인
라인 17~38 : 맵리듀스 결과 값을 루프 돌면서 일자, 종가격 정보를 지정하여 json 형태로
데이터를 출력하는 라인이다.
- 라인 57~70 : 카테고리 코드와, 카테고리 명을 넣어서 json 파일 형태로 데이터를 출력하는
라인이다.

I. 개요

II. 수집

III. 가공

IV. 저장

V. 분석

VI. 시각화

VI. 시각화

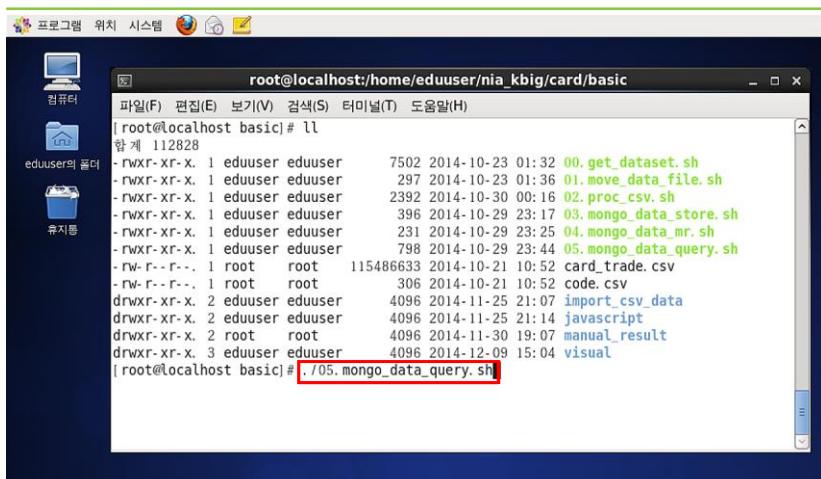
```
69.     print("]");  
70.     print(":");  
71. }  
72. // 복수개의 부류명을 지정하여 결과 값을 CSV로 출력한다.  
73. function printAsCSV() {  
74.     print("Kind,Date,Value");  
75.     printResultCSV("CA", "마트/쇼핑");  
76.     printResultCSV("CB", "외식/부식");  
77.     printResultCSV("CC", "커피/간식");  
78. }  
79. // mongo shell에서 parameter로 지정한 출력 포맷에 따라서 처리한다.  
80. if (param == 'json') {  
81.     printAsJSON();  
82. } else {  
83.     printAsCSV();  
84. }
```



- 분석 결과 출력 스크립트 소스(javascript/card_mr_result_query.js)
- 라인 73~77 : 카테고리 코드와, 카테고리 명을 넣어서 csv 파일 형태로 데이터를 출력하는 라인이다.
- 라인 80~83 : json 형태와 csv 형태에 따라 함수를 호출하는 라인이다.

▶ 분석 결과 파일 저장

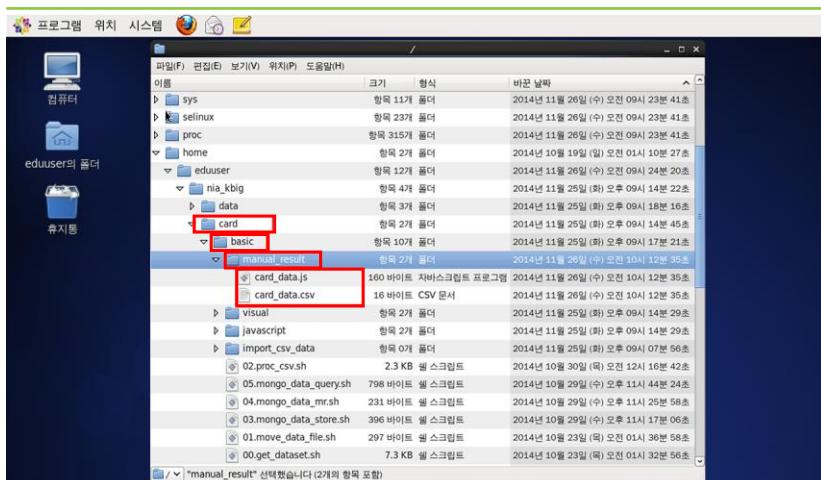
▶ 분석결과 데이터 저장 스크립트 실행



```
root@localhost:/home/eduuser/nia_kbig/card/basic
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
[ root@localhost basic ]# ll
합계 112828
eduuser의 폴더
- rwxr-xr-x. 1 eduuser eduuser 7502 2014-10-23 01:32 00.get_dataset.sh
- rwxr-xr-x. 1 eduuser eduuser 297 2014-10-23 01:36 01.move_data_file.sh
- rwxr-xr-x. 1 eduuser eduuser 2392 2014-10-30 00:06 02.proc_csv.sh
- rwxr-xr-x. 1 eduuser eduuser 396 2014-10-29 23:17 03.mongo_data_store.sh
- rwxr-xr-x. 1 eduuser eduuser 231 2014-10-29 23:25 04.mongo_data_mr.sh
- rwxr-xr-x. 1 eduuser eduuser 798 2014-10-29 23:44 05.mongo_data_query.sh
- rw-r--r--. 1 root root 115486633 2014-10-21 10:52 card_trade.csv
- rw-r--r--. 1 root root 306 2014-10-21 10:52 code.csv
drwxr-xr-x. 2 eduuser eduuser 4096 2014-11-25 21:07 import_csv_data
drwxr-xr-x. 2 eduuser eduuser 4096 2014-11-25 21:14 javascript
drwxr-xr-x. 2 root root 4096 2014-11-30 19:07 manual_result
drwxr-xr-x. 3 eduuser eduuser 4096 2014-12-09 15:04 visual
[ root@localhost basic ]# ./05.mongo_data_query.sh
```

./05.mongo_data_query.sh 입력 후 엔터

▶ 분석 결과 파일 저장 폴더



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

> 시각화 과정

> 시각화 절차



> 시각화 과정

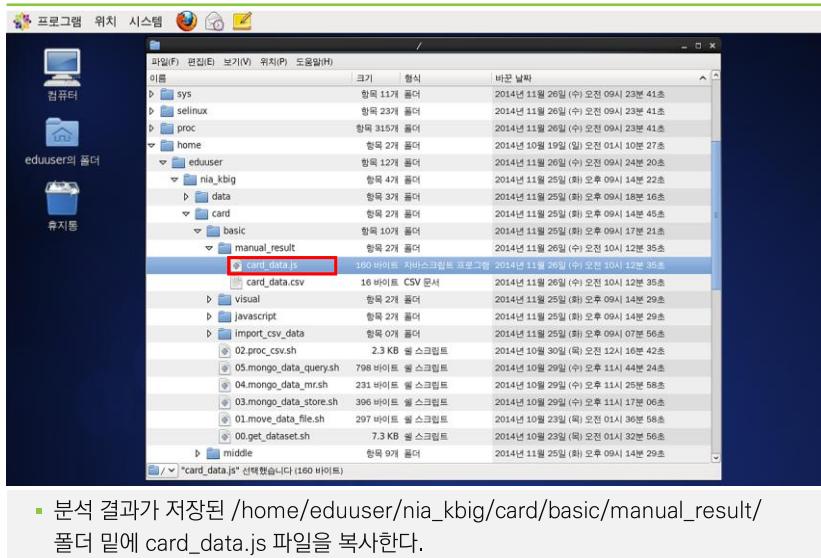
- d3.v3.js 라이브러리 파일을 제공 사이트에서 다운로드하여 저장한다.
- 시각화할 HTML 페이지를 생성한다. D3 Chart 라이브러리 모듈을 HTML 페이지에 삽입한다.
- D3 Chart Data를 읽어 오는 부분(product_data.js)에 결과 데이터를 삽입한다.
- X축과 Y축의 값을 지정한다.
- html 페이지를 웹브라우저에서 실행한다.

> D3 Chart 모듈 삽입과 결과 데이터 삽입

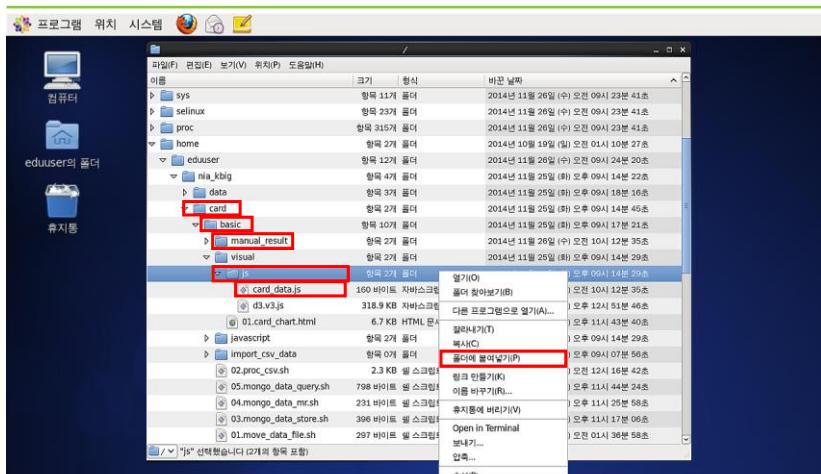
```
<!-- d3 모듈을 불러온다 -->
<script src="js/d3.v3.js"></script>
<!--d3 Chart data 연계 -->
<script src="js/card_data.js"></script>
```

> 분석 데이터 시작화

> 분석 데이터 파일 복사

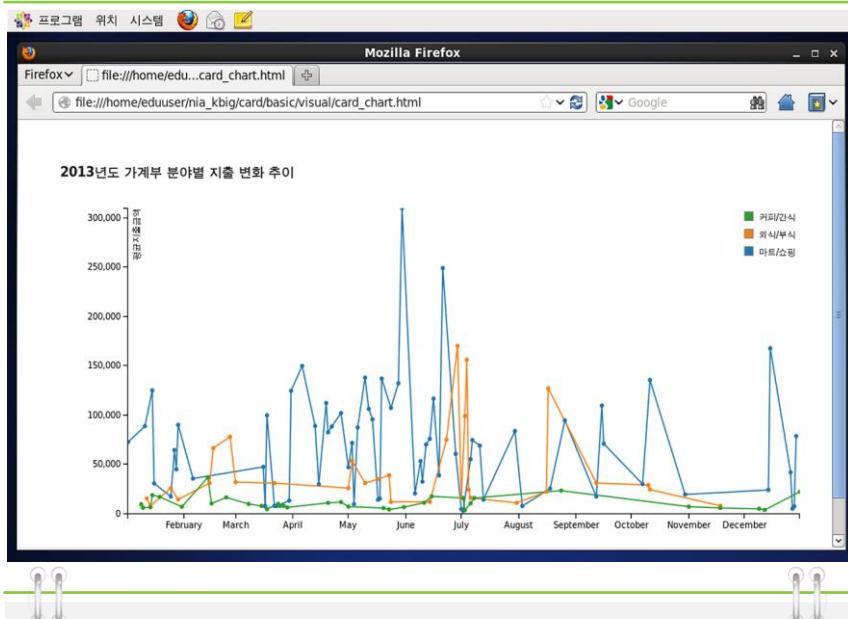


> 시작화 차트 실행



> 데이터 분석

> 2013년도 가계부 분야별 지출 변화 추이



- **마트/쇼핑** : 월별 마트/쇼핑 지출 비용이 평균 100,000 이상 지출하는 것으로 판단된다.
- 4월 ~ 8월 부분에 마트/ 쇼핑 지출 비용이 증가하고 패턴을 보인다.
- **외식/부식** : 7월, 8월에 외식이 증가하는 패턴을 보임. 2013년 7월에 초복과 중복이 있고, 8월 중순에 말복이 있다.
- 더위가 시작되는 시점과 복날 시점으로 외식이 증가한 것으로 판단된다.
- **커피/간식** : 월별 커피/간식 지출하는 비용이 평균적으로 비슷한 패턴을 보인다.
- 일상생활 시 정기적으로 지출하는 비용으로 판단된다.

