# INFO 7250 Final Project Report

## Los Angeles Crime & Arrest Data

Mentor: Yusuf Ozbek

Student: Jingyi Lin

NUID: 001888193

Jingyi

lin.jin@husky.neu.edu

## Table of Contents

# INFO 7250 Final Project Report

## Los Angeles Crime & Arrest Data

Name: Jingyi Lin NUID: 001888193

## 1. Data Resource Introduction

### 1.1 Database Title

Los Angeles Crime & Arrest Data

### 1.2 Database Sources

Los Angeles Open Data

https://www.kaggle.com/cityofLA/los-angeles-crime-arrest-data#crime-data-from-2010-to-present.csv

### 1.3 Database Description

This is a dataset hosted by the city of Los Angeles. The organization has an open data platform found here and they update their information according the amount of data that is brought in. All of the data sources available through the city of Los Angeles organization page.

There are two files in the dataset:

Arrest-data-from-2010-to-present
Crime-data-from-2010-to-present

### 1.4 Problem

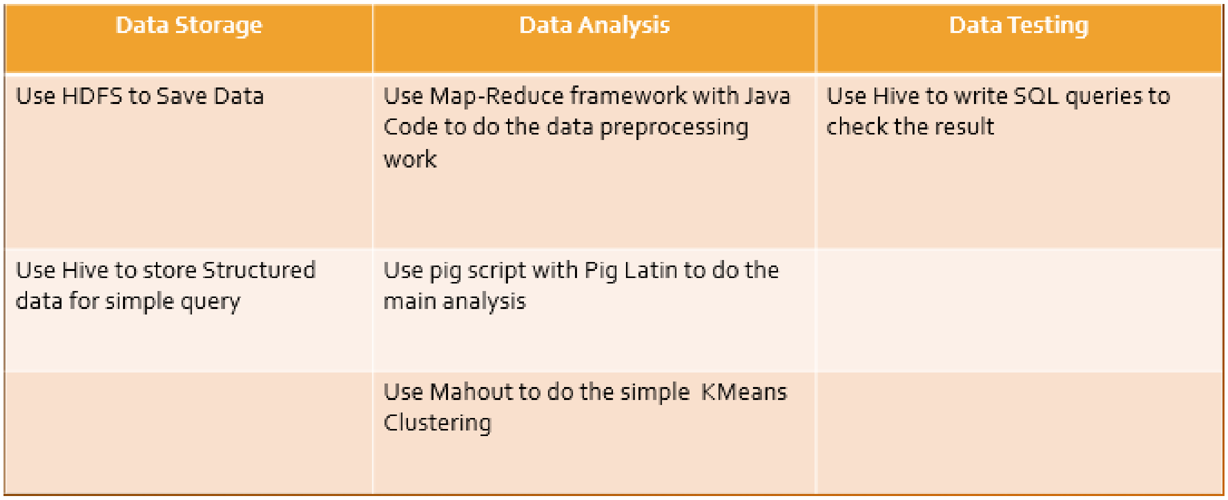
Crime and Arrest situation analysis in LA since 2010 to present

## 2. Technology Stack

### 2.1 Technology

Hadoop, Pig, Hive, Mahout, Map Reduce, HDFS, Ubuntu, Java.

### 2.2 Introduction



| Data Storage | Data Analysis | Data Testing |
|---|---|---|
| Use HDFS to Save Data | Use Map-Reduce framework with Java Code to do the data preprocessing work | Use Hive to write SQL queries to check the result |
| Use Hive to store Structured data for simple query | Use pig script with Pig Latin to do the main analysis | |
| | Use Mahout to do the simple KMeans Clustering | |

HDFS and Hive are used to store data. All the data are saved on HDFS. Hive is used to store structured data for simple query.

Data preprocessing work is finished by Map-Reduce framework with Java code. Pig script with Pig Latin is used to do the main analysis. Hive is used for simple query and result checking.

Finally, Mahout is involved for simple KMeans Clustering.

## 3. Analysis & Implementation

### 3.1 MapReduce Filtering Patterns: Distinct Pattern

#### 3.1.1 Extract Duplicated data from the original dataset and store them in new data files as key value pairs.

**Arrest:**

Area ID/AreaName

| AreaID | AreaName |
|---|---|
| 1 | Central |
| 10 | West Valley |
| 11 | Northeast |
| 12 | 77th Street |
| 13 | Newton |
| … | … |

Charge Code/Charge Description

| Charge | ChargeDescription |
|---|---|
| 103.102LAMC | CAFE ENTERTAINMENT VIOL |
| 103.106BLAM | CONDUCT DANCE W/O PERMIT |
| 103.107.1BL | ESCORT WITHOUT PERMIT |
| 103.107BLAM | RUN ESCORT SERVICE W/O PERMIT |
| 103.112ALAM | BUSINESS REGS |
| … | … |

Charge Group Code/Charge Group Description

| ChargeGroupCode | ChargeGroupDescription |
|---|---|
| 1 | Homicide |
| 10 | Fraud/Embezzlement |
| 11 | Receive Stolen Property |
| 12 | Weapon (carry/poss) |
| 13 | Prostitution/Allied |
| … | … |

Crime:

Area ID/AreaName

| AreaID | AreaName |
|---|---|
| 1 | Central |
| 10 | West Valley |

| | |
|---|---|
| 11 | Northeast |
| 12 | 77th Street |
| 13 | Newton |
| … | … |

Crime Code/Crime Code Description

| CrimeCode | CrimeCodeDescription |
|---|---|
| 110 | CRIMINAL HOMICIDE |
| 113 | MANSLAUGHTER, NEGLIGENT |
| 121 | RAPE, FORCIBLE |
| 122 | RAPE, ATTEMPTED |
| 210 | ROBBERY |
| … | … |

Premise Code/Premise Description

| PremiseCode | PremiseDescription |
|---|---|
| 101 | STREET |
| 102 | SIDEWALK |
| 103 | ALLEY |
| 104 | DRIVEWAY |
| 105 | PEDESTRIAN OVERCROSSING |
| … | … |

Weapon Used Code/Weapon Description

| WeapoUsedCode | WeaponDescription |
|---|---|
| 101 | REVOLVER |
| 102 | HAND GUN |
| 103 | RIFLE |
| 104 | SHOTGUN |
| 105 | SAWED OFF RIFLE/SHOTGUN |
| … | … |

Status Code/Status Description

| StatusCode | StatusDescription |
|---|---|
| 13 | UNK |
| 19 | UNK |
| AA | Adult Arrest |
| AO | Adult Other |
| CC | UNK |
| … | … |

### 3.1.2 Clean main data and add Year, Month features for further analysis.

Arrest:

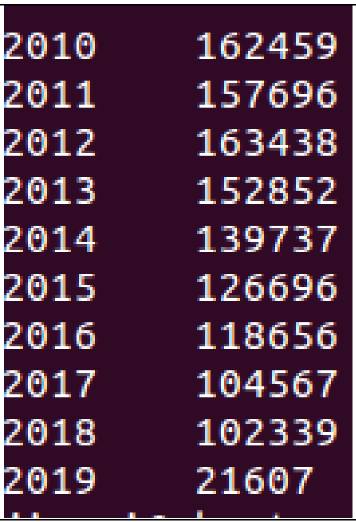
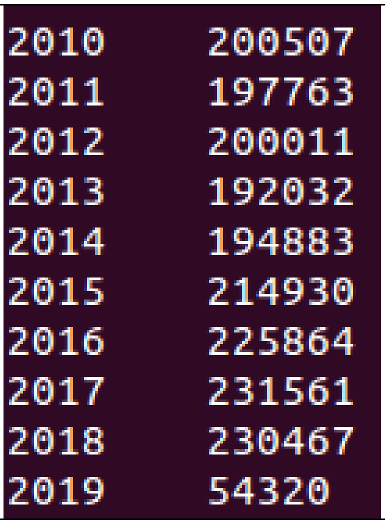
| | | |
|---|---|---|
| Report ID | Number | ID for the arrest |
| Arrest Date | Date | MM/DD/YYYY |

| | | |
|---|---|---|
| Year | String | YYYY |
| Month | String | MM |
| Area ID | Number | The LAPD has 21 Community Police Stations referred to as Geographic Areas within the department. These Geographic Areas are sequentially numbered from 1-21. |
| Reporting District | Number | A four-digit code that represents a sub-area within a Geographic Area. |
| Age | Number | Two character numeric |
| Sex Code | Char | F - Female M – Male |
| Descent Code | Char | Descent Code. |
| Charge Group Code | Char | Category of arrest charge |
| Arrest Type Code | Char | A code to indicate the type of charge the individual was arrested for. |
| Charge | String | The charge the individual was arrested for. |
| Address | String | Street address of crime incident rounded to the nearest hundred block to maintain anonymity. |
| Location | Location | The location where the crime incident occurred. |

Crime:

| Attribute Name | Data Type | Description |
|---|---|---|
| DR Number | Number | Division of Records Number: Official file number made up of a 2 digit year, area ID, and 5 digits |
| Date Reported | Date | MM/DD/YYYY |
| Year | String | YYYY |
| Month | String | MM |
| Date Occurred | Date | MM/DD/YYYY |
| Time Occurred | Number | In 24 hour military time. |
| Area ID | Number | The LAPD has 21 Community Police Stations referred to as Geographic Areas within the department. These Geographic Areas are sequentially numbered from 1-21. |
| Reporting District | Number | A four-digit code that represents a sub-area within a Geographic Area. |
| Crime Code | Number | Indicates the crime committed. (Same as Crime Code 1) |
| Victim Age | Number | Two character numeric |
| Victim Sex | Char | F - Female M - Male X - Unknown |
| Victim Descent | Char | Descent Code |
| Premise Code | Number | The type of structure, vehicle, or location where the crime took place. |
| Weapon Used Code | Number | The type of weapon used in the crime. |
| Status Code | String | Status of the case. (IC is the default) |
| Crime Code | Number | May contain a code for an additional crime, less serious than Crime Code 1. |
| Address | String | Street address of crime incident rounded to the nearest hundred block to maintain anonymity. |
| Location | Location | The location where the crime incident occurred. Actual address is omitted for confidentiality. XY coordinates reflect the nearest 100 block. |

3.2 Use MapReduce Summarization Patterns: Counter Pattern:

| Count arrest number by Year. | Count crime reported number by Year. |
|---|---|
| <br>2010 162459<br>2011 157696<br>2012 163438<br>2013 152852<br>2014 139737<br>2015 126696<br>2016 118656<br>2017 104567<br>2018 102339<br>2019 21607 | <br>2010 200507<br>2011 197763<br>2012 200011<br>2013 192032<br>2014 194883<br>2015 214930<br>2016 225864<br>2017 231561<br>2018 230467<br>2019 54320 |

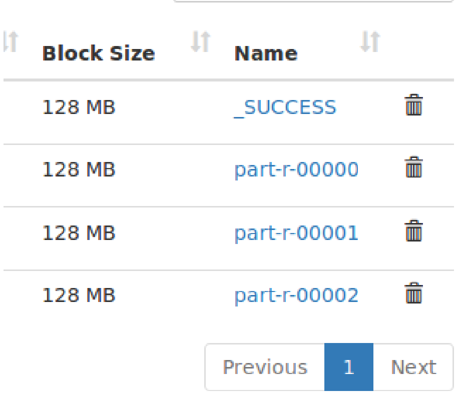
3.3 Use MapReduce Organization Patterns: Partitioning Pattern:

Split two main datasets by YEAR for analysis convenience. Use Partitioner to implement the splitting in MapReduce framework.

part-r-00000: 2010-2012

part-r-00000: 2013-2015

part-r-00000: 2016-2019



| Block Size | Name | |
|---|---|---|
| 128 MB | _SUCCESS | 🗑 |
| 128 MB | part-r-00000 | 🗑 |
| 128 MB | part-r-00001 | 🗑 |
| 128 MB | part-r-00002 | 🗑 |

3.4 Use MapReduce to prepare data for Mahout.

List all the weapons used in the crime dataset and store to a new file.

3.5 Use Pig to Analyze

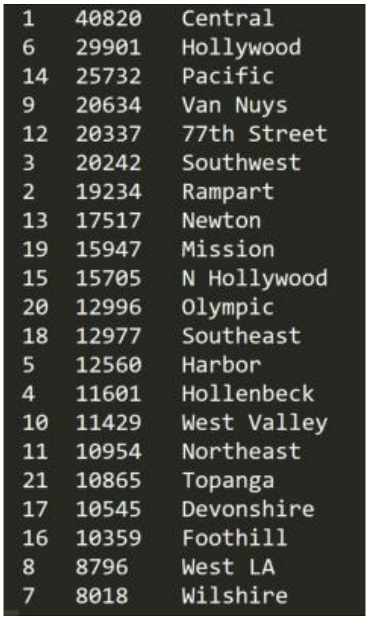
Chose data from 2013-2015 and 2016-2019 to do the analysis. All the resource and result of Pig are stored in HDFS. The command to run the Pig Latin Script is:

```
pig –x mapreduce xx.pig
```

3.5.1 Arrest Analysis:

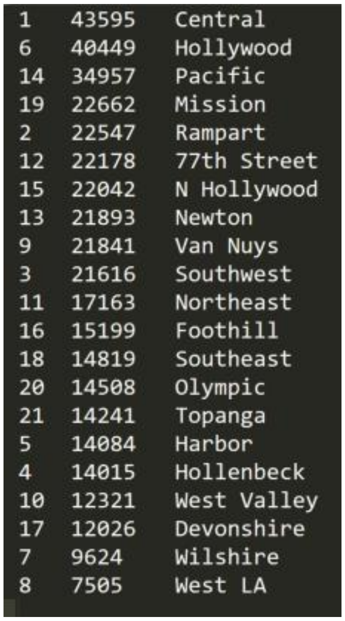
(1) Sorted Area by the incidence of Arrest: left outer join.

