911
통화내역 분석

강민희 박유진 배은구 황혜정 홍민기

INDEX

1

분석개요

출　　처 : Kaggle
데이터 기간 : 2015년 12월 – 2018년 12월
데이터 건수 : 42만 3천건, 8개 카테고리
데이터 내용 : 미국 펜실베이나주 몽고메리 지역의 통화내역

분석개요

kaggle

분석 데이터

펜실베니아주의 몽고메리 지역의 911 통화내역으로서,
크게 응급의료서비스 (EMS), 화재(Fire) 및 교통사고(Traffic)으로 나뉜다.

데이터 출처

Kaggle

분석 의도

2015년 12월 부터 2018년 12월 까지의 총 423909건의 911 통화내역을 분석하여,
각 카테고리별, 도시별, 시간대별 발생 건수를 분석하고자 하였다.

분석 도구

Pig(data loading & parsing), HIVE(analysis), Zeppelin(visualization)

분석개요

8개 컬럼정보

컬럼이름	설명	의미	분석할 컬럼이름
lat	Latitude	위도	-
lng	Longitude	경도	-
desc	Description of Emergency	비상 사태에 대한 설명	-
zip	ZIP Code	우편번호	-
title	Title of Emergency	비상사태 제목 및 내용	title subtitle
time	StampDate and time of the call	통화 시간 및 날짜	etime edate
twp	Town	도시	town
addr	Address	상세 주소	-

분석개요

실제 컬럼정보

	lat	lng	desc	zip	title	timeStamp	twp	addr
0	40.297876	-75.581294	REINDEER CT & DEAD END; NEW HANOVER; Station ...	19525.0	EMS: BACK PAINS/INJURY	2015-12-10 17:40:00	NEW HANOVER	REINDEER CT & DEAD END
1	40.258061	-75.264680	BRIAR PATH & WHITEMARSH LN; HATFIELD TOWNSHIP...	19446.0	EMS: DIABETIC EMERGENCY	2015-12-10 17:40:00	HATFIELD TOWNSHIP	BRIAR PATH & WHITEMARSH LN
2	40.121182	-75.351975	HAWS AVE; NORRISTOWN; 2015-12-10 @ 14:39:21-St...	19401.0	Fire: GAS-ODOR/LEAK	2015-12-10 17:40:00	NORRISTOWN	HAWS AVE
3	40.116153	-75.343513	AIRY ST & SWEDE ST; NORRISTOWN; Station 308A;...	19401.0	EMS: CARDIAC EMERGENCY	2015-12-10 17:40:01	NORRISTOWN	AIRY ST & SWEDE ST
4	40.251492	-75.603350	CHERRYWOOD CT & DEAD END; LOWER POTTSGROVE; S...	NaN	EMS: DIZZINESS	2015-12-10 17:40:01	LOWER POTTSGROVE	CHERRYWOOD CT & DEAD END

2

분석방식

데이터 흐름도 : Pig(Data loading & Parsing), HIVE(Analysis), Zeppelin(Visualization)
DB Table : data911(Main Table), weekofday(Sub Table)
각 항목별 분석 : title, subtitle, year, month, day, time, town

데이터 흐름도

hadoop Echosystem

[Pig]
Data loading & Parsing

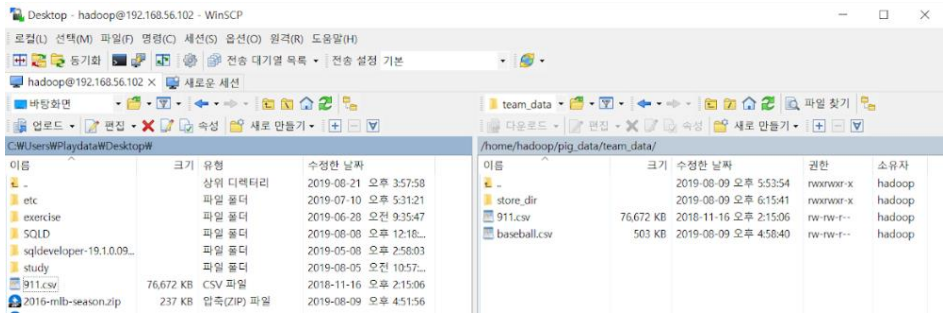
HIVE

[HIVE]
Analysis

Apache Zeppelin

[Zeppelin]
Visualization

Pig



	A	B	C	D	E	F	G	H
1	lat	lng	desc	zip	title	timeStamp	twp	addr
2	40.29788	-75.5813	REINDEER	19525	EMS: BACI	#######	NEW HAN	REINDEER
3	40.25806	-75.2647	BRIAR PAT	19446	EMS: DIAB	#######	HATFIELD	BRIAR PAT
4	40.12118	-75.352	HAWS AVI	19401	Fire: GAS-(	#######	NORRISTC	HAWS AVI
5	40.11615	-75.3435	AIRY ST &	19401	EMS: CARI	#######	NORRISTC	AIRY ST &
6	40.25149	-75.6033	CHERRYWOOD CT &		EMS: DIZZ	#######	LOWER PC	CHERRYW
7	40.25347	-75.2832	CANNON	19446	EMS: HEAI	#######	LANSDALE	CANNON
8	40.18211	-75.1278	LAUREL A	19044	EMS: NAU	#######	HORSHAM	LAUREL A
9	40.21729	-75.4052	COLLEGEV	19426	EMS: RESP	#######	SKIPPACK	COLLEGEV
10	40.28903	-75.3996	MAIN ST &	19438	EMS: SYN(	#######	LOWER SA	MAIN ST &