I. 개요

II. 수집

III. 가공

IV. 저장

V. 분석

VI. 시각화



VII 예제문제

예제 문제1

61

예제 문제2

62

예 / 제 / 문 / 제

예제 1

가계 지출 내역을 통한 개인소비 성향을 분석하라.

- 가계지출 내역을 통하여 개인 사용자의 지출 비율을 구성하고 지출 비율에 따른 개인 소비 성향을 그룹으로 분류하여 비교하라.

- 소비 가계부 데이터 셋에서 2013년 사용자 10명을 추출한다.
- 사용자를 5명으로 구성하여 2개의 사용자 그룹으로 분류한다.
- 그룹별 2013년도 사용내역을 카테고리별로 분류를 하여 일별 사용 평균을 구한다.
- 그룹별, 카테고리별, 월별 지출 비용의 평균을 구한다.
- 그룹별 카테고리 평균 가격을 시각화하여 비교한다.

예제 2

카테고리별 소비 패턴을 분석하라.

- 연도별로 사용자들의 카테고리별 지출 패턴 분석하고 연도별, 카테고리별 지출 패턴을 시각화하라.

- 연도별, 카테고리별로 지출 가격의 평균을 구한다.
- 연도별로 지출 패턴을 카테고리별로 출력한다.
- 연도를 그룹화하여 카테고리를 지출 비용을 막대그래프로 시각화한다.



소비 

Intermediate Level

중급과정







I 개요

개요

67

I

개요

> 개요

개인별 소비지출 정보인 소비(가계부 정보) 데이터를 바탕으로 분야별 지출한 비용 중 2013년 데이터 정보를 추출하여 소비 패턴을 월별, 분야별로 그룹화하여 소비패턴을 2013년도의 월별 평균온도와 비교하여 온도에 따른 사용자의 소비 패턴을 알아보고자 한다. 분석에 사용할 데이터는 임의 사용자를 선정하여 2013년도 데이터만을 가공하여 사용하고, 2013년도 기상청 데이터를 활용한다. 두 개의 데이터를 매쉬업하고, 그 결과를 시계열 분석을 통해 2013년도 월별 온도 변화에 따른 사용자 소비 패턴을 파악하고자 한다.

> 활용 데이터

- **card_trade.csv** : 2013년 카드 사용 정보 데이터
- **code.csv** : 코드 정보 데이터
- **weather.csv** : 기상 데이터

> 선행학습

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

> 요구사항

- 분석에 사용되는 소비 가계부 데이터 셋은 2013년 데이터 중 마트/쇼핑, 외식/부식, 커피/간식 분야로 지출되는 비용을 그룹화하여 각 분야별 소비 패턴과 기상의 온도와의 연관성을 파악해 본다.

> 분석 절차

- 수집된 소비 가계부 정보 데이터와 기상청 데이터를 로드한다.
- 제공된 소비 가계부 정보에서 식별코드가 “20131210175053253232”에 해당하는 사용자의 2013년도의 지출 정보를 시계열 분석에 용이한 데이터 형태로 변화하기 위해 추출하여 가공된 데이터를 CSV 파일로 저장한다.
- 사용자의 월별, 분야별 지출 비용을 그룹화하여 월별 평균 지출 비용을 구한다.
- 기상 데이터는 서울지역 2013년 월별 평균기온을 추출하여 CSV 형태 파일로 저장한다.
- 추출한 2013년 소비 데이터와 2013년 월별 평균기온 데이터를 하둡 파일 시스템에 업로드한다.
- 월별, 분야별 평균 지출 비용을 분석하기 위해서 패턴분석을 위한 분석 스크립트 (맵리듀스)를 실행한다.
- 마트/쇼핑, 외식/부식, 커피/간식 분야의 분석 데이터를 하둡 분산 파일 시스템에 결과 파일을 CSV, JSON 형태로 저장한다.
- 분석된 데이터를 엑셀 형식이나 D3 차트 형식으로 보기 위한 데이터를 서버 로컬 폴더로 저장한다.
- 저장된 JSON 파일을 불러와서 D3 차트 중 꺾은선 그래프로 월별 지출 데이터와 월별 온도 데이터를 시각화하여 소비 패턴 분석한다.



II 수집

개요	71
교육용 데이터 샘플	72
데이터 수집	74
데이터 작업 영역 이동 스크립트	77



수집

> 개요

소비 가계부 데이터는 소비 데이터는 국내 XX 카드사에서 제공받은 2012~2013년 개인별 소비지출 정보인 소비(가계부 정보) 데이터를 수집하여 분석 목적을 달성할 수 있는 한도 내에서 2013년도 데이터를 추출하여 개인 정보, 카드 정보 등을 비식별화 처리를 통해 분석에 용이하게 편집하여 제공한다. 소비 가계부 정보 데이터는 개인이 카드 사용 시 지출되는 비용을 문자로 전송 받은 데이터를 스마트 폰 가계부 앱을 통해서 수집되는 데이터이다. 기상 데이터는 기상청 사이트의 날씨 > 기후자료 > 과거자료 메뉴에서 기상대 지점별 기상데이터를 확보하여 제공한다.

> 수집 방법

- **데이터 제공 :** 소비 가계부 정보 데이터는 국내 XX카드사에서 제공하는 데이터를 OpenAPI, 자료수집기(Crawler)를 통하여 데이터를 수집하였고, 실습용 자료는 빅데이터 분석 활용센터에 접속하여 유통 데이터 셋을 다운로드할 수 있도록 원시데이터를 제공하고 있다.
- **데이터 제공 :** 기상청 데이터
http://www.kma.go.kr/weather/climate/past_cal.jsp



용어정리

- **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

> 소비 가계부 정보 데이터(card_trade.csv)

거래정보	사용자식별 코드	카테 고리	문자수신 일자	결 제 타 입	거 래 타 입	결제 코드	사용처	지출 비용	거래 일시
20131210 17505310 1518	2013121017 5053253232	CZ9 999	2013111 4174310	PA 12	TT 01	BC1 40	밝은세상이비	57,900	201310 311730 00
20131210 17505310 1519	2013121017 5053253232	CA9 999	2013111 4174251	PA 03	TT 01	BC1 41	(주)농협유통양 재하나로클	215,510	201310 261859 00
20131210 17505410 1520	2013121017 5053253232	CA9 999	2013111 4174231	PA 03	TT 01	BC1 41	임광마트	11,700	201310 241956 00
20131210 17505410 1521	2013121017 5053253232	CB9 999	2013111 4174210	PA 03	TT 01	BC1 41	김경자소문난 대구왕뽈찜	48,000	201310 241947 00
20131210 17505410 1522	2013121017 5053253232	CA9 999	2013111 4174151	PA 03	TT 01	BC1 41	씨앤피스유통(주) 삼익주유	49,000	201310 230657 00
20131210 17505410 1523	2013121017 5053253232	CE0 011	2013111 4174129	PA 03	TT 01	BC1 41	메디팜효자약국	19,000	201310 211958 00
20131210 17505410 1524	2013121017 5053253232	CA0 081	2013111 4174113	PA 12	TT 01	BC1 40	씨유양재알뜰점	14,000	201310 211323 00
20131210 17505410 1525	2013121017 5053253232	CI9 999	2013111 4174047	PA 03	TT 01	BC1 41	솔대어린이집 [아이사랑]	253,000	201310 210728 00
20131210 17505410 1526	2013121017 5053253232	CF9 999	2013111 4174022	PA 03	TT 01	BC1 41	리헤어갤러리	10,000	201310 201835 00

> 기상 데이터(weather.csv)

지역	기상구분	측정값	일자
서울(청)	평균기온	-6.8	20110101
서울(청)	평균기온	-0.2	20110201
서울(청)	평균기온	0.5	20110301
서울(청)	평균기온	9.1	20110401
서울(청)	평균기온	12.5	20110501
서울(청)	평균기온	18	20110601
서울(청)	평균기온	25.1	20110701
서울(청)	평균기온	25.6	20110801
서울(청)	평균기온	27	20110901
서울(청)	평균기온	12.7	20111001

II. 수집

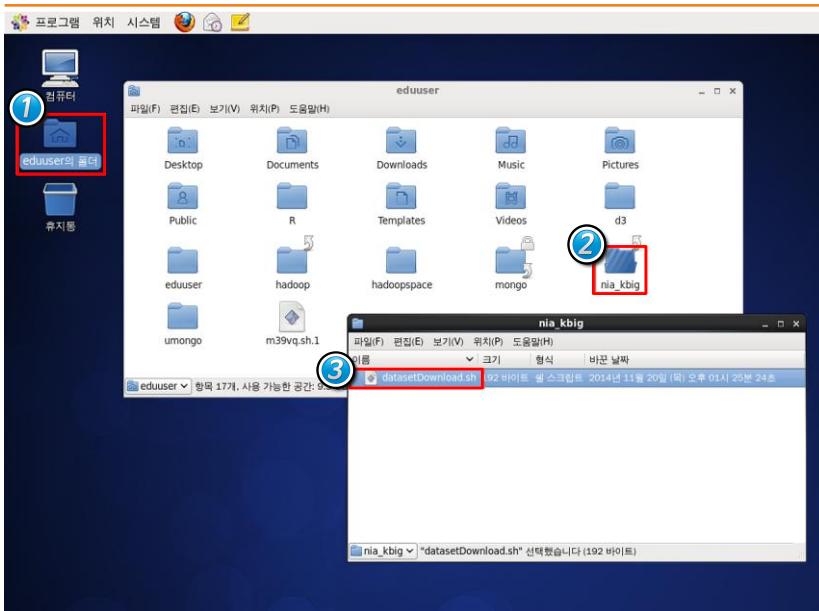
▶ 코드설명 데이터(code.csv)

코드	코드설명	코드	코드설명	코드	코드설명
CA0000	마트/쇼핑	CF0000	미용/뷰티	CK0000	외환/해외
CB0000	외식/부식	CG0000	교통/주유	CL0000	기타
CC0000	커피/간식	CH0000	주거/생활	CZ9999	미지정
CD0000	레저/문화	CI0000	교육/학원		
CE0000	건강/의료	CJ0000	보험/세금		