2016-2019



```
1   40820   Central
6   29901   Hollywood
14  25732   Pacific
9   20634   Van Nuys
12  20337   77th Street
3   20242   Southwest
2   19234   Rampart
13  17517   Newton
19  15947   Mission
15  15705   N Hollywood
20  12996   Olympic
18  12977   Southeast
5   12560   Harbor
4   11601   Hollenbeck
10  11429   West Valley
11  10954   Northeast
21  10865   Topanga
17  10545   Devonshire
16  10359   Foothill
8   8796    West LA
7   8018    Wilshire
```

2013-2015



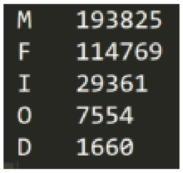
```
1   43595   Central
6   40449   Hollywood
14  34957   Pacific
19  22662   Mission
2   22547   Rampart
12  22178   77th Street
15  22042   N Hollywood
13  21893   Newton
9   21841   Van Nuys
3   21616   Southwest
11  17163   Northeast
16  15199   Foothill
18  14819   Southeast
20  14508   Olympic
21  14241   Topanga
5   14084   Harbor
4   14015   Hollenbeck
10  12321   West Valley
17  12026   Devonshire
7   9624    Wilshire
8   7505    West LA
```

Conclusion: Central is the area where Arrest happens most and Hollywood, Pacific and Wan Nuys follow by.

(2) Sorted Arrest type by the incidence of Arrest.

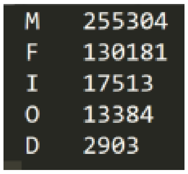
D - Dependent F - Felony I - Infraction M - Misdemeanor O – Other

2016-2019



```
M   193825
F   114769
I   29361
O   7554
D   1660
```

2013-2015



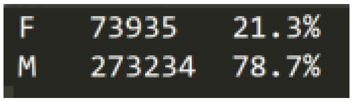
```
M   255304
F   130181
I   17513
O   13384
D   2903
```

Conclusion: From the result we can conclude that most of people are arrested for misdemeanor and half less people are arrested for felony. We can also conclude that the number of each kind of arrest is declining.

(3) Proportion of 2 genders being arrested.

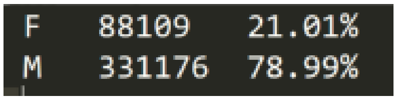
COUNT, SUM, ROUND_TO and CONCAT functions are used.

2016-2019



```
F   73935    21.3%
M   273234   78.7%
```

2013-2015

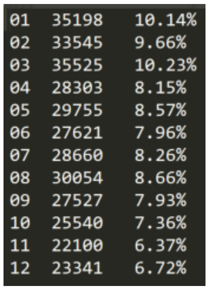


```
F   88109    21.01%
M   331176   78.99%
```

Conclusion: From the result we can conclude that males are the main group of detainees. We can also conclude that the total number of people being arrested is declining.

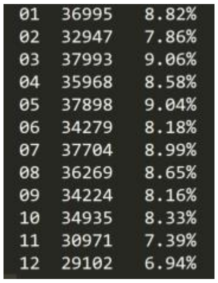
(4) Month ratio of Arrest.

2016-2019



```
01  35198   10.14%
02  33545   9.66%
03  35525   10.23%
04  28303   8.15%
05  29755   8.57%
06  27621   7.96%
07  28660   8.26%
08  30054   8.66%
09  27527   7.93%
10  25540   7.36%
11  22100   6.37%
12  23341   6.72%
```

2013-2015



```
01  36995   8.82%
02  32947   7.86%
03  37993   9.06%
04  35968   8.58%
05  37898   9.04%
06  34279   8.18%
07  37704   8.99%
08  36269   8.65%
09  34224   8.16%
10  34935   8.33%
11  30971   7.39%
12  29102   6.94%
```

Conclusion: Arrest happens most on January, February and March. Cold winter has the least number of arrest.

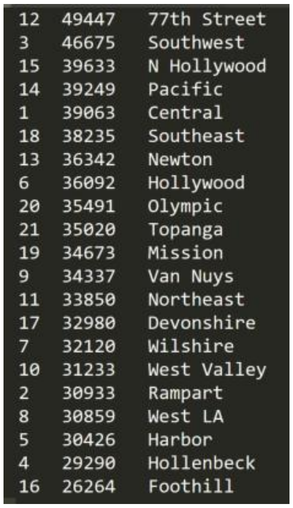
(5) Proportion of different age being arrested.

The result is from 0 years old to 92 years old. More analysis of this topic will be discussed in Hive Part.

3.5.2 Crime Analysis:

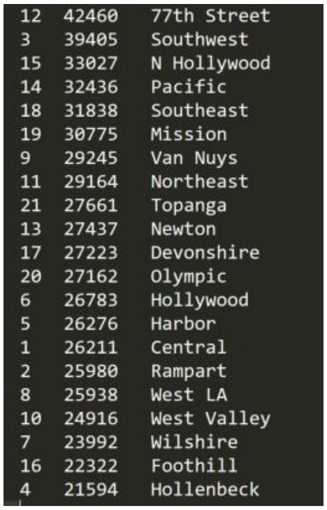
(1) Sorted area by the incidence of Crime reported in 2016-2019: left outer join

2016-2019



```
12  49447   77th Street
3   46675   Southwest
15  39633   N Hollywood
14  39249   Pacific
1   39063   Central
18  38235   Southeast
13  36342   Newton
6   36092   Hollywood
20  35491   Olympic
21  35020   Topanga
19  34673   Mission
9   34337   Van Nuys
11  33850   Northeast
17  32980   Devonshire
7   32120   Wilshire
10  31233   West Valley
2   30933   Rampart
8   30859   West LA
5   30426   Harbor
4   29290   Hollenbeck
16  26264   Foothill
```

2013-2015



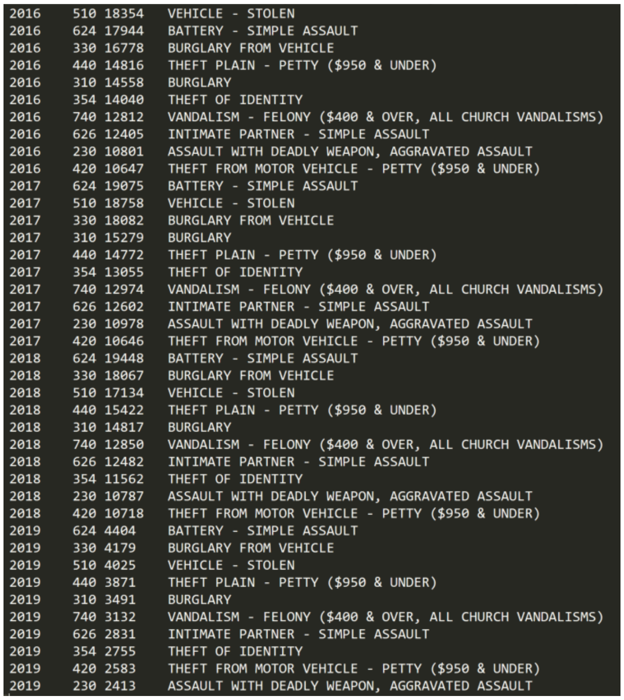
```
12  42460   77th Street
3   39405   Southwest
15  33027   N Hollywood
14  32436   Pacific
18  31838   Southeast
19  30775   Mission
9   29245   Van Nuys
11  29164   Northeast
21  27661   Topanga
13  27437   Newton
17  27223   Devonshire
20  27162   Olympic
6   26783   Hollywood
5   26276   Harbor
1   26211   Central
2   25980   Rampart
8   25938   West LA
10  24916   West Valley
7   23992   Wilshire
16  22322   Foothill
4   21594   Hollenbeck
```

Conclusion: 77th Street is the area where Crime happens most and Southwest, N Hollywood and Pacific follow by.