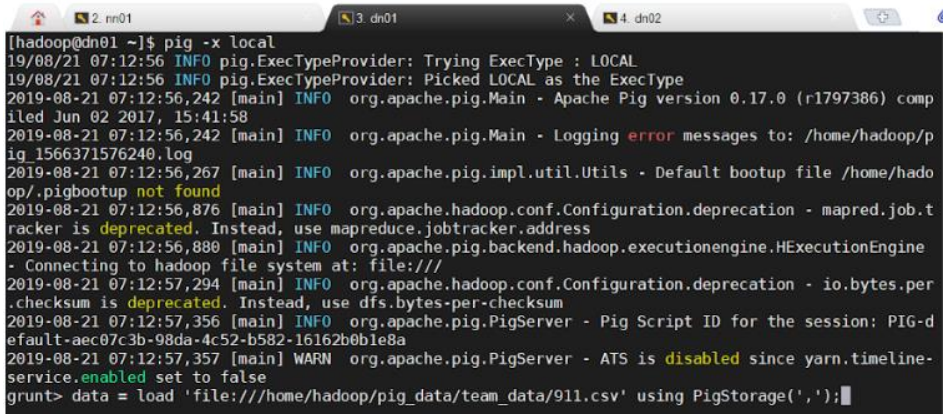
911.csv파일 업로드

WinSCP를 이용해
/home/hadoop/pig_data/team_data/에 911.csv 파일 업로드

911.csv파일의 첫 줄(컬럼명) 삭제

[hadoop@dn01 ~]$ sed -i '1d' /home/hadoop/pig_data/team_data/911.csv

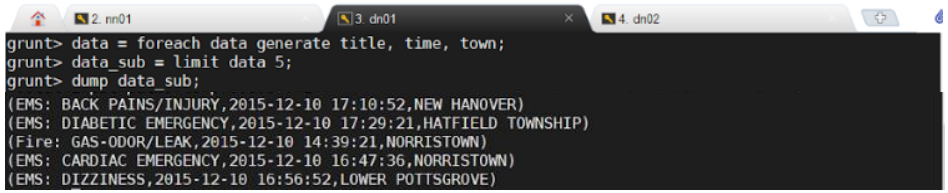
Pig



Pig –x local를 실행해 911.csv 파일을 ‘data’ 로 지정해 로드

```
[hadoop@dn01 ~]$ pig -x local
grunt> data = load 'file:///home/hadoop/pig_data/team_data/911.csv'
>> using PigStorage(',') as (
>> lat:chararray,
>> lng:chararray,
>> desc:chararray,
>> zip:int,
>> title:chararray,
>> time:chararray,
>> town:chararray,
>> addr:chararray
>> );
```

Pig



컬럼명이 title, time, town인 컬럼만 저장

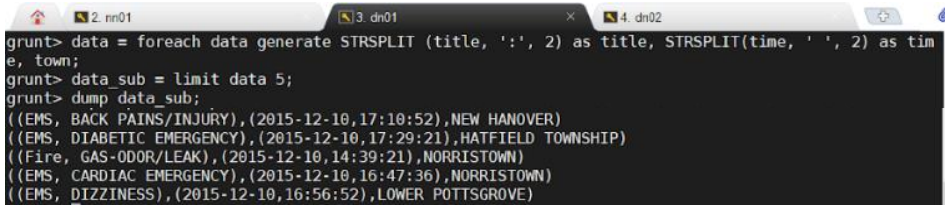
grunt> data = foreach data generate title, time, town;

// 옳은 값만 가져왔는지 data_sub를 만들어 확인
grunt> data_sub = limit data 5;

grunt> dump data_sub;
(EMS: BACK PAINS/INJURY,2015-12-10 17:10:52,NEW HANOVER)
(EMS: DIABETIC EMERGENCY,2015-12-10 17:29:21,HATFIELD TOWNSHIP)
(Fire: GAS-ODOR/LEAK,2015-12-10 14:39:21,NORRISTOWN)
(EMS: CARDIAC EMERGENCY,2015-12-10 16:47:36,NORRISTOWN)
(EMS: DIZZINESS,2015-12-10 16:56:52,LOWER POTTSGROVE)