> 데이터 수집(datasetDownload.sh)

- 데이터 저장소에서 서버 로컬로 소비 데이터 셋을 복사해 온다.
 - card_trade.csv** : 2013년 카드 사용 정보 데이터
 - code.csv** : 코드 정보 데이터
 - weather.csv** : 기상 데이터

> 실습코드 디렉토리로 이동

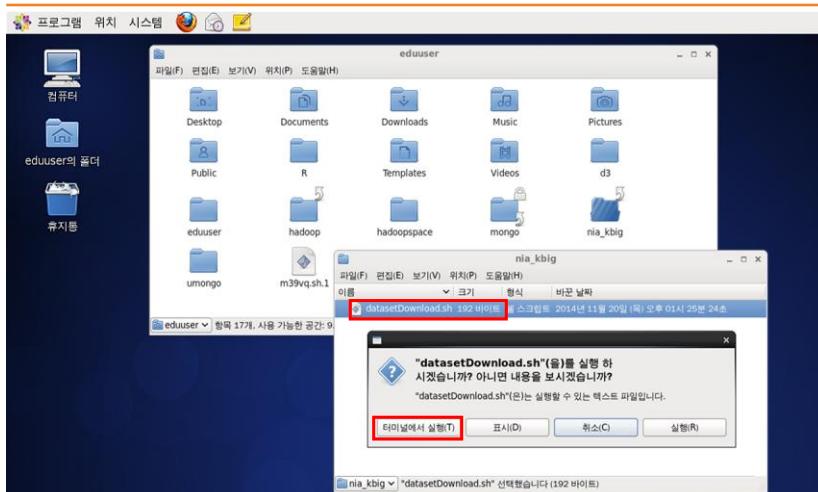


- 로그인 후 바탕화면에서 eduuser 폴더를 오픈한다.
- nia_kbig 폴더를 오픈한다.
- datasetDownload.sh를 더블클릭하여 실행한다.

II. 수집

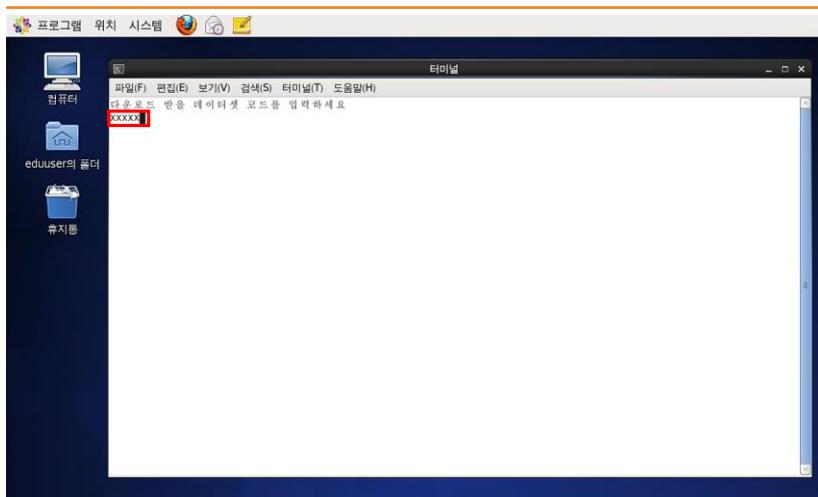
▶ 레파지토리에서 데이터 수집

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 : 월시데이터에서 분석할 대상을 추출하여 저장하는 스크립트
- 03.upload_csv.sh : 하둡파일시스템 가공데이터 파일 업로드 실행 스크립트
- 04.run.sh : 가공데이터 분석 맵리듀스 실행 스크립트
- wheather.csv : 기상데이터
- code.csv : 카테고리 분류 코드 데이터
- card_trade.csv : 소비 가계부 데이터

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. #소비 가계부 데이터
04. TARGET_CARD_TRADE=/home/eduuser/nia_kbig/card/middle/card_trade.csv
05. #카테고리 코드설명 데이터
06. TARGET_CODE=/home/eduuser/nia_kbig/card/middle/code.csv
07. #기상데이터
08. TARGET_WEATHER=/home/eduuser/nia_kbig/card/middle/weather.csv
09. # 작업영역 디렉토리 정의
10. LOCAL_DIR=/home/eduuser/nia_kbig/data/
11. # 대상 디렉토리로 이동
12. mv $TARGET_CARD_TRADE $LOCAL_DIR
13. mv $TARGET_CODE $LOCAL_DIR
14. mv $TARGET_WEATHER $LOCAL_DIR
```



- 데이터 작업 영역 이동 스크립트 소스(01.move_data_file.sh)
- 라인 04~08 : 다운로드한 원시데이터 card_trade.csv, code.csv, wheather.csv 파일을 설정하는 라인이다.
- 라인 10 : 작업 폴더를 설정하는 라인이다.
- 라인 12~14 : 작업 폴더로 다운로드한 원시데이터를 이동하는 라인이다.

▶ 수집 데이터 셋 작업 영역 폴더 이동

The screenshot shows a terminal window titled "root@localhost:/home/eduuser/nia_kbig/card/middle". The window displays a file listing with the command "ll". A red box highlights the command "./01.move_data_file.sh".

```
root@localhost middle]# ll
total 137172
drwxr-xr-x 1 eduuser eduuser 7290 2014-11-26 12:17 00.get_dataset.sh
drwxr-xr-x 1 eduuser eduuser 390 2014-11-07 14:55 01.move_data_file.sh
drwxr-xr-x 1 eduuser eduuser 3660 2014-11-02 22:05 02.proc_csv.sh
drwxr-xr-x 1 eduuser eduuser 671 2014-11-07 13:42 03.upload_csv.sh
drwxr-xr-x 1 eduuser eduuser 1639 2014-11-02 22:23 04.run.sh
drwxr-xr-x 1 eduuser eduuser 115486633 2014-10-21 10:52 card_trade.csv
drwxr-xr-x 2 eduuser eduuser 4096 2014-11-25 21:08 import_csv_data
drwxr-xr-x 3 eduuser eduuser 4096 2014-11-25 21:14 java_source
drwxr-xr-x 2 eduuser eduuser 4096 2014-11-26 14:57 manual_result
drwxr-xr-x 3 eduuser eduuser 4096 2014-11-25 21:14 visual
drwxr-xr-x 1 eduuser eduuser 24926973 2014-10-31 09:29 weather.csv
root@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



III 가공

개요

81

데이터 가공 스크립트

83



가공

> 개요

작업 영역 폴더에 복사한 소비 가계부 데이터(card_trade.csv)의 가공은 전처리 단계에서 수집된 데이터에서 분석할 대상의 범위와 분석 목표를 설정한 뒤 분석 대상자 1인을 선정하고 2013년도 기상 데이터를 추출하여 하둡 파일 시스템의 맵리듀스를 적용하여 시계열 분석의 패턴 분석에 유용한 객체 형태로 변환하도록 한다.

> 가공 방법

- 하둡 파일 시스템에 올라가 있는 2013_card_trade.csv를 맵리듀스 작업을 통해서 2013년도 월별로 3개 분야(마트/쇼핑, 외식/부식, 커피/간식)에 대한 지출 변화 추이 데이터를 뽑아낸다.
- 하둡 파일 시스템에 올라가 있는 기상 데이터(weather.csv)를 맵리듀스 작업을 통해서 2013년도 서울시의 월별 평균기온 데이터를 뽑아 card_trade.csv 파일에 저장을 한다.
- 결과값은 하둡 파일 시스템의 '/user/bigdata/card/out/2013' 경로에 파일로 출력하도록 한다.
- 지출 변화 추이 데이터는 JSON 배열로 출력하고 서울시 평균온도 데이터는 자바스크립트의 배열 형태로 출력한다.

▶ 데이터셋

-2013_card_trade.csv(2013년도 소비 가계부 데이터)

거래정보	사용자식별 코드	카테 고리	문자수신 일자	결 제 타 입	거 래 타 입	결제 코드	사용처	지출비용	거래 일시
2013121 0175053 101518	2013121017 5053253232	CZ99 99	2013111 4174310	PA 12	TT 01	BC140	밝은세상이비	57,900	20131 03117 3000
2013121 0175053 101519	2013121017 5053253232	CA9 999	2013111 4174251	PA 03	TT 01	BC141	(주)농협유통 양재하나로클	215,510	20131 02618 5900
2013121 0175054 101520	2013121017 5053253232	CA9 999	2013111 4174231	PA 03	TT 01	BC141	임광마트	11,700	20131 02419 5600
2013121 0175054 101521	2013121017 5053253232	CB9 999	2013111 4174210	PA 03	TT 01	BC141	김경자소문난 대구왕뽈찜	48,000	20131 02419 4700
2013121 0175054 101522	2013121017 5053253232	CA9 999	2013111 4174151	PA 03	TT 01	BC141	씨엔에스유통 (주)삼익주유	49,000	20131 02306 5700
2013121 0175054 101523	2013121017 5053253232	CE0 011	2013111 4174129	PA 03	TT 01	BC141	메디팜효자 약국	19,000	20131 02119 5800
2013121 0175054 101524	2013121017 5053253232	CA0 081	2013111 4174113	PA 12	TT 01	BC140	씨유양재 알뜰점	14,000	20131 02113 2300
2013121 0175054 101525	2013121017 5053253232	CI99 99	2013111 4174047	PA 03	TT 01	BC141	솔대어린이집 [아이사랑]	253,000	20131 02107 2800
2013121 0175054 101526	2013121017 5053253232	CF99 99	2013111 4174022	PA 03	TT 01	BC141	리헤어갤러리	10,000	20131 02018 3500
2013121 0175054 101527	2013121017 5053253232	CA0 040	2013111 4174001	PA 03	TT 01	BC141	대보유통(주)/ 화성하주유소	80,000	20131 01802 5000
2013121 0175054 101528	2013121017 5053253232	CA9 999	2013111 4173857	PA 03	TT 01	BC141	씨엔에스유통 (주)삼익주유	48,000	20131 10406 0700
2013121 0175054 101529	2013121017 5053253232	CZ99 99	2013111 4173830	PA 03	TT 01	BC141	OK25	4,600	20131 10209 0100
2013121 0175054 101530	2013121017 5053253232	CD9 999	2013111 4173808	PA 03	TT 01	BC141	점풀린파크 (복수원)	4,000	20131 10315 2700

III. 가공

> 데이터 가공 스크립트(02.proc_csv.sh)