(2) Which kind of crime occurs most frequently each year top 10?

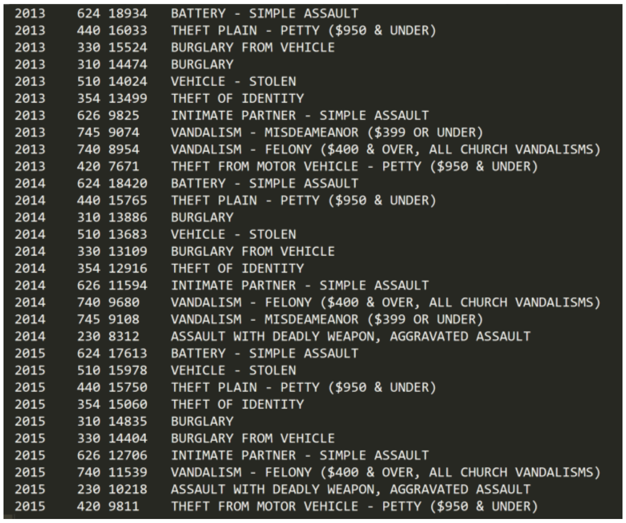
Use JOIN, Secondary Sorting and Limit.

2016-2019



```
2016    510 18354   VEHICLE - STOLEN
2016    624 17944   BATTERY - SIMPLE ASSAULT
2016    330 16778   BURGLARY FROM VEHICLE
2016    440 14816   THEFT PLAIN - PETTY ($950 & UNDER)
2016    310 14558   BURGLARY
2016    354 14040   THEFT OF IDENTITY
2016    740 12812   VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)
2016    626 12405   INTIMATE PARTNER - SIMPLE ASSAULT
2016    230 10801   ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT
2016    420 10647   THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)
2017    624 19075   BATTERY - SIMPLE ASSAULT
2017    510 18758   VEHICLE - STOLEN
2017    330 18082   BURGLARY FROM VEHICLE
2017    310 15279   BURGLARY
2017    440 14772   THEFT PLAIN - PETTY ($950 & UNDER)
2017    354 13055   THEFT OF IDENTITY
2017    740 12974   VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)
2017    626 12602   INTIMATE PARTNER - SIMPLE ASSAULT
2017    230 10978   ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT
2017    420 10646   THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)
2018    624 19448   BATTERY - SIMPLE ASSAULT
2018    330 18067   BURGLARY FROM VEHICLE
2018    510 17134   VEHICLE - STOLEN
2018    440 15422   THEFT PLAIN - PETTY ($950 & UNDER)
2018    310 14817   BURGLARY
2018    740 12850   VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)
2018    626 12482   INTIMATE PARTNER - SIMPLE ASSAULT
2018    354 11562   THEFT OF IDENTITY
2018    230 10787   ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT
2018    420 10718   THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)
2019    624 4404    BATTERY - SIMPLE ASSAULT
2019    330 4179    BURGLARY FROM VEHICLE
2019    510 4025    VEHICLE - STOLEN
2019    440 3871    THEFT PLAIN - PETTY ($950 & UNDER)
2019    310 3491    BURGLARY
2019    740 3132    VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)
2019    626 2831    INTIMATE PARTNER - SIMPLE ASSAULT
2019    354 2755    THEFT OF IDENTITY
2019    420 2583    THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)
2019    230 2413    ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT
```

2013-2015

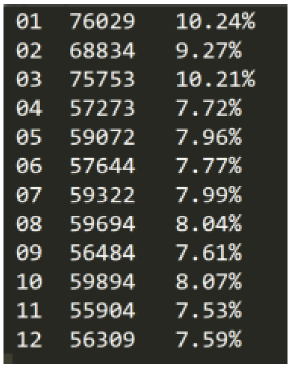


```
2013    624 18934   BATTERY - SIMPLE ASSAULT
2013    440 16033   THEFT PLAIN - PETTY ($950 & UNDER)
2013    330 15524   BURGLARY FROM VEHICLE
2013    310 14474   BURGLARY
2013    510 14024   VEHICLE - STOLEN
2013    354 13499   THEFT OF IDENTITY
2013    626 9825    INTIMATE PARTNER - SIMPLE ASSAULT
2013    745 9074    VANDALISM - MISDEAMEANOR ($399 OR UNDER)
2013    740 8954    VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)
2013    420 7671    THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)
2014    624 18420   BATTERY - SIMPLE ASSAULT
2014    440 15765   THEFT PLAIN - PETTY ($950 & UNDER)
2014    310 13886   BURGLARY
2014    510 13683   VEHICLE - STOLEN
2014    330 13109   BURGLARY FROM VEHICLE
2014    354 12916   THEFT OF IDENTITY
2014    626 11594   INTIMATE PARTNER - SIMPLE ASSAULT
2014    740 9680    VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)
2014    745 9108    VANDALISM - MISDEAMEANOR ($399 OR UNDER)
2014    230 8312    ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT
2015    624 17613   BATTERY - SIMPLE ASSAULT
2015    510 15978   VEHICLE - STOLEN
2015    440 15750   THEFT PLAIN - PETTY ($950 & UNDER)
2015    354 15060   THEFT OF IDENTITY
2015    310 14835   BURGLARY
2015    330 14404   BURGLARY FROM VEHICLE
2015    626 12706   INTIMATE PARTNER - SIMPLE ASSAULT
2015    740 11539   VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)
2015    230 10218   ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT
2015    420 9811    THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)
```

Conclusion: VEHCLE – STOLEN, BATTERY – SIMPLE ASSULT and BURGLARY FROM VEHICLE are the top three occurring most frequently crime in 2016, 2017, 2018 and the first 4 month of 2019.