전체 8개 컬럼 중 title, timeStamp, twn의 컬럼 선택

Pig



컬럼명이 title, time인 컬럼의 값 문자열로 나눔

```
grunt> data = foreach data generate STRSPLIT (title, ': ', 2) as title,
        STRSPLIT(time, ' ', 2) as time, town;

grunt> data_sub = limit data 5;

grunt> dump data_sub;
((EMS, BACK PAINS/INJURY),(2015-12-10,17:10:52),NEW HANOVER)
((EMS, DIABETIC EMERGENCY),(2015-12-10,17:29:21),HATFIELD TOWNSHIP)
((Fire, GAS-ODOR/LEAK),(2015-12-10,14:39:21),NORRISTOWN)
((EMS, CARDIAC EMERGENCY),(2015-12-10,16:47:36),NORRISTOWN)
((EMS, DIZZINESS),(2015-12-10,16:56:52),LOWER POTTSGROVE)
```

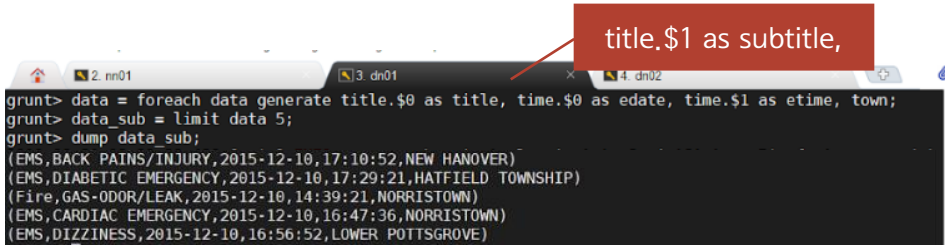
title은 title과 subtitle로 나누어 새로운 두 개의 컬럼 생성

ex) EMS: Back Pains/Injury

timeStamp는 공백을 기준으로 년월일/시분초로 나누어 두 개의 컬럼 생성

ex) 2015-12-10 17:40:00

Pig



컬럼명이 title의 $0, $1, time의 $0, $1, town만 저장

```
grunt> data = foreach data generate title.$0 as title,
        title.$1 as subtitle, time.$0 as edate,
        time.$1 as etime, town;

grunt> data_sub = limit data 5;

grunt> dump data_sub;
(EMS,BACK PAINS/INJURY,2015-12-10,17:10:52,NEW HANOVER)
(EMS,DIABETIC EMERGENCY,2015-12-10,17:29:21,HATFIELD TOWNSHIP)
(Fire,GAS-ODOR/LEAK,2015-12-10,14:39:21,NORRISTOWN)
(EMS,CARDIAC EMERGENCY,2015-12-10,16:47:36,NORRISTOWN)
(EMS,DIZZINESS,2015-12-10,16:56:52,LOWER POTTSGROVE)
```

Pig

```
grunt> store data into '/home/hadoop/pig_data/team_data/store_dir';
```

```
[hadoop@dn01 ~]$ ls /home/hadoop/pig_data/team_data/store_dir911
part-m-00000  part-m-00001  part-m-00002  _SUCCESS
[hadoop@dn01 ~]$ cat /home/hadoop/pig_data/team_data/store_dir911/part-m-00000 | head -10
EMS     BACK PAINS/INJURY       2015-12-10      17:10:52        NEW HANOVER
EMS     DIABETIC EMERGENCY      2015-12-10      17:29:21        HATFIELD TOWNSHIP
Fire    GAS-ODOR/LEAK   2015-12-10      14:39:21        NORRISTOWN
EMS     CARDIAC EMERGENCY       2015-12-10      16:47:36        NORRISTOWN
EMS     DIZZINESS       2015-12-10      16:56:52        LOWER POTTSGROVE
EMS     HEAD INJURY     2015-12-10      15:39:04        LANSDALE
EMS     NAUSEA/VOMITING 2015-12-10      16:46:48        HORSHAM
EMS     RESPIRATORY EMERGENCY   2015-12-10      16:17:05        SKIPPACK
EMS     SYNCOPAL EPISODE        2015-12-10      16:51:42        LOWER SALFORD
Traffic VEHICLE ACCIDENT -      2015-12-10      17:35:41        PLYMOUTH
[hadoop@dn01 ~]$ cat /home/hadoop/pig_data/team_data/store_dir911/part-m-00000 | tail -10
EMS     HEAD INJURY     2017-03-24      22:09:50        FRANCONIA
Traffic VEHICLE ACCIDENT -      2017-03-24      22:13:06        NORRISTOWN
EMS     VEHICLE ACCIDENT        2017-03-24      22:18:00        NORRISTOWN
EMS     DIZZINESS       2017-03-24      22:20:39        WEST POTTSGROVE
EMS     UNKNOWN MEDICAL EMERGENCY       2017-03-24      22:48:21        LOWER MERION
EMS     EYE INJURY      2017-03-24      22:52:52        UPPER MERION
Fire    WOODS/FIELD FIRE        2017-03-24      22:53:56        ABINGTON
EMS     DEHYDRATION     2017-03-24      23:00:24        WORCESTER
Fire    WOODS/FIELD FIRE        2017-03-24      22:53:56        ABINGTON
EMS     CARDIAC EMERGENCY       2017-03-24      22:58:45        NORRISTOWN
```