- 셀스크립트를 이용하여 소비 가계부 데이터 (card_trade.csv) 파일에서 2013년도 데이터만 추출하여 2013_card_trade.csv 파일을 생성한다. 기상정보 데이터도 2013년도의 서울지역의 데이터만을 추출하여 2013_weather.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/card_trade.csv'
04. # 출력결과 CSV 파일 지정
05. OUTPUT_FILE='/home/eduuser/nia_kbig/data/2013_card_trade.csv'
06. # 기상데이터 입력 CSV 파일 지정
07. WEATHER_INPUT_FILE='/home/eduuser/nia_kbig/data/weather.csv'
08. # 2013년 서울지역 평균온도 출력결과 CSV파일 지정
09. WEATHER_OUTPUT_FILE='/home/eduuser/nia_kbig/data/2013_weather.csv'
10. # 2013년도 데이터만을 대상으로 설정
11. TARGET_YEAR='2013'
12. # 평균기온만을 대상으로 설정
13. TARGET_TYPE='평균기온'
14. # 서울지역의 온도를 대상으로 설정
15. TARGET_AREA='서울'
16. # 2013년 서울지역 평균기온 출력결과 CSV HEADER컬럼 출력
17. echo "Date,Temperature" > $WEATHER_OUTPUT_FILE
18. #'를 구분자로 해서 파일을 읽어들인다.
19. IFS=':'
20. while read AREA TYPE VALUE DATE
21. do
22. # 측정값이 빈 것은 SKIP처리 한다.
23. if [ -z $VALUE ]; then
24.     continue;
25. fi
26. #echo "$DATE , $TYPE , $AREA"
27. # TARGET_YEAR로 시작하는 년도인지 체크한다.
28. # TARGET_TYPE(평균기온)인지 체크한다.
29. # TARGET_AREA(서울)인지 체크한다.
30. if [ [ ( $DATE == ${TARGET_YEAR}* ) && ( $TYPE == ${TARGET_TYPE}* ) &&
31. ($AREA == ${TARGET_AREA}* ) ] ]; then
# 해당년도의 데이터만을 CSV로 출력한다.
```

```

32.         echo "$DATE,$VALUE" >> $WEATHER_OUTPUT_FILE
33.     fi
34. done < $WEATHER_INPUT_FILE
35.
36. # 2013년 가계부 출력결과 CSV HEADER컬럼 출력
37. echo "u_trade_no,user_sid,category_type,corp_id,sms_receive_dt,pay_type,trad
38. ↪ e_type,pay_cd,pay_account,trade_site_nm,trade_site_id,trade_money,trade_
39. ↪ dt,quota_month,balance_money,point_type,add_point,use_point,sms_org,c
40. ↪ ar_fill_yn,online_site_yn,online_site_id,company_card_yn,inarea_yn,foreign_
41. ↪ _amount,foregin_unit,auto_pay_seq,parse_seq,callback_num,category_nm,m
42. ↪ emo,pay_nm,result_parse,reg_dt,trade_stat,use_yn,reg_type,sms_send_id,
43. ↪ ref_u_trade_no,mod_dt,user_nm,check_money,card_add_money,notapply_ch
44. ↪ eck_yn,notapply_check_dt,app_yn,autopay_type,autopay_remaind_cnt,join_ty
45. ↪ pe,trade_yn" > $OUTPUT_FILE
46. # ','를 구분자로 해서 파일을 읽어들인다.
47. IFS=':'
48. while read U_TRADE_NO USER_SID CATEGORY_TYPE CORP_ID SMS_RECEIVE_
49. ↪ DT PAY_TYPE TRADE_TYPE PAY_CD PAY_ACCOUNT TRADE_SITE_NM TRAD
50. ↪ E_SITE_ID TRADE MONEY TRADE_DT QUOTA_MONTH BALANCE MONEY P
51. ↪ OINT_TYPE ADD_POINT USE_POINT SMS_ORG CAR_FILL_YN ONLINE_SITE_Y
52. ↪ N ONLINE_SITE_ID COMPANY_CARD_YN INAREA_YN FOREIGN_AMOUNT F
53. ↪ OREGIN_UNIT AUTO_PAY_SEQ PARSE_SEQ CALLBACK_NUM CATEGORY_N
54. ↪ M MEMO PAY_NM RESULT_PARSE REG_DT TRADE_STAT USE_YN REG_TYP
55. ↪ E SMS_SEND_ID REF_U_TRADE_NO MOD_DT USER_NM CHECK_MONEY CA
56. ↪ RD_ADD_MONEY NOTAPPLY_CHECK_YN NOTAPPLY_CHECK_DT APP_YN AU
57. ↪ TOPAY_TYPE AUTOPAY_REMAIND_CNT JOIN_TYPE TRADE_YN
58. do
59. # TRADE_DT가 TARGET_YEAR로 시작하는 년도인지 체크한다.
60. if [[ $TRADE_DT == ${TARGET_YEAR}* ]]; then
61.     # 해당년도의 데이터만을 CSV로 출력한다.

```



- 데이터 가공 스크립트 소스(02.proc_csv.sh)
- 라인 03~05 : 소비 원시데이터(card_trade.csv) 지정을 하고 가공 결과를 2013_card_trade.csv로 지정하는 라인이다.
- 라인 07~09 : 기상 원시데이터(weather.csv)를 지정을 하고 가공 결과물을 2013_weather.csv 파일로 지정하는 라인이다.
- 라인 10~15 : 가공 대상인 년도(2013), 기온(평균기온), 지역(서울)을 설정하는 라인이다.
- 라인 17~31 : 기상 파일을 읽어서 2013년도 서울지역 기상만을 추출하여 저장하는 라인이다.
- 라인 37 : 출력 파일의 헤더 정보를 출력하는 라인이다.
- 라인 39~47 : 소비 가계부 데이터를 읽어서 2013년도 소비 지출정보만 출력하여 저장하는 라인이다.

I. 개요

II. 수집

III. 가공

IV. 저장

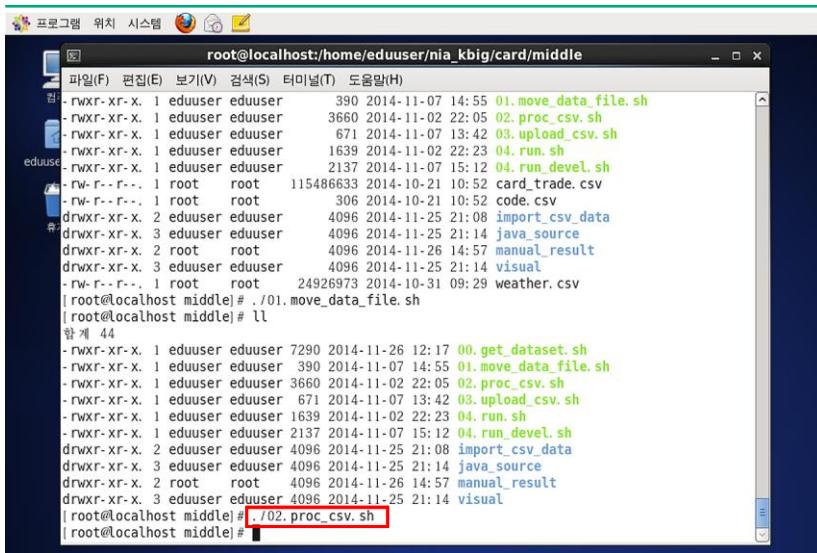
V. 분석

VI. 시각화

III. 가공

```
45. echo "$U_TRADE_NO,$USER_SID,$CATEGORY_TYPE,$CORP_ID,$SMS_
    ↵ RECEIVE_DT,$PAY_TYPE,$TRADE_TYPE,$PAY_CD,$PAY_ACCOUNT,
    $TRADE_SITE_NM,$TRADE_SITE_ID,$TRADE MONEY,$TRADE_DT,$
    QUOTA_MONTH,$BALANCE MONEY,$POINT_TYPE,$ADD_POINT,$U
    SE_POINT,$SMS_ORG,$CAR_FILL_YN,$ONLINE_SITE_YN,$ONLINE_S
    ITE_ID,$COMPANY_CARD_YN,$INAREA_YN,$FOREIGN_AMOUNT,$F
    OREGIN_UNIT,$AUTO_PAY_SEQ,$PARSE_SEQ,$CALLBACK_NUM,$C
    ATEGORY_NM,$MEMO,$PAY_NM,$RESULT_PARSE,$REG_DT,$TRAD
    E_STAT,$USE_YN,$REG_TYPE,$SMS_SEND_ID,$REF_U_TRADE_NO,
    $MOD_DT,$USER_NM,$CHECK_MONEY,$CARD_ADD_MONEY,$NOT
    APPLY_CHECK_YN,$NOTAPPLY_CHECK_DT,$APP_YN,$AUTOPAY_T
    YPE,$AUTOPAY_REMAIND_CNT,$JOIN_TYPE,$TRADE_YN" >> $OUTP
    UT_FILE
46.     fi
47. done < $INPUT_FILE
48.
```

> 원시데이터에서 분석 대상 데이터 가공



- 원시 데이터 셋에서 분석할 데이터를 공유하여 2013_card_trade.csv, 2013_weather.csv 파일을 생성한다. `./02.proc_csv.sh` 입력 후 엔터
 - 2013_card_trade.csv, 2013_weather.csv 2개 파일 생성한다.





IV 저장

개요	89
가공 데이터 하둡 파일시스템 업로드	90
가공 데이터 하둡 파일시스템 저장	91
하둡 파일 시스템 파일 조회	92
하둡 명령어로 파일 조회	93

IV

저장

> 개요

하둡 파일 시스템의 하둡 맵리듀스를 이용하기 위해 분석에 필요한 자료들을 하둡 파일 시스템에 업로드 시킨다. 분석에 사용될 데이터는 2013년도 가계부 데이터와 2013년도 서울 평균기온 데이터이다. 하둡의 put 명령어를 사용하여 하둡 파일시스템의 /user/bigdata/ 폴더로 자료를 업로드 한다.

> 저장 방법

- 2013년도 가계부 데이터 파일(2013_card_trade.csv)과 2013년도 서울 평균기온 데이터 파일(2013_weather.csv)을 하둡에 업로드한다.
- 하둡 커맨드를 이용해서 가공된 데이터를 하둡 파일 시스템에 업로드한다.

> 가공 데이터 하둡 파일시스템 업로드(03.upload_csv.sh)

> 하둡 파일시스템에 업로드 스크립트

- 원시데이터에서 가공된 2013_card_trade.csv, 2013_weather.csv 파일을 하둡시스템에 업로드한다.

03.upload_csv.sh (가공데이터를 하둡파일시스템으로 업로드)