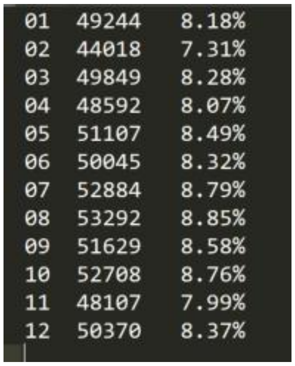
(3) Month ratio of Crime
Use COUNT, SUM, ROUND_TO, CONCAT

2016-2019



```
01  76029   10.24%
02  68834   9.27%
03  75753   10.21%
04  57273   7.72%
05  59072   7.96%
06  57644   7.77%
07  59322   7.99%
08  59694   8.04%
09  56484   7.61%
10  59894   8.07%
11  55904   7.53%
12  56309   7.59%
```

2013-2015



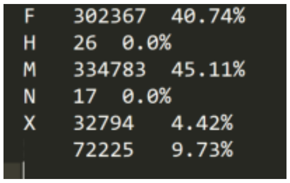
```
01  49244   8.18%
02  44018   7.31%
03  49849   8.28%
04  48592   8.07%
05  51107   8.49%
06  50045   8.32%
07  52884   8.79%
08  53292   8.85%
09  51629   8.58%
10  52708   8.76%
11  48107   7.99%
12  50370   8.37%
```

Conclusion: Crime happens most on January, February and March. Cold winter has the least number of Crime.

(4) Proportion of different Victim age.
The result is from -9 years old to 118 years old. More analysis of this topic will be discussed in Hive Part.

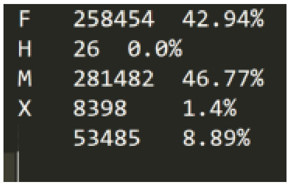
(5) Proportion of different Vitim gender.

2016-2019



```
F   302367  40.74%
H   26  0.0%
M   334783  45.11%
N   17  0.0%
X   32794   4.42%
    72225   9.73%
```

2013-2015

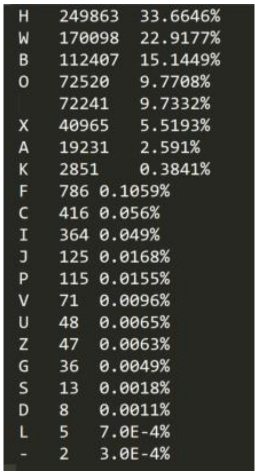


```
F   258454  42.94%
H   26  0.0%
M   281482  46.77%
X   8398    1.4%
    53485   8.89%
```

Conclusion: Males and Females have almost the same possibility to become the victim of crime. The victim number of Crime is declining.

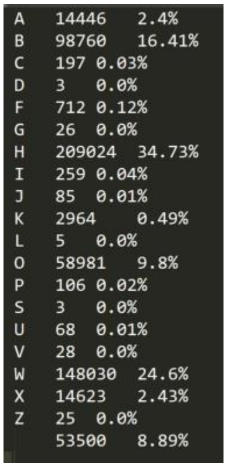
(6) Proportion of different Victim descent.
Descent Code: A - Other Asian B - Black C - Chinese D - Cambodian F - Filipino G - Guamanian H - Hispanic/Latin/Mexican I - American Indian/Alaskan Native J - Japanese K - Korean L - Laotian O - Other P - Pacific Islander S - Samoan U - Hawaiian V - Vietnamese W - White X - Unknown Z - Asian Indian

2016-2019



```
H   249863  33.6646%
W   170098  22.9177%
B   112407  15.1449%
O   72520   9.7708%
    72241   9.7332%
X   40965   5.5193%
A   19231   2.591%
K   2851    0.3841%
F   786 0.1059%
C   416 0.056%
I   364 0.049%
J   125 0.0168%
P   115 0.0155%
V   71  0.0096%
U   48  0.0065%
Z   47  0.0063%
G   36  0.0049%
S   13  0.0018%
D   8   0.0011%
L   5   7.0E-4%
-   2   3.0E-4%
```

2013-2015

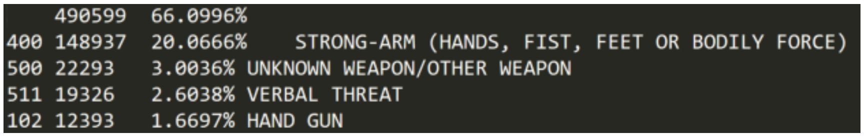


```
A   14446   2.4%
B   98760   16.41%
C   197 0.03%
D   3   0.0%
F   712 0.12%
G   26  0.0%
H   209024  34.73%
I   259 0.04%
J   85  0.01%
K   2964    0.49%
L   5   0.0%
O   58981   9.8%
P   106 0.02%
S   3   0.0%
U   68  0.01%
V   28  0.0%
W   148030  24.6%
X   14623   2.43%
Z   25  0.0%
    53500   8.89%
```

Conclusion: Victim Descent Top 3: Hispanic/Latin/Mexican, White and Black.

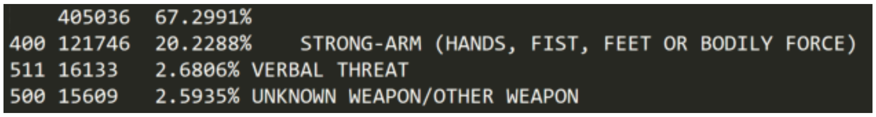
(7) Proportion of different Weapons used in crime.

2016-2019 (top 3)



```
    490599  66.0996%
400 148937  20.0666%    STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)
500 22293   3.0036% UNKNOWN WEAPON/OTHER WEAPON
511 19326   2.6038% VERBAL THREAT
102 12393   1.6697% HAND GUN
```

2013-2015 (top3)

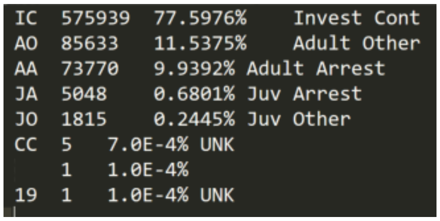


```
    405036  67.2991%
400 121746  20.2288%    STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)
511 16133   2.6806% VERBAL THREAT
500 15609   2.5935% UNKNOWN WEAPON/OTHER WEAPON
```

Conclusion: Most of crimes reported do not have weapons. STRONG-ARM is the weapon being used most in crime.

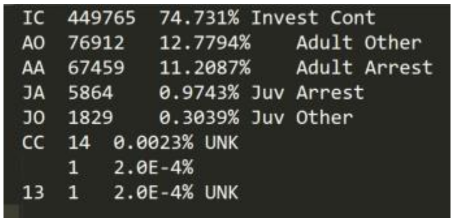
(8) Proportion of different Status of crime.

2016-2019



```
IC  575939  77.5976%    Invest Cont
AO  85633   11.5375%    Adult Other
AA  73770   9.9392% Adult Arrest
JA  5048    0.6801% Juv Arrest
JO  1815    0.2445% Juv Other
CC  5   7.0E-4% UNK
    1   1.0E-4%
19  1   1.0E-4% UNK
```

2013-2015



```
IC  449765  74.731% Invest Cont
AO  76912   12.7794%    Adult Other
AA  67459   11.2087%    Adult Arrest
JA  5864    0.9743% Juv Arrest
JO  1829    0.3039% Juv Other
CC  14  0.0023% UNK
    1   2.0E-4%
13  1   2.0E-4% UNK
```

Conclusion: Most Crime being reported even several years ago are under Investigation.