정제한 값을 ‘data’라는 이름으로 저장

grunt> store data into '/home/hadoop/pig_data/team_data/store_dir911';

저장된 ‘data’값 확인

[hadoop@dn01 ~]$ ls /home/hadoop/pig_data/team_data/store_dir911
part-m-00000 part-m-00001 part-m-00002 _SUCCESS

[hadoop@dn01 ~]$ cat /home/hadoop/pig_data/team_data/
store_dir911/part-m-00000 | head -10
EMS BACK PAINS/INJURY 2015-12-10 17:10:52 NEW HANOVER
EMS DIABETIC EMERGENCY 2015-12-10
17:29:21 HATFIELD TOWNSHIP
Fire GAS-ODOR/LEAK 2015-12-10 14:39:21 NORRISTOWN
.
.
이하 생략

HIVE

```
[hadoop@dn01 ~]$ systemctl status mariadb.service
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2019-08-20 00:35:28 UTC; 1 day 8h ago
  Process: 878 ExecStartPost=/usr/libexec/mariadb-wait-ready $MAINPID (code=exited, status=0/SUCCESS
)
[hadoop@dn01 ~]$ hive --service metastore &
[hadoop@dn01 ~]$ hive --service hiveserver2 &
```

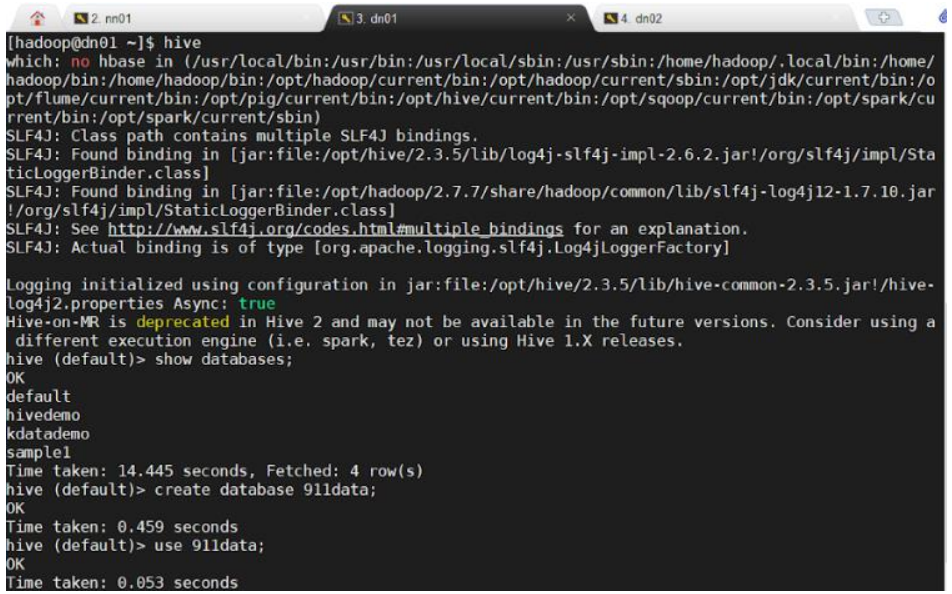
HIVE 실행

// mariadb.service 상태 확인
[hadoop@dn01 ~]$ systemctl status mariadb.service

// 메타스토어 & 하이브서버 구동
[hadoop@dn01 ~]$ hive --service metastore &

[hadoop@dn01 ~]$ hive --service hiveserver2 &

HIVE



```
[hadoop@dn01 ~]$ hive
which: no hbase in (/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/hadoop/.local/bin:/home/
hadoop/bin:/home/hadoop/bin:/opt/hadoop/current/bin:/opt/hadoop/current/sbin:/opt/jdk/current/bin:/o
pt/flume/current/bin:/opt/pig/current/bin:/opt/hive/current/bin:/opt/sqoop/current/bin:/opt/spark/cu
rrent/bin:/opt/spark/current/sbin)
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/hive/2.3.5/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/Sta
ticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/opt/hadoop/2.7.7/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar
!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/opt/hive/2.3.5/lib/hive-common-2.3.5.jar!/hive-
log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a
 different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive (default)> show databases;
OK
default
hivedemo
kdatademo
sample1
Time taken: 14.445 seconds, Fetched: 4 row(s)
hive (default)> create database 911data;
OK
Time taken: 0.459 seconds
hive (default)> use 911data;
OK
Time taken: 0.053 seconds
```