```

01.#!/bin/bash
02. # 2013년 가계부 출력결과 CSV 파일 지정
03. CARD_OUTPUT_FILE='/home/eduuser/nia_kbig/data/2013_card_trade.csv'
04. # 2013년 서울지역 평균기온 출력결과 CSV파일 지정
05. WEATHER_OUTPUT_FILE='/home/eduuser/nia_kbig/data/2013_weather.csv'
06. # 하둡의 2013년 가계부 출력결과 저장 위치
07. HDFS_CARD=/user/bigdata/2013_card_trade.csv
08. # 하둡의 2013년 서울지역 평균기온 출력결과 저장 위치
09. HDFS_WEATHER=/user/bigdata/2013_weather.csv
10. # make directory on hadoop
11. hadoop fs -mkdir -p /user/bigdata
12. # upload target file to HDFS
13. hadoop fs -put $WEATHER_OUTPUT_FILE $HDFS_WEATHER
14. hadoop fs -put $CARD_OUTPUT_FILE $HDFS_CARD

```



- 가공 데이터 하둡 파일시스템 업로드 스크립트 소스(03.upload_csv.sh)
- 라인 03 : 가공된 데이터 파일(2013_card_trade.csv)을 설정하는 라인이다.
- 라인 05 : 2013년도 서울지역 평균기온(2013_weather.json)을 설정하는 라인이다.
- 라인 07 : 하둡 파일 시스템에 가공 소비가계부 데이터(2013_card_trade.csv) 저장 위치를 지정하는 라인이다.
- 라인 09 : 하둡 파일 시스템에 가공 기상 데이터(2013_weather.csv) 저장 위치를 지정하는 라인이다.
- 라인 11 : 하둡 파일 시스템 /user/bigdata/ 폴더를 생성하는 라인이다.
- 라인 13~14 : 하둡 파일 시스템에 가공 기상 데이터와 가공 소비가계부 데이터를 업로드하는 라인이다.

IV. 저장

▶ 가공 데이터 하둡 파일시스템 저장

> 가공 데이터 하둡 파일시스템 저장



```
eduuser@localhost:~/nia_kbig/card/middle
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
할계 40
-rwxr-xr-x. 1 eduuser eduuser 7290 2014-11-26 12:17 00.get_dataset.sh
-rwxr-xr-x. 1 eduuser eduuser 390 2014-11-07 14:55 01.move_data_file.sh
-rwxr-xr-x. 1 eduuser eduuser 3660 2014-11-02 22:05 02.proc_csv.sh
-rwxr-xr-x. 1 eduuser eduuser 671 2014-11-07 13:42 03.upload_csv.sh
-rwxr-xr-x. 1 eduuser eduuser 1639 2014-11-02 22:23 04.rum.sh
drwxr-xr-x. 2 eduuser eduuser 4096 2014-11-25 21:08 import_csv_data
drwxr-xr-x. 3 eduuser eduuser 4096 2014-11-25 21:14 java_source
drwxr-xr-x. 2 eduuser eduuser 4096 2014-11-25 14:57 manual_result
drwxr-xr-x. 3 eduuser eduuser 4096 2014-11-25 21:14 visual
[eduuser@localhost middle]$ ./03.upload_csv.sh
14/11/26 18:07:45 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your
platform.. using builtin-java classes where applicable
14/11/26 18:07:46 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your
platform.. using builtin-java classes where applicable
put: '/user/bigdata/2013.weather.csv': File exists
14/11/26 18:07:47 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your
platform.. using builtin-java classes where applicable
[eduuser@localhost middle]$
```

- 가공한 2013_card_trade.csv, 2013_weather.csv 파일을 하둡 파일시스템에 업로드한다. `./03.upload_csv.sh` 입력 후 엔터

▶ 하둡 파일 시스템 파일 조회

▶ 하둡 파일 시스템 접속

Contents of directory [/user/bigdata](#)

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
2013_card_trade.csv	file	72.31 MB	1	128 MB	2014-12-09 16:16	rw-r--r--	eduuser	supergroup
2013_weather.csv	file	9.68 KB	1	128 MB	2014-12-09 16:16	rw-r--r--	eduuser	supergroup

[Go back to DFS home](#)

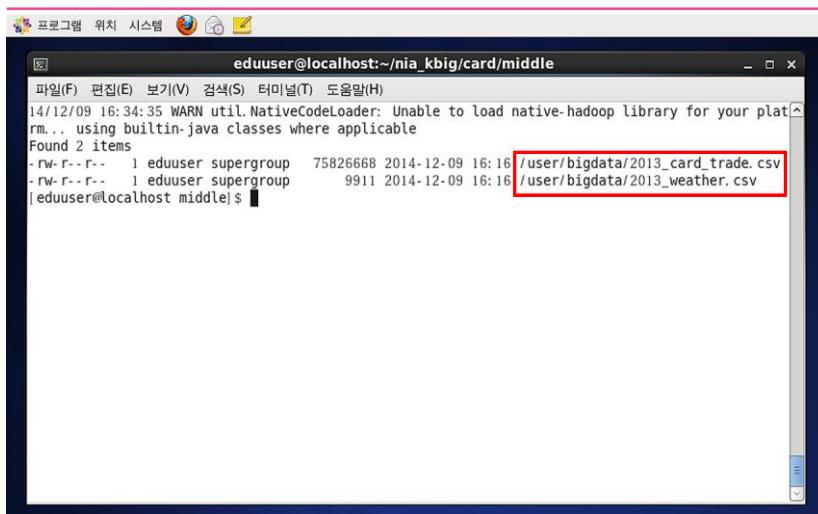
Local logs

[Log directory](#)
[Hadoop, 2014.](#)

- 파일어폭스 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 폴더 / bigdata 폴더를 클릭하면 업로드한 가공 데이터 목록을 볼 수 있다.

> 하둡 명령어로 파일 조회

> 하둡 파일시스템 조회



A screenshot of a terminal window titled "eduuser@localhost:~/nia_kbig/card/middle". The window shows the following command and its output:

```
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
14/12/09 16:34:35 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 75826668 2014-12-09 16:16 /user/bigdata/2013_card_trade.csv
- rw- r-- r-- 1 eduuser supergroup      9911 2014-12-09 16:16 /user/bigdata/2013_weather.csv
[eduuser@localhost middle]$
```

The last two lines of the output, which list the files "2013_card_trade.csv" and "2013_weather.csv", are highlighted with a red box.

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

I. 개요

II. 수집

III. 가공

IV. 저장

V. 분석

VI. 시각화

W





V 분석

개요	97
데이터 분석 스크립트	99
데이터 분석 맵리듀스 실행	101
분석 데이터 파일 조회	102
분석 후 결과 데이터	103

V 분석

> 개요

소비 데이터의 분석은 하둡 파일 시스템의 맵리듀스 함수를 활용하여 시계열 분석에 필요한 소비 지출 패턴 분석을 한다. 2013년도 소비 데이터의 월별, 분야별 평균 지출 비용과 2013년 서울 월별 평균기온을 비교 분석을 하여 온도 변화에 따른 사용자의 월별 분야별 소비 패턴 분석을 검증해 본다

> 분석 방법

- 자바로 하둡 파일 시스템에 올라가 있는 2013_card_trade.csv, 2013_weather.csv를 대상으로 맵리듀스 작업을 통해서 월별로 지출 비용 데이터를 뽑아낸다.
- 하둡 파일 시스템에 올라가 있는 2013년도 서울시 기상 데이터(2013_weather.csv)를 맵리듀스 작업을 통해서 2013년도 서울시의 일별 평균기온 데이터를 뽑아낸다.
- 결과값은 하둡 파일 시스템의 '/user/bigdata/card/out/2013' 경로에 파일로 출력하도록 한다.
- 맵리듀스를 실행하는 프로그램은 자바를 이용해서 구현하고 card.jar로 만들어서 실행한다.
- 콘솔에 로그인해서 실행은 하둡의 yarn 커맨드로 실행하고 결과 파일을 구한다.

> 가공 데이터 샘플

- 2013년 가계부 데이터

거래정보	사용자식별 코드	카테고리	문자수신일자	결제 타입	거래 타입	결제 코드	사용처	지출 비용	거래 일시
20131210 17505310 1518	2013121017 5053253232	CZ9999	20131 11417 4310	PA 12	TT 01	BC 140	밝은세상이비	57,900	2013 1031 1730 00
20131210 17505310 1519	2013121017 5053253232	CA9999	20131 11417 4251	PA 03	TT 01	BC 141	(주)농협유통 양재하나로클	215,510	2013 1026 1859 00
20131210 17505410 1520	2013121017 5053253232	CA9999	20131 11417 4231	PA 03	TT 01	BC 141	임광마트	11,700	2013 1024 1956 00
20131210 17505410 1521	2013121017 5053253232	CB9999	20131 11417 4210	PA 03	TT 01	BC 141	김경자소문난 대구왕뽈찜	48,000	2013 1024 1947 00
20131210 17505410 1522	2013121017 5053253232	CA9999	20131 11417 4151	PA 03	TT 01	BC 141	씨엔에스유통 (주)삼익주유	49,000	2013 1023 0657 00
20131210 17505410 1523	2013121017 5053253232	CE0011	20131 11417 4129	PA 03	TT 01	BC 141	메디팜효자약 국	19,000	2013 1021 1958 00
20131210 17505410 1524	2013121017 5053253232	CA0081	20131 11417 4113	PA 12	TT 01	BC 140	씨유양재 알뜰점	14,000	2013 1021 1323 00

- 2013년 서울지역 일별 평균기온

지역	기상구분	측정값	일자
서울(청)	평균기온	-4.7	20130101
서울(청)	평균기온	-11.7	20130102
서울(청)	평균기온	-13.2	20130103
서울(청)	평균기온	-10.7	20130104
서울(청)	평균기온	-7	20130105
서울(청)	평균기온	-6.3	20130106
서울(청)	평균기온	-5.1	20130107
서울(청)	평균기온	-4.6	20130108
서울(청)	평균기온	-9	20130109

➤ 데이터 분석 스크립트(04.run.sh)

➤ 가공 데이터 분석 실행 셸스크립트

- 맵리듀스를 처리하는 프로그램은 card.java에 구현되어 있다.
- 자바 프로그램을 컴파일하여 card.jar 파일로 만든 후 yarn 커맨드를 이용해서 card.jar 파일로 맵리듀스 작업을 수행한다.
- 분석 결과는 하둡 파일시스템의 지정한 디렉토리에 저장을 한다.

04.run.sh (맵리듀스 실행)

```

01.#!/bin/bash
02. # 현재 위치를 지정한다.
03. CURRENT_DIR=/home/eduuser/nia_kbig/card/middle
04. # 컴파일하여 생성할 프로그램(jar) 경로를 지정한다.
05. TARGET_JAR=$CURRENT_DIR/card.jar
06. # 소스파일 디렉토리를 지정한다.
07. TRAGET_SOURCE_DIR=$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.card
14. # Hadoop상에 존재하는 가계부 파일을 지정한다.
15. INPUT_CARD_DATA=/user/bigdata/2013_card.csv
16. # Hadoop상에 존재하는 기상(온도)정보 파일을 지정한다.
17. INPUT_WEATHER_DATA=/user/bigdata/2013_weather.csv
18. # MapReduce로 처리한 결과 데이터파일을 생성할 디렉토리를 지정한다.
19. OUTPUT_DIR=/user/bigdata/card/out/2013
20. # 소스 디렉토리로 이동한다.
21. cd $TRAGET_SOURCE_DIR
22. #컴파일에 필요한 hadoop 라이브러리 패스와 함께 source를 컴파일한다.

```