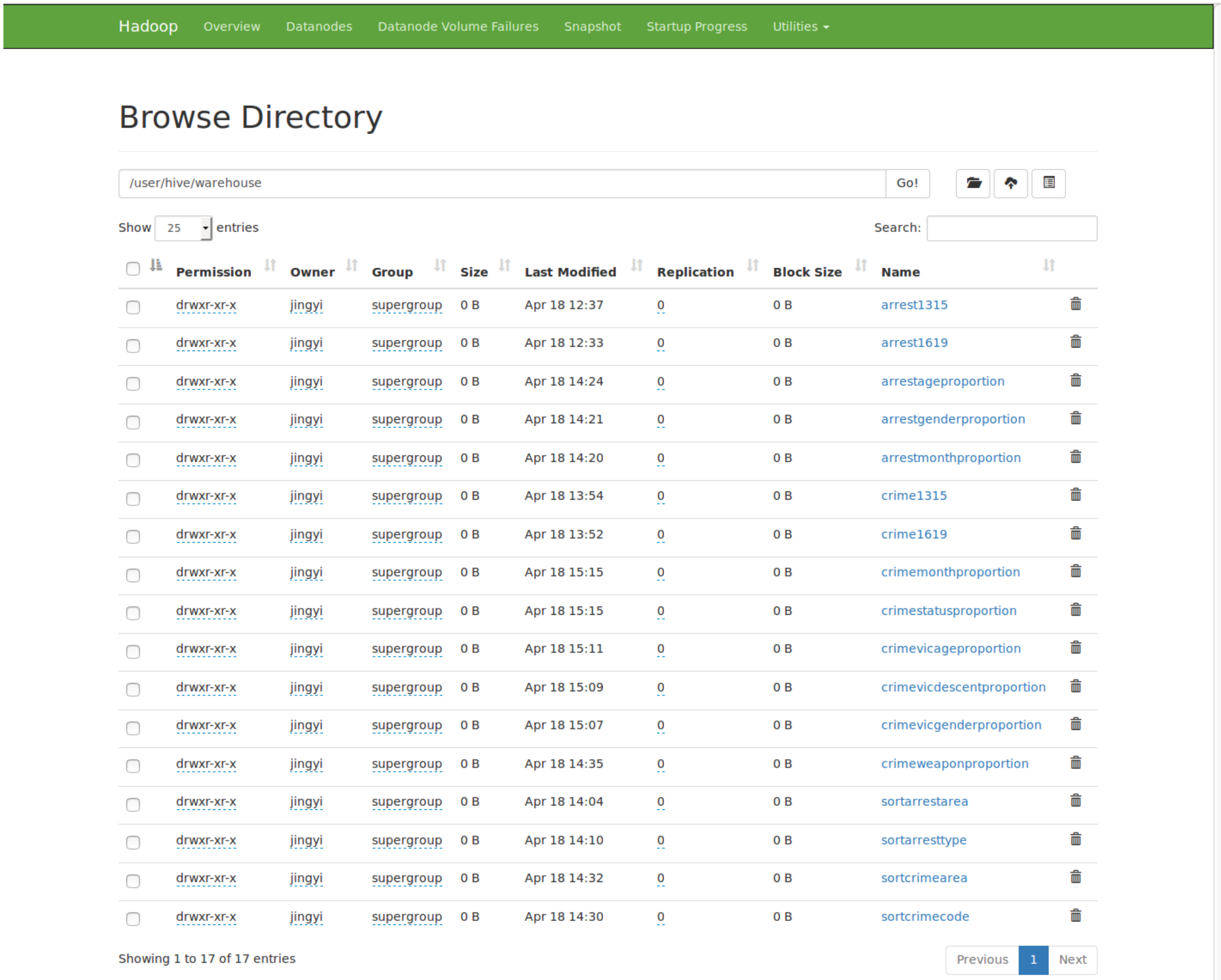
3.6 Use Hive to Store the result of Pig and doing simple query.

Save all the result of Pig to Hive on HDFS so that it will be more convenient to do query since Hive support simple SQL language.

First, using hive-sql language to create table and then load data into the table. Finally test query and doing simple analysis using hive.

Command:

```
CREATE TABLE
LOAD DATA
```

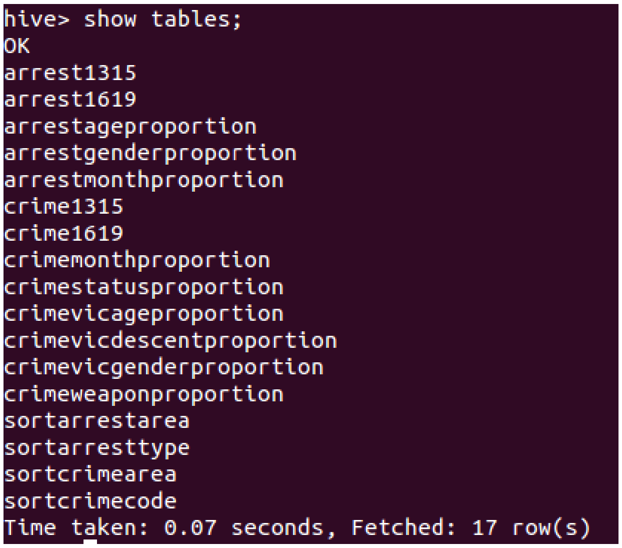


Browse Directory

/user/hive/warehouse

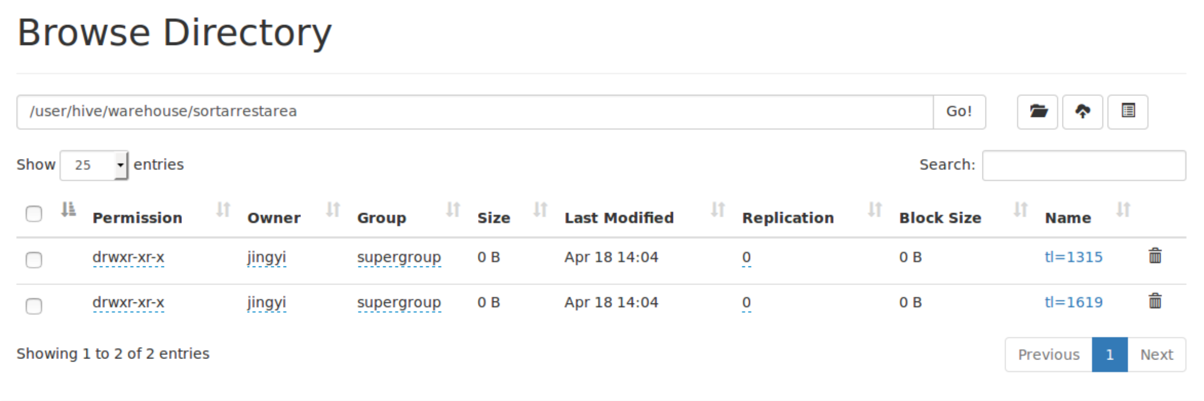
| □ | Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|---|---|---|---|---|---|---|---|---|
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 12:37 | 0 | 0 B | arrest1315 |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 12:33 | 0 | 0 B | arrest1619 |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 14:24 | 0 | 0 B | arrestageproportion |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 14:21 | 0 | 0 B | arrestgenderproportion |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 14:20 | 0 | 0 B | arrestmonthproportion |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 13:54 | 0 | 0 B | crime1315 |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 13:52 | 0 | 0 B | crime1619 |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 15:15 | 0 | 0 B | crimemonthproportion |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 15:15 | 0 | 0 B | crimestatusproportion |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 15:11 | 0 | 0 B | crimevicageproportion |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 15:09 | 0 | 0 B | crimevicdescentproportion |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 15:07 | 0 | 0 B | crimevicgenderproportion |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 14:35 | 0 | 0 B | crimeweaponproportion |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 14:04 | 0 | 0 B | sortarrestarea |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 14:10 | 0 | 0 B | sortarresttype |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 14:32 | 0 | 0 B | sortcrimearea |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 14:30 | 0 | 0 B | sortcrimecode |