HIVE 실행 및 DB 생성

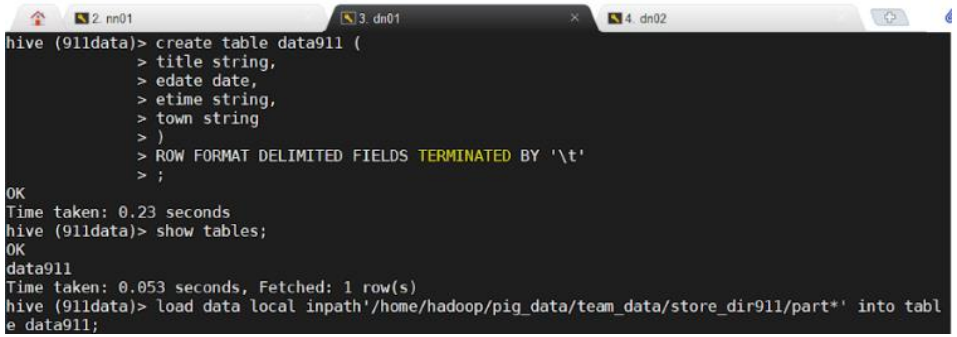
[hadoop@dn01 ~]$ hive

hive (default)> show databases;
OK
default
hivedemo
kdatademo
sample1
Time taken: 14.445 seconds, Fetched: 4 row(s)

hive (default)> create database 911data;
OK
Time taken: 0.459 seconds

hive (default)> use 911data;
OK
Time taken: 0.053 seconds

HIVE



테이블에 생성 및 pig에서 정제한 데이터 삽입

```
hive (911data)> create table data911 (
              > title string,
              > subtitle string,
              > edate date,
              > etime string,
              > town string
              > )
              > ROW FORMAT DELIMITED FIELDS TERMINATED BY '₩t'
              > ;

hive (911data)> show tables;
data911

hive (911data)> load data local inpath'/home/hadoop/pig_data
/team_data/store_dir911/part*' into table data911;
```

HIVE

```
2. nn01   3. dn01   4. dn02
EMS      208676
Traffic  151458
Fire     63775
Time taken: 75.548 seconds, Fetched: 3 row(s)
hive (911data)> select title, count(*) as call_cnt from data911 group by title order by call_cnt des
c;
```

```
%pig.query

b = group emergency by title;
c =foreach b generate group, COUNT($1) as COUNT;
foreach c generate *;
```

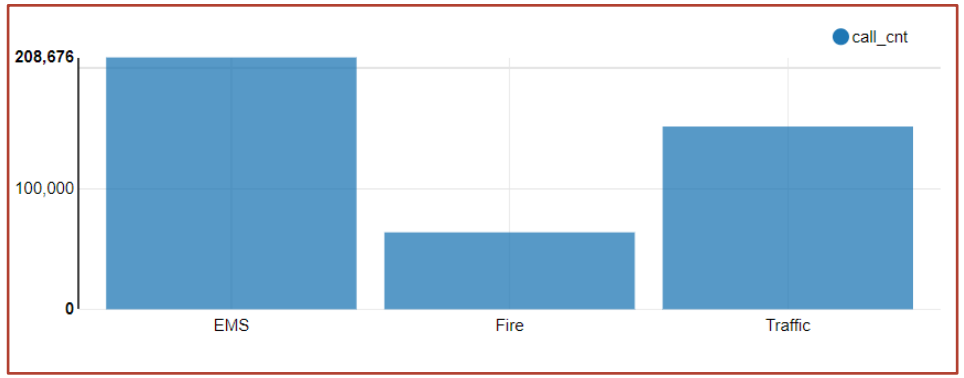
settings

group	COUNT
EMS	208676
Fire	63775
Traffic	151458

** Image produced using Zeppelin

Title을 기준으로 각 목록별 건수 분석

// 문의된 신고 접수의 개수를 title으로 묶어
call_cnt를 오름차순으로 정렬
hive (911data)> select title, count(*) as call_cnt
from data911 group by title order by call_cnt desc;
EMS 208676
Traffic 151458
Fire 63775



** Image produced using Zeppelin

HIVE

```
%pig.query
a = group data2 by content;
b = foreach a generate group, COUNT($1);
c = order b by $1 desc;
d = limit c 10;
foreach d generate $0, $1;
```