```

23. javac -classpath /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapr
   ↵ educe-client-core-2.2.0.jar:/usr/local/hadoop/share/hadoop/common/li
   b/commons-cli-1.2.jar:/usr/local/hadoop/share/hadoop/common/hadoo
   p-common-2.2.0.jar $TARGET_SOURCE
24. # 컴파일한 *.class 파일을 jar로 압축한다.
25. jar cf $TARGET_JAR $TARGET_CLASSES
26. # yarn 커맨드로 Hadoop에서 TARGET_JAR 프로그램을 돌려서 Map/Reduce를
   ↵ 실행한다.
27. yarn jar $TARGET_JAR $EXE_CLASS $INPUT_CARD_DATA $INPUT_WEATH
   ↵ ER_DATA $OUTPUT_DIR
28. # 작업 수행이 완료되었다면 소스 디렉토리에서 나온다.
29. cd ..

```



- 데이터 분석 스크립트 소스(04.run.sh)
- 라인 03~11 : 자바 소스를 컴파일하여 클래스 파일을 card.jar 파일로 압축하는 라인이다.
- 라인 13 : 실행시킬 클래스 명을 지정하는 라인이다.
- 라인 15~17 : 하둡 파일 시스템에 있는 소비 가계부 가공 데이터와 기상 가공 데이터 파일을 지정하는 라인이다.
- 라인 19 : 맵리듀스 결과 파일을 저장할 폴더를 지정하는 라인이다.
- 라인 23 : 자바 컴파일을 실행하는 라인이다.
- 라인 25 : jar 명령어를 이용하여 클래스 파일을 card.jar 파일로 압축하는 라인이다.
- yarn jar jar 압축파일명 실행 클래스명 가공 소비 가계부 데이터 가공 기상 데이터 결과 저장 위치

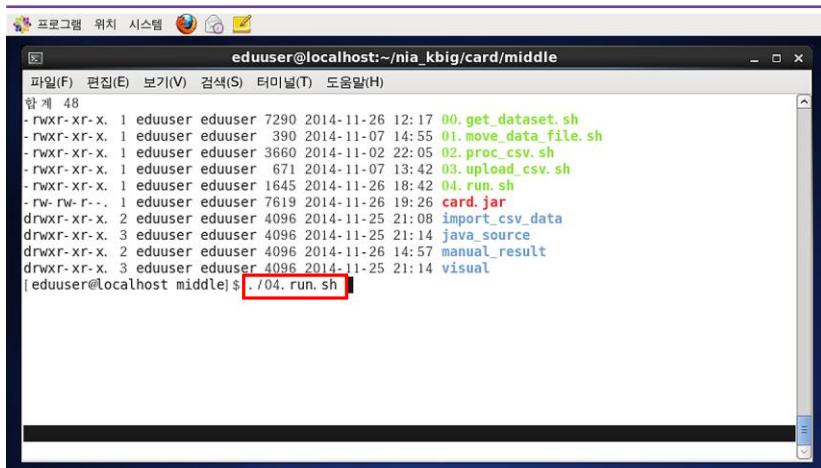
Tip ↵

- 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

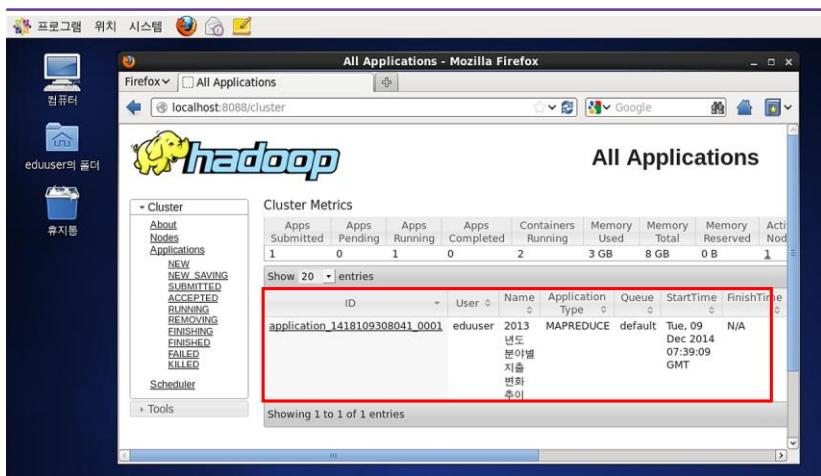
> 분석 맵리듀스 실행



```
eduuser@localhost:~/nia_kbig/card/middle
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
할 게 48
- rwxr-xr-x. 1 eduuser eduuser 7290 2014-11-26 12:17 00.get_dataset.sh
- rwxr-xr-x. 1 eduuser eduuser 390 2014-11-07 14:55 01.move_data_file.sh
- rwxr-xr-x. 1 eduuser eduuser 3660 2014-11-02 22:05 02.proc_csv.sh
- rwxr-xr-x. 1 eduuser eduuser 671 2014-11-07 13:42 03.upload_csv.sh
- rwxr-xr-x. 1 eduuser eduuser 1645 2014-11-26 18:42 04.run.sh
drwxr-xr-x. 2 eduuser eduuser 4096 2014-11-25 21:08 import_csv_data
drwxr-xr-x. 3 eduuser eduuser 4096 2014-11-25 21:14 java_source
drwxr-xr-x. 2 eduuser eduuser 4096 2014-11-26 14:57 manual_result
drwxr-xr-x. 3 eduuser eduuser 4096 2014-11-25 21:14 visual
[eduuser@localhost middle]$ ./04.run.sh
```

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

> 맵리듀스 실행 현황 조회



All Applications - Mozilla Firefox

localhost:8088/cluster

All Applications

ID	User	Name	Application Type	Queue	StartTime	FinishTime
application_1418109308041_0001	eduuser	2013	MAPREDUCE	default	Tue, 09 Dec 2014 07:39:09 GMT	N/A

Show 1 to 1 of 1 entries

▪ 파이어폭스 브라우저를 클릭한 후 주소 입력창에 `http://localhost:8088`을 입력 후 엔터를 치면 맵리듀스 진행 과정을 볼 수 있다.

> 분석 데이터 파일 조회

> 맵리듀스 분석 결과 파일 조회

The screenshot shows the Hadoop NameNode interface running on port 9000. The main content area displays the following information:

- Started:** Tue Nov 25 11:06:11 KST 2014
- Version:** 2.2.0_1529768
- Compiled:** 2013-10-07T06:28Z by hortonmu from branch-2.2.0
- Cluster ID:** CID-97aa1e3-2002-4dad-9652-1be0c3d55ee
- Block Pool ID:** BP-1924143028-127.0.0.1-1413649470850

Below this, there is a red box around the "Browse the filesystem" link. Further down, another red box surrounds the "NameNode Logs" link.

Cluster Summary

Security is OFF
30 files and directories, 14 blocks = 44 total.
Heap Memory used 32.23 MB is 21% of Committed Heap Memory 147 MB. Max Heap Memory is 889 MB.
Non-Heap Memory used 27.21 MB is 94% of Committed Non-Heap Memory 28.94 MB. Max Non-Heap Memory is 214 MB.

Configured Capacity	:	66.50 GB
DFS Used	:	3.89 MB
Non DFS Used	:	12.40 GB
DFS Remaining	:	54.10 GB
DFS Used%	:	0.01%
DFS Remaining%	:	81.35%
Block Pool Used	:	3.89 MB

- 맵리듀스의 분석 결과 파일을 확인하기 위해서 파일어플스 브라우저 창에 **localhost:50070** 입력 후 엔터.
- Browse the filesystem 링크 클릭하여 하위 폴더로 접근하면 출력 결과물 파일이 나온다.

The screenshot shows the contents of the directory **/user/bigdata/card/out/2013**. The table displays the following files:

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
CA-r-00000	file	480 B	1	128 MB	2014-11-26 18:47	rw-r--r--	eduuser	supergroup
CB-r-00000	file	434 B	1	128 MB	2014-11-26 18:47	rw-r--r--	eduuser	supergroup
CC-r-00000	file	430 B	1	128 MB	2014-11-26 18:47	rw-r--r--	eduuser	supergroup
CSV-r-00000	file	589 B	1	128 MB	2014-11-26 18:47	rw-r--r--	eduuser	supergroup
JSON-r-00000	file	1.34 KB	1	128 MB	2014-11-26 18:47	rw-r--r--	eduuser	supergroup
WT-r-00000	file	429 B	1	128 MB	2014-11-26 18:47	rw-r--r--	eduuser	supergroup
SUCCESS	file	0 B	1	128 MB	2014-11-26 18:47	rw-r--r--	eduuser	supergroup

- 출력 결과 폴더 **/user/bigdata/card/out/2013**
CA-r-0000(쇼핑/마트), CB-r-00000(와식/부식), CC-r-00000(커피/간식) – json 파일, CSV-r-0000(분석결과 CSV파일), JSON-r-00000(분석결과 json 파일), WT-r-00000(평균온도 – json 파일)을 조회할 수 있다.

➤ 분석 후 결과 데이터

➤ card_data

```

01. var data =
02. [
03.     {
04.         "kind": " 마트/쇼핑",
05.         "Data": [
06.             { "Date": "201301", "Value": "57420.454545454544" },
07.             { "Date": "201302", "Value": "17619.583333333332" },
08.             { "Date": "201303", "Value": "40968.333333333336" },
09.             { "Date": "201304", "Value": "36798.75" },
10.             { "Date": "201305", "Value": "28981.5625" },
11.             { "Date": "201306", "Value": "43503.846153846156" },
12.             { "Date": "201307", "Value": "16280.0" },
13.             { "Date": "201308", "Value": "24076.666666666668" },
14.             { "Date": "201309", "Value": "106933.3333333333" },
15.             { "Date": "201310", "Value": "19161.25" },
16.             ...
17.         ]
18.     },
19.     {
20.         "kind": " 주거/생활",
21.         "Data": [
22.             { "Date": "201301", "Value": "12000.0" },
23.             ...
24.             { "Date": "201311", "Value": "67320.64285714286" },
25.             { "Date": "201312", "Value": "13924.0" }
26.         ]
27.     }
28. ];

```

> weather