Showing 1 to 17 of 17 entries



```
hive> show tables;
OK
arrest1315
arrest1619
arrestageproportion
arrestgenderproportion
arrestmonthproportion
crime1315
crime1619
crimemonthproportion
crimestatusproportion
crimevicageproportion
crimevicdescentproportion
crimevicgenderproportion
crimeweaponproportion
sortarrestarea
sortarresttype
sortcrimearea
sortcrimecode
Time taken: 0.07 seconds, Fetched: 17 row(s)
```

We can also definite partition for tables so that we are able to store data by different groups.



## Browse Directory

/user/hive/warehouse/sortarrestarea Go!

Show 25 entries Search:

| □ | Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name | |
|---|---|---|---|---|---|---|---|---|---|
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 14:04 | 0 | 0 B | tl=1315 | 🗑 |
| □ | drwxr-xr-x | jingyi | supergroup | 0 B | Apr 18 14:04 | 0 | 0 B | tl=1619 | 🗑 |

Showing 1 to 2 of 2 entries

Previous 1 Next

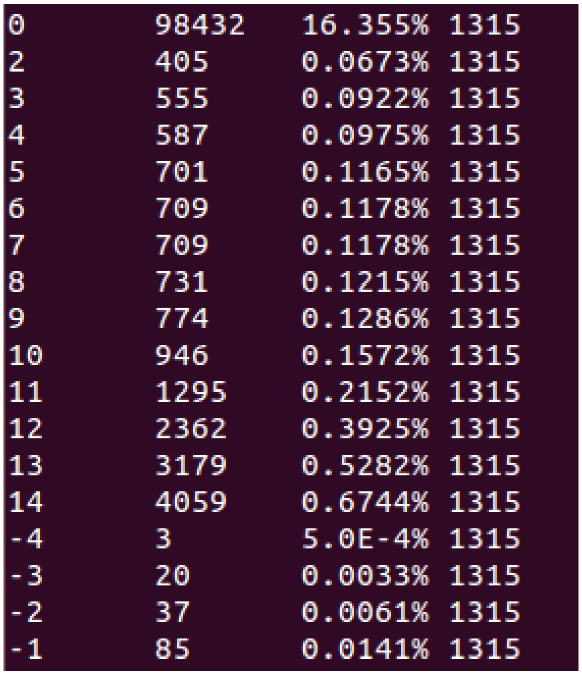
When you use 'LOAD DATA INPATH' command, the data get MOVED (instead of copy) from data location to location that you specified while creating Hive table.

Next, use query to analyze the Age proportion of Arrest and Crime dataset.

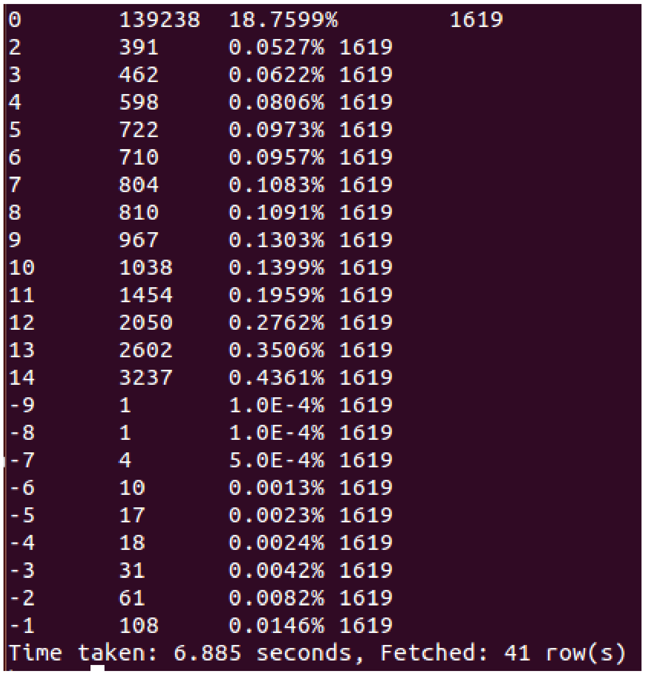
Query 1:

```
SELECT * FROM CrimeVicAgeProportion WHERE Age <= 14;
```

Result:



```
0       98432   16.355% 1315
2       405     0.0673% 1315
3       555     0.0922% 1315
4       587     0.0975% 1315
5       701     0.1165% 1315
6       709     0.1178% 1315
7       709     0.1178% 1315
8       731     0.1215% 1315
9       774     0.1286% 1315
10      946     0.1572% 1315
11      1295    0.2152% 1315
12      2362    0.3925% 1315
13      3179    0.5282% 1315
14      4059    0.6744% 1315
-4      3       5.0E-4% 1315
-3      20      0.0033% 1315
-2      37      0.0061% 1315
-1      85      0.0141% 1315
```

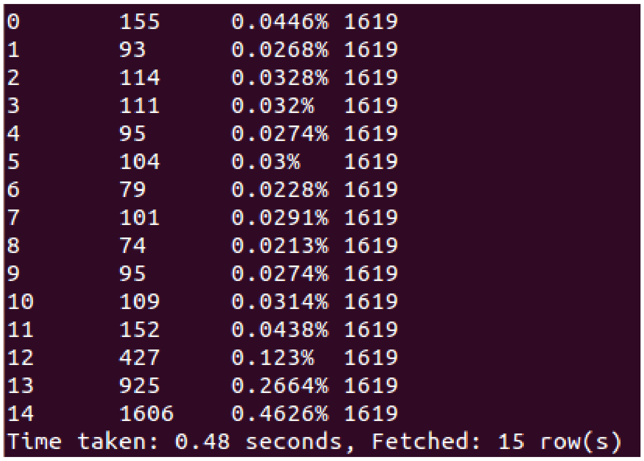


```
0               139238  18.7599%                1619
2               391     0.0527% 1619
3               462     0.0622% 1619
4               598     0.0806% 1619
5               722     0.0973% 1619
6               710     0.0957% 1619
7               804     0.1083% 1619
8               810     0.1091% 1619
9               967     0.1303% 1619
10              1038    0.1399% 1619
11              1454    0.1959% 1619
12              2050    0.2762% 1619
13              2602    0.3506% 1619
14              3237    0.4361% 1619
-9              1       1.0E-4% 1619
-8              1       1.0E-4% 1619
-7              4       5.0E-4% 1619
-6              10      0.0013% 1619
-5              17      0.0023% 1619
-4              18      0.0024% 1619
-3              31      0.0042% 1619
-2              61      0.0082% 1619
-1              108     0.0146% 1619
Time taken: 6.885 seconds, Fetched: 41 row(s)
```

Conclusion: Children under 14 years old are easy to be the victims of crime. More than 15% victims are infants, even pregnant women are possible to become the victim.

Query 2:

```
SELECT * FROM ArrestAgeProportion WHERE Age <= 14 AND tl="1619";
```

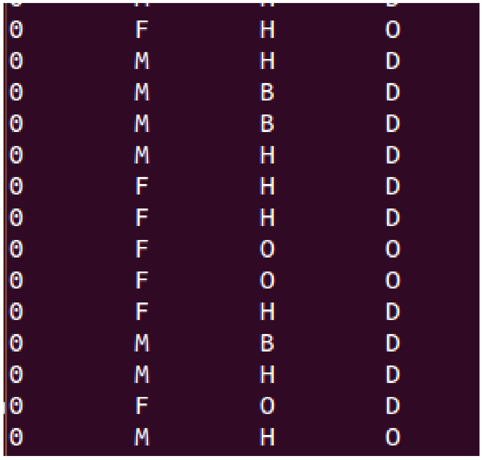


```
0       155     0.0446% 1619
1       93      0.0268% 1619
2       114     0.0328% 1619
3       111     0.032%  1619
4       95      0.0274% 1619
5       104     0.03%   1619
6       79      0.0228% 1619
7       101     0.0291% 1619
8       74      0.0213% 1619
9       95      0.0274% 1619
10      109     0.0314% 1619
11      152     0.0438% 1619
12      427     0.123%  1619
13      925     0.2664% 1619
14      1606    0.4626% 1619
Time taken: 0.48 seconds, Fetched: 15 row(s)
```

There are also lots of children under 14 were arrested during the past several years.

Query 3:

```
SELECT Age, SexCode,DescentCode,ArrestTypeCode FROM arrest1619 WHERE Age == 0;
```



```
0       F       H       O
0       M       H       D
0       M       B       D
0       M       B       D
0       M       H       D
0       F       H       D
0       F       H       D
0       F       O       O
0       F       O       O
0       F       H       D
0       M       B       D
0       M       H       D
0       F       O       D
0       M       H       O
```

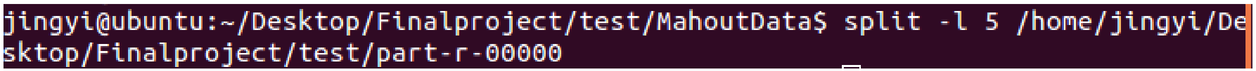
Most of the infants being arrested because of their parents are being arrested.
(ArrestTypeCode D: dependent)

### 3.7 Use Mahout to Clustering

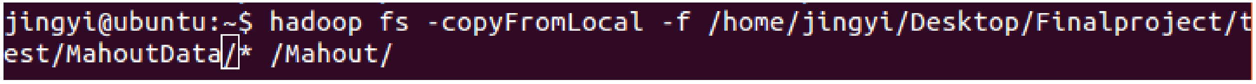
Clustering analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups(clusters). K-means is a simple clustering algorithms.

Data: Weapons being used in crimes.

First, split weapon data into several files by lines.

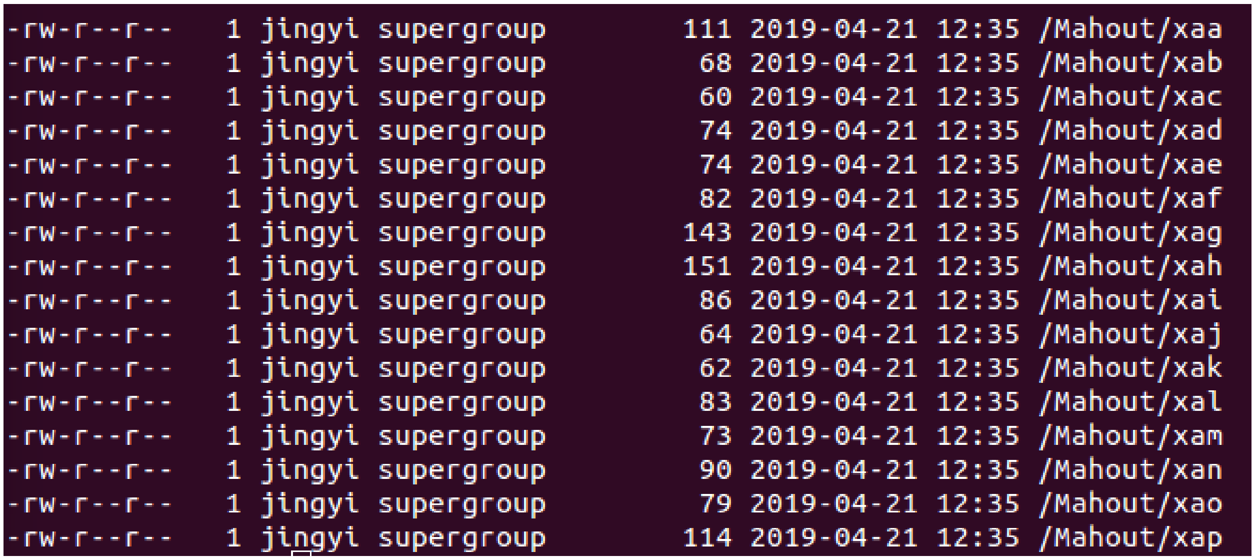


```
jingyi@ubuntu:~/Desktop/Finalproject/test/MahoutData$ split -l 5 /home/jingyi/Desktop/Finalproject/test/part-r-00000
```



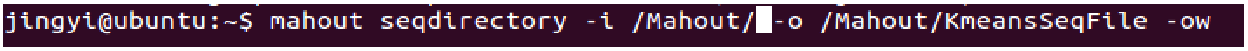
```
jingyi@ubuntu:~$ hadoop fs -copyFromLocal -f /home/jingyi/Desktop/Finalproject/test/MahoutData/* /Mahout/
```

Copy weapon data to HDFS:



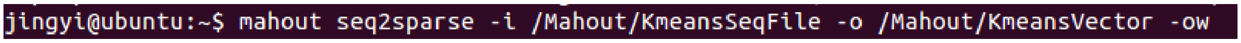
```
-rw-r--r--   1 jingyi supergroup        111 2019-04-21 12:35 /Mahout/xaa
-rw-r--r--   1 jingyi