FINISHED

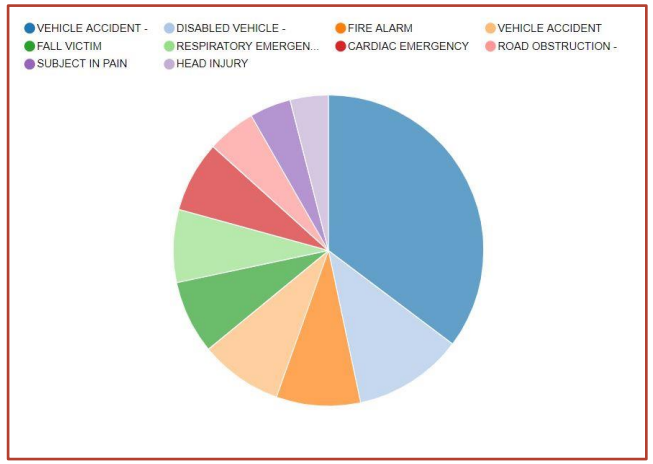
settings

group	col_1
VEHICLE ACCIDENT -	98401
DISABLED VEHICLE -	31871
FIRE ALARM	24459
VEHICLE ACCIDENT	24081
FALL VICTIM	21258
RESPIRATORY EMERGENCY	21159
CARDIAC EMERGENCY	20620
ROAD OBSTRUCTION -	14134
SUBJECT IN PAIN	12004
HEAD INJURY	11105

** Image produced using Zeppelin

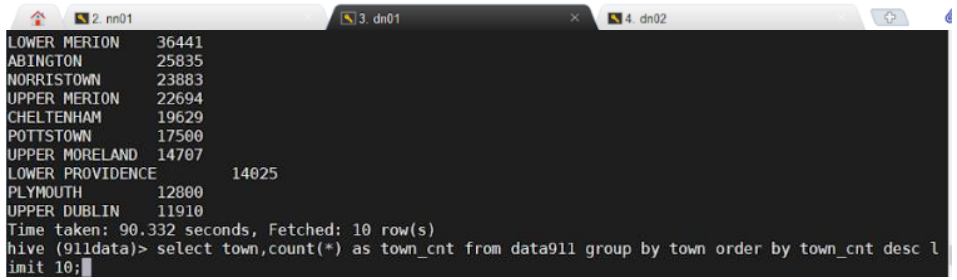
사건 세부목록을 기준으로 각 목록별 건수 분석

// 문의된 신고 접수 세부내용의 개수를 subtitle으로 묶어
call_cnt를 오름차순으로 정렬해 10개만 출력
hive (911data)> select subtitle, count(*) as call_cnt from data911
group by subtitle order by call_cnt desc limit 10;



** Image produced using Zeppelin

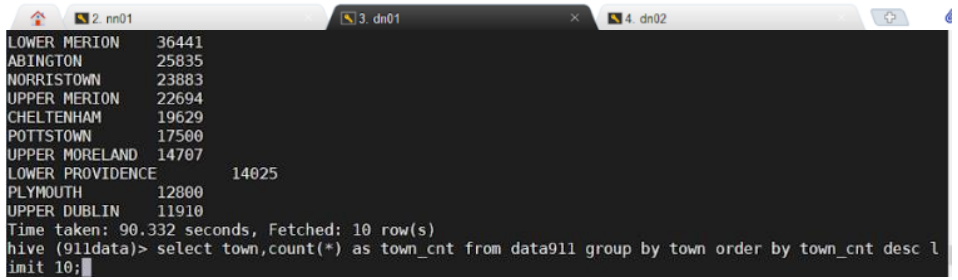
HIVE

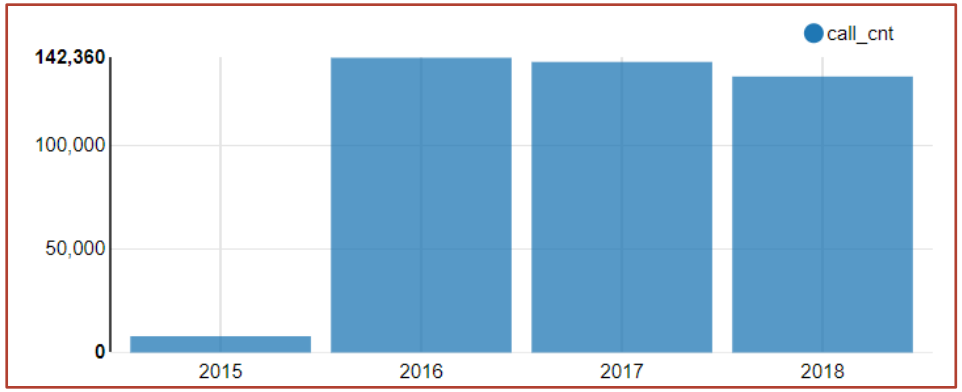


도시별 총 개수에서 내림차순으로 정렬해 상위 10개만 출력

```
hive (911data)> select town,count(*) as town_cnt from data911
group by town order by town_cnt desc limit 10;
LOWER MERION 36441
ABINGTON 25835
NORRISTOWN 23883
UPPER MERION 22694
CHELTENHAM 19629
POTTSTOWN 17500
UPPER MORELAND 14707
LOWER PROVIDENCE 14025
PLYMOUTH 12800
UPPER DUBLIN 11910
```