```
01. var temperatures =  
02. [  
03.     ["2013-01-01",-4.7],  
04.     ["2013-01-02",-11.7],  
05.     ["2013-01-03",-13.2],  
06.     ["2013-01-04",-10.7],  
07.     ["2013-01-05",-7],  
08.     ["2013-01-06",-6.3],  
09.     ["2013-01-07",-5.1],  
10.     ["2013-01-08",-4.6],  
11.     ["2013-01-09",-9],  
12.     ["2013-01-10",-8.3],  
13.     ["2013-01-11",-3.2],  
14.     ["2013-01-12",0],  
15.     ["2013-01-13",-0.5],  
16. ]
```



1

2



VI 시각화

개요	107
분석 데이터 저장 방법 1	108
분석 데이터 저장 방법 2	110
시각화 과정	114
분석 데이터 시각화	115
데이터 분석	116

VI

시각화

▶ 개요

소비 가계부 데이터의 하둡 맵리듀스로 분석한 데이터를 시각화하기 위해서 서버에 CSV, JSON 형태의 결과 파일을 저장해야 한다. CSV 형태 파일은 오픈오피스 스프레드시트에서 불러와서 시계열 차트를 생성 할 수 있으며, JSON 형태의 파일은 D3 차트에서 시각화할 때 사용되는 데이터 형식의 파일이다. 소비 가계부 데이터 중 3개 분야(마트/쇼핑, 외식/부식, 커피/간식)에 대해서 사용자의 소비 지출 정보를 분야별, 월별 합계금액을 꺾은선 그래프로 출력하고, 기상 데이터의 평균온도를 차트에 같이 출력하여 온도 변화에 따른 소비패턴을 비교 분석한다.

▶ 시각화 방법 및 활용기술

- 출력 결과 파일은 out으로 지정한 디렉토리 아래에 JSON-r-00000(지출 비용 데이터 JSON 포맷), CSV-r-00000(지출비용 데이터 CSV 포맷), WT -r-00000(평균기온) 이란 이름으로 존재한다.
- 서버 로컬에 다운로드한 파일을 이용하여 card_data.js (JSON 배열), weather.js (자바스크립트 배열) 파일에 저장 후 **/home/eduuser/nia_kbig /card/middle/visual/js/** 폴더로 파일을 복사한 후 D3 Chart의 꺾은선 그래프를 활용하여 시각화한다.

> 분석 데이터 저장 방법 1(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/card/middle/visual/js / 폴더로 파일을 가져온다.

05.hadoop_filecopy.sh (하둡 파일시스템에서 서버 로컬로 파일 복사)

```

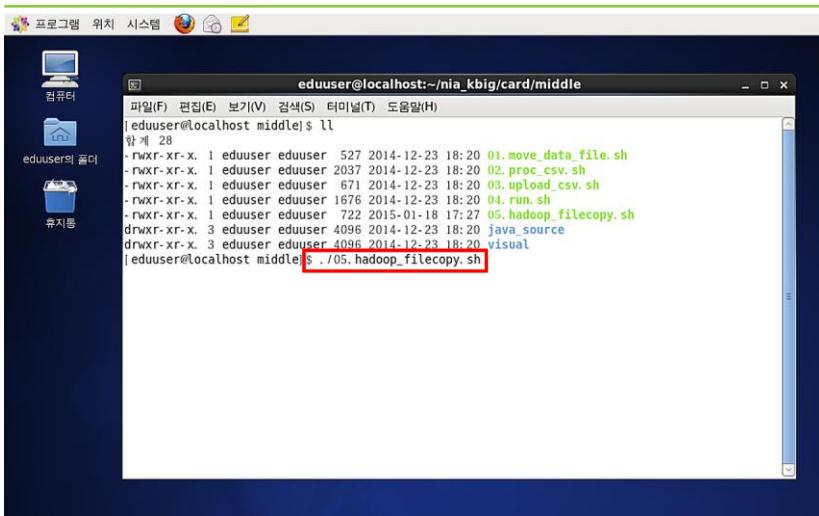
01. # 기존의 빈 파일을 삭제 한다.
02. rm -f visual/js/card_data.js
03. rm -f visual/js/card_data.csv
04. rm -f visual/js/weather.js
05. # 하둡파일 시스템에서 로컬 파일로 다운로드 한다.
06. $ hadoop fs -get /user/bigdata/card/out/2013/JSON-r-00000
    ↪ /home/eduuser/nia_kbig/card/middle/visual/js/card_data.js
07. $ hadoop fs -get /user/bigdata/card/out/2013/CSV-r-00000 /home/eduuse
    ↪ r/nia_kbig/card/middle/visual/js/card_data.csv
08. $ hadoop fs -get /user/bigdata/card/out/2013/WT-r-00000
    ↪ /home/eduuser/nia_kbig/card/middle/visual/js/weather.js

```



- 분석 데이터 저장 스크립트 소스(05.hadoop_filecopy.sh)
- **라인 02** : 소비 가계부 맵리듀스 분석 결과를 card_data.js 파일로 /home/eduuser/nia_kbig/card/middle/visual/js 폴더 위치에 저장하는 라인이다.
- **라인 03** : 소비 가계부 맵리듀스 분석 결과를 card_data.csv 파일로 /home/eduuser/nia_kbig/card/middle/visual/js 폴더 위치에 저장하는 라인이다.
- **라인 04** : 기상 맵리듀스 분석 결과를 weather.js 파일로 /home/eduuser/nia_kbig/card/middle/visual/js 폴더 위치에 저장하는 라인이다.

▶ 데이터 저장 스크립트 실행



```
eduuser@localhost middle]$ ll
합계 28
-rwxr-xr-x. 1 eduuser eduuser 527 2014-12-23 18:20 01.move_data_file.sh
-rwxr-xr-x. 1 eduuser eduuser 2037 2014-12-23 18:20 02.proc_csv.sh
-rwxr-xr-x. 1 eduuser eduuser 671 2014-12-23 18:20 03.upload_csv.sh
-rwxr-xr-x. 1 eduuser eduuser 1676 2014-12-23 18:20 04.run.sh
-rwxr-xr-x. 1 eduuser eduuser 722 2015-01-18 17:27 05.hadoop_filecopy.sh
drwxr-xr-x. 3 eduuser eduuser 4096 2014-12-23 18:20 java_source
drwxr-xr-x. 3 eduuser eduuser 4096 2014-12-23 18:20 visual
[eduuser@localhost middle]$ ./05.hadoop_filecopy.sh
```

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

> 분석 데이터 저장 방법 2

- 웹 브라우저에서 분석 데이터 확인 및 저장

> card_data.js 파일 만들기

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner
CA-r-00000	file	480 B	1	128 MB	2014-11-26 18:47	rw-r--r--	eduuser
CB-r-00000	file	434 B	1	128 MB	2014-11-26 18:47	rw-r--r--	eduuser
CC-r-00000	file	430 B	1	128 MB	2014-11-26 18:47	rw-r--r--	eduuser
CSV-r-00000	file	589 B	1	128 MB	2014-11-26 18:47	rw-r--r--	eduuser
JSON-r-00000	file	1.34 KB	1	128 MB	2014-11-26 18:47	rw-r--r--	eduuser
WT-r-00000	file	429 B	1	128 MB	2014-11-26 18:47	rw-r--r--	eduuser
SUCCESS	file	0 B	1	128 MB	2014-11-26 18:47	rw-r--r--	eduuser
part-r-00000	file	0 B	1	128 MB	2014-11-26	rw-r--r--	eduuser

- 맵리듀스의 결과파일 목록에서 JSON-r-00000 파일 클릭한다.

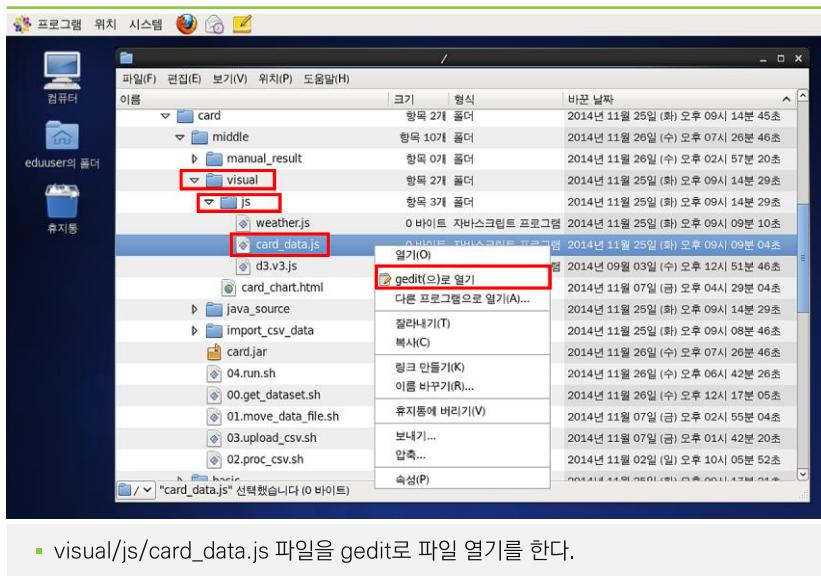
```

[{"Date": "201303", "Value": 308770.0}, {"Date": "201304", "Value": 653970.0}, {"Date": "201305", "Value": 1271590.0}, {"Date": "201306", "Value": 717860.0}, {"Date": "201307", "Value": 304320.0}, {"Date": "201308", "Value": 127810.0}, {"Date": "201309", "Value": 197980.0}, {"Date": "201310", "Value": 185000.0}, {"Date": "201311", "Value": 324960.0}, {"Date": "201312", "Value": 175500.0}], [{"kind": "외가/부식", "Date": [ {"Date": "201301", "Value": 64000.0}, {"Date": "201302", "Value": 175500.0} ]}]

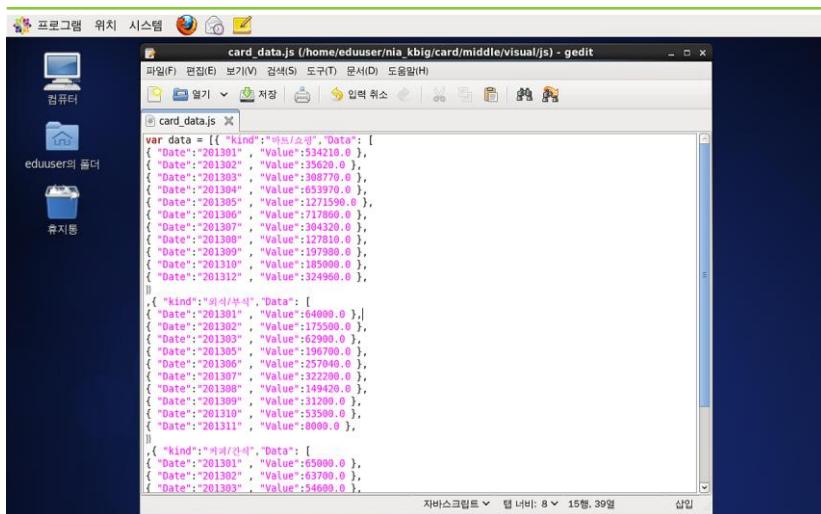
```

- JSON-r-00000 파일 데이터가 출력이 되면 전체 선택 후 복사를 하여 /home/eduuser/nia_kbig/card/middle/visual/js/ 폴더 밑에 card_data.js 파일을 gedit로 파일열기를 한다.

VI. 시각화



- visual/js/card_data.js 파일을 gedit로 파일 열기를 한다.



- gedit로 열기 한 파일에 맵리듀스에서 분석한 JSON 파일 데이터를 복사한 것을 붙여넣기하고 저장한다.

▶ weather.js 파일 만들기

Local logs

WT-r-00000 파일 클릭

- 맵리듀스의 결과 파일 목록에서 목록에서 WT-r-00000 파일 클릭한다.

File: /user/bigdata/card/out/2013/WT-r-00000

Goto : /user/bigdata/card/out/2013/ go

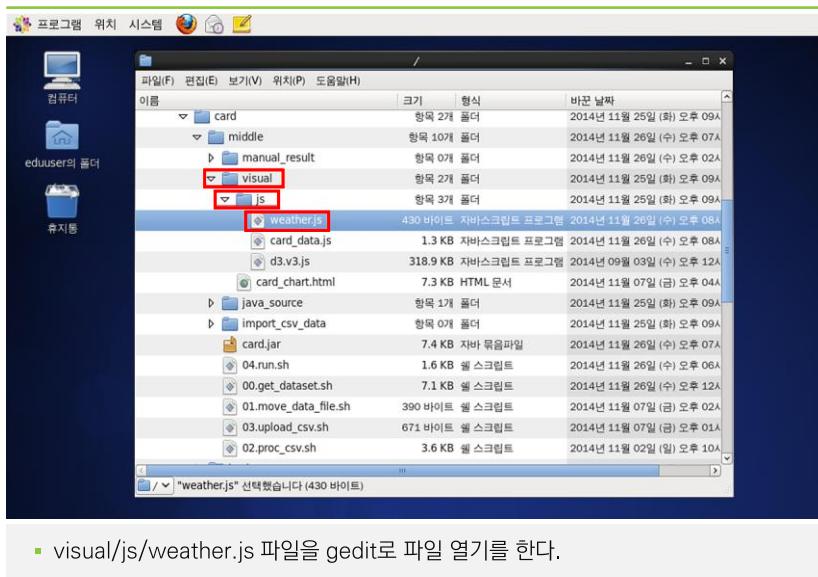
Go back to dir listing
Advanced view/download options