HIVE





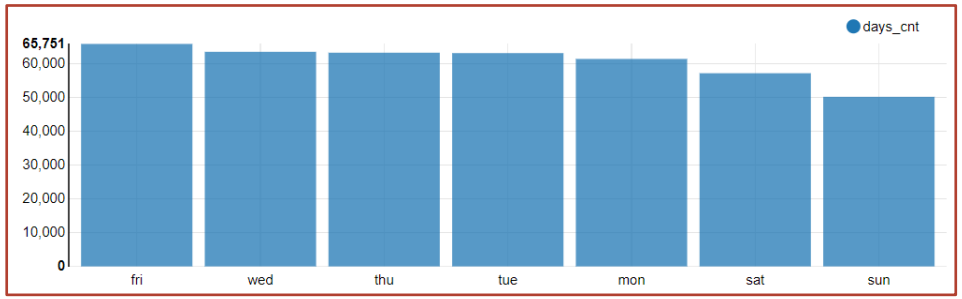
(※ 2015년은 12월 데이터만 있기 때문에 다른 연도에 비해 값이 작음)
** Image produced using Zeppelin

연도별 총 신고건수

```
hive (911data)> select YEAR(edate) as years, count(*) as call_cnt
from data911 group by YEAR(edate) order by years, call_cnt desc;
2015    7916
2016    142360
2017    140343
2018    133290
```

HIVE

```
fri     65751
wed     63423
thu     63159
tue     63041
mon     61318
sat     57093
sun     50124
Time taken: 115.404 seconds, Fetched: 7 row(s)
hive (911data)> select b.word, count(*) as days_cnt from data911 a join weekofday b on (DAYOFWEEK(a.
edate) = b.num) group by b.word order by days_cnt desc;
```



** Image produced using Zeppelin

요일별 총 신고건수

```
hive (911data)> create table weekofday (
              > num int, word string);

hive (911data)> insert into weekofday values
              > (1, 'sun'), (2, 'mon'), (3, 'tue'), (4, 'wed'),
              > (5, 'thu'), (6, 'fri'), (7, 'sat');

hive (911data)> select b.word, count(*) as days_cnt
from data911 a join weekofday b
on (DAYOFWEEK(a.edate) = b.num)
group by b.word order by days_cnt desc;
```

요일 테이블을 추가로 만들어, 두 개의 테이블을 조인하여 출력

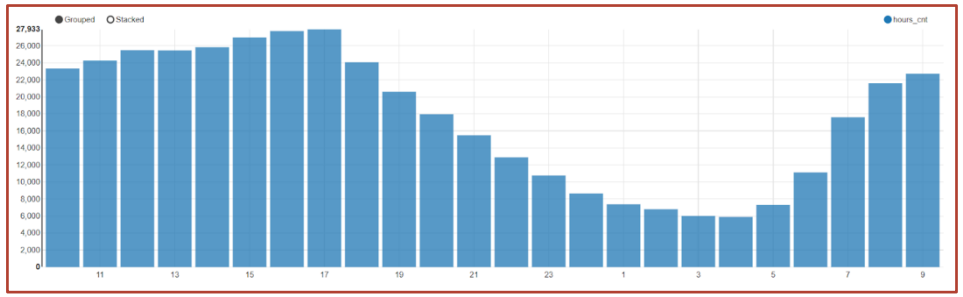
HIVE

```
2. nn01    3. dn01    4. dn02
00      8650
01      7381
02      6796
03      6014
04      5901
05      7307
06      11124
07      17611
08      21601
09      22715
10      23334
11      24263
12      25486
13      25460
14      25836
15      26976
16      27738
17      27933
18      24071
19      20600
20      17960
21      15489
22      12896
23      10767
Time taken: 97.23 seconds, Fetched: 24 row(s)
hive (911data)> select substr(etime, 1, 2) as hours, count(*) as hours_cnt from data911 group by sub
str(etime, 1, 2) order by hours;
```

시간대별 신고건수

```
hive (911data)> select substr(etime, 1, 2) as hours,
count(*) as hours_cnt from data911
group by substr(etime, 1, 2) order by hours;

// 01시~24시로 출력
hive (911data)> select substr(etime, 1, 2)+1 as hours,
count(*) as hours_cnt from data911
group by substr(etime, 1, 2)+1 order by hours;
```



** Image produced using Zeppelin

3

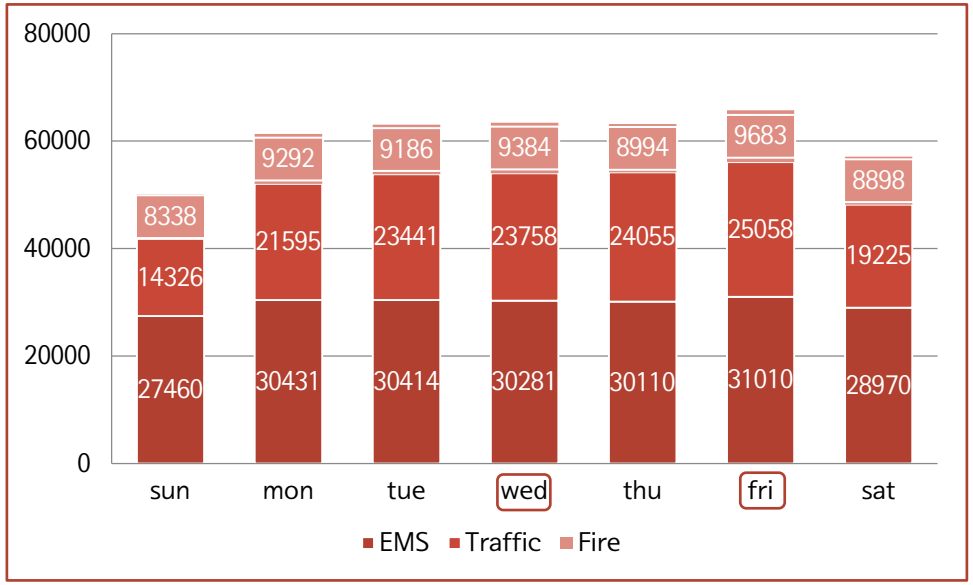
분석결과

최다 통화 목록 : EMS(응급의료서비스)
최다 통화 세부목록 : Vehicle Accident(차량사고)
최다 신고 지역 : Lower Merion
최다 신고 요일 : 금요일
최다 신고 시간 : 오후 6시

분석 결과

가설1

금,토요일에 가장 많은 신고가 접수될 것이다.

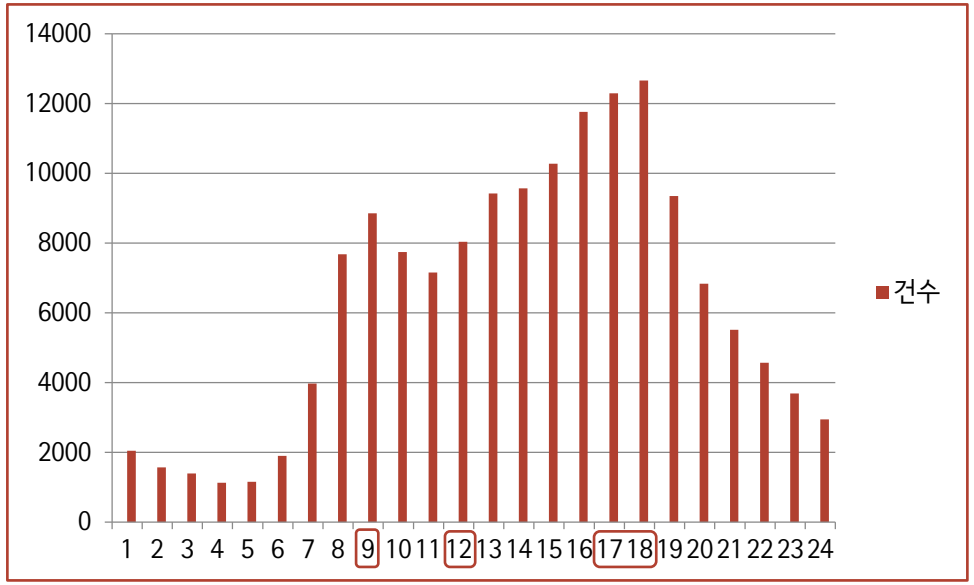


> **금요일**(65751건), **수요일**(63423건)이 가장 많은 신고가 접수되었다.

```
hive (911data)> select b.word, count(*) as days_cnt from data911 a
join weekofday b on (DAYOFWEEK(a.edate) = b.num)
group by b.word order by days_cnt desc;
```

가설2

교통 관련 신고는 출/퇴근 시간에 가장 많이 발생할 것이다.



> 01시~12시는 **9시**(8848건), 13시~24시는 **18시**(12657건)에 가장 많은 교통 관련 신고가 접수되었다.

```
hive (911data)> select substr(etime,1,2)+1 as hours, count(*) as tra_cnt
from data911 where title='Traffic' group by substr(etime, 1, 2)+1;
```

분석 결과

미국 펜실베니아 주 몽고메리 지역의 2015년 12월 부터 2018년 12월 까지의

약 42만 3천건의 911 통화내역을 분석한 결과,

가장 많은 통화 목록은 EMS(응급의료서비스) 요청 건 이며,

세부 항목 중에서는 차량사고(Vehicle Accident)가 가장 많았고,

전체 기간 중 Lower Merion시의 통화건수가 가장 많았으며,

2016년부터 2018년 기간 동안 통화 수는 하락세를 보였고,

가장 많은 통화를 기록한 시간대는 금요일 오후 6시,

가장 적은 시간대는 일요일 오전 5시임

현재 컬럼 별 총 건수를 기반으로 분석하였으나,

추후 컬럼 별 상관관계 등 좀더 다각적이고 심층적인 분석을 해 볼 예정임

THANK YOU :)