```
var temperatures = [
  ["2013-01", -3.432258064516128],
  ["2013-02", -1.239285714285714],
  ["2013-03", 5.135483870967742],
  ["2013-04", 18.006666666666666],
  ["2013-05", 18.222580645161297],
  ["2013-06", 24.409999999999993],
  ["2013-07", 25.519354838709678],
  ["2013-08", 27.6967741935484],
  ["2013-09", 21.753333333333333],
  ["2013-10", 15.777419354838711],
  ["2013-11", 6.156000000000001],
  ["2013-12", -0.1838709674193556]
]
```

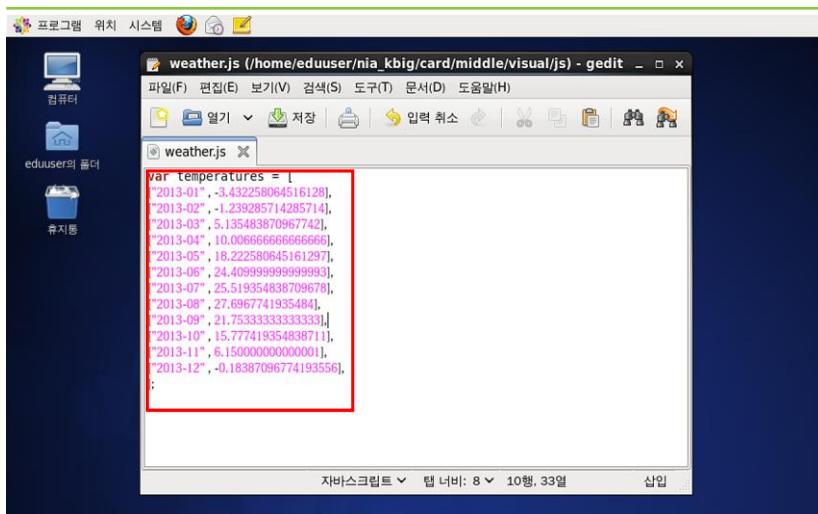
선택 항목:
 1. 절라내기(I)
 2. 복사(C)
 3. 붙여넣기(V)
 4. 삭제(D)
 5. 모두 선택(A) **선택됨**
 6. 요소 검사(Q)

- WT-r-00000 폴더에 데이터가 출력이 되면 전체 선택 후 복사하여 /home/eduuser/nia_kbig/product/middle/visual/js/ 폴더 밑에 weather.js 파일을 gedit로 파일 열기를 한다.

VI. 시각화



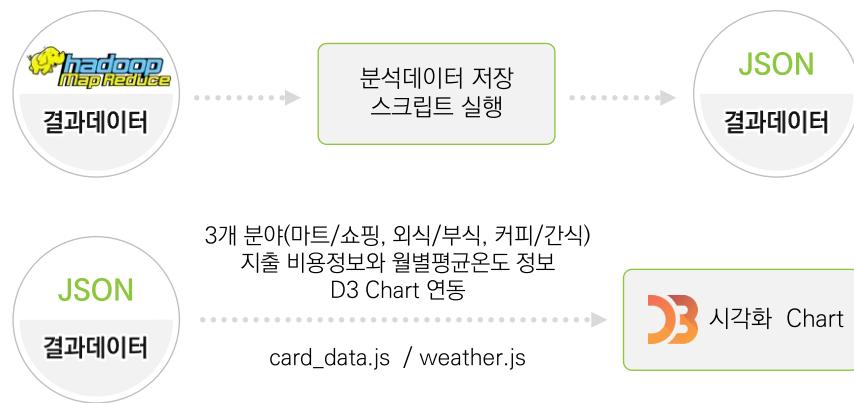
- visual/js/weather.js 파일을 gedit로 파일 열기를 한다.



- gedit로 열기 한 파일에 맵리듀스에서 분석한 JSON 파일 데이터를 복사한 것을 붙여넣기 하고 저장한다.

> 시각화 과정

> 시각화 절차



> 시각화 과정

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

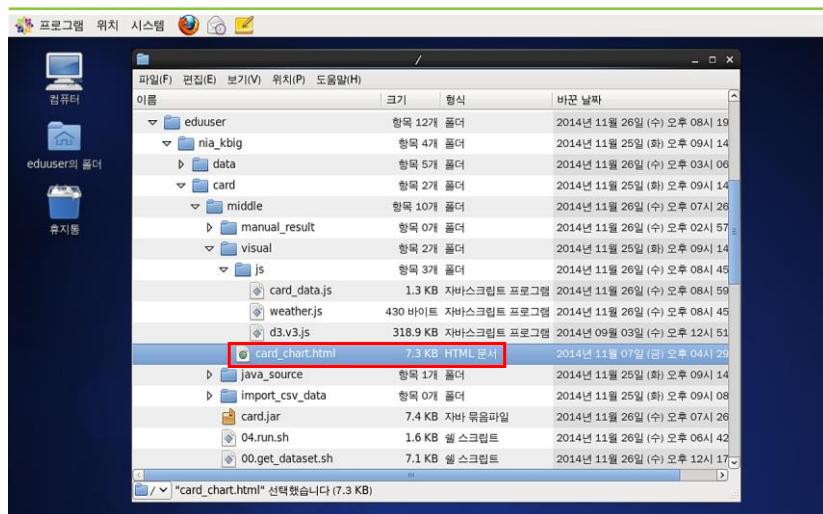
> D3 Chart 모듈 삽입과 결과 데이터 삽입

```

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

▶ 분석 데이터 시각화

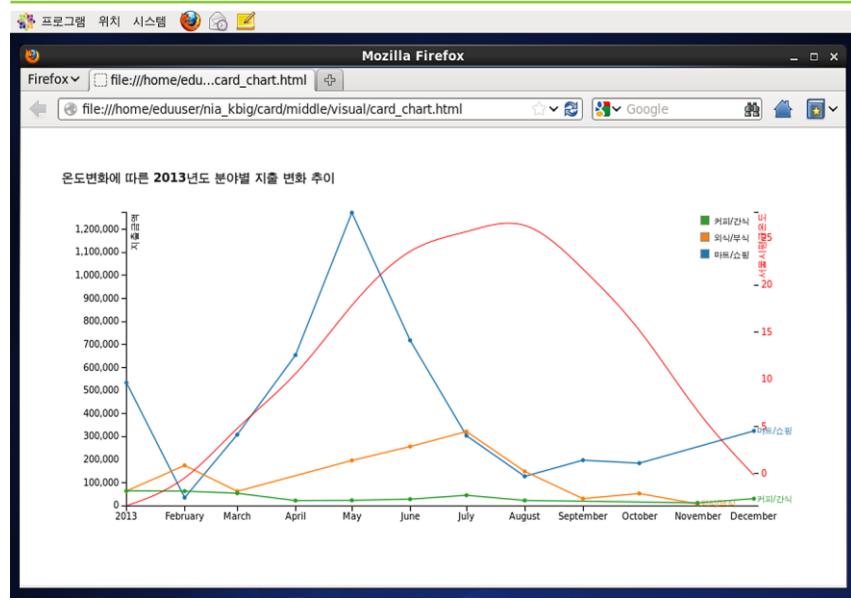
▶ 시각화 차트 실행



- visual 폴더 있는 card_chart.html을 더블클릭하여 ‘표시’ 버튼을 클릭하면 브라우저로 오픈한다.

> 데이터 분석

> 온도변화에 따른 2013년도 분야별 지출 변화 추이



- 외식/부식의 경우 날씨가 더워지면서 외식/부식의 지출이 증가하며 추워지면서 감소하는 패턴을 보인다.
- 커피/간식은 온도에 상관없이 일정하게 비용 지출하는 패턴을 보인다.
- 마트/쇼핑은 계절과 연관성이 있어 보인다.



VII 예제문제

예제 문제1

119

예제 문제2

120

예 / 제 / 문 / 제

예제 1

카테고리 소비 지출 패턴과 강우량을 비교 분석하라.

- 임의의 10인 사용자를 2개 그룹으로 분류하여 강우량에 따라서 카테고리별 소비지출 패턴을 비교하라.

- 소비 가계부 정보에서 임의의 10인 사용자 아이디를 추출한다.
- 5명씩 2개 그룹으로 분리하여 2013년도 지출 정보를 추출한다.
- 그룹별 추출된 금액의 평균을 구하여 A 그룹과 B 그룹으로 분류한다.
- 기상 데이터에서 2013년도 강수량을 추출한다.
- 강수량과 그룹별 카테고리 소비 지출 패턴을 시각화한다.

예제 2

사용자별, 분류별, 일자별 가계 지출 패턴을 분석하라.

- 임의 5인 사용자를 추출하여 5인 사용의 카테고리별 이상 지출 패턴 형태를 찾아내어 비교 분석하라.

- 임의의 5인 사용자의 소비 지출 정보를 추출한다.
- 5인 사용자의 2013년도 월별 소비 지출 정보를 가져온다.
- 사용자별, 카테고리별 소비 지출 정보를 분리한다.
- 사용자별, 카테고리별 소비 패턴을 쳐트화하여 비교 분석한다.

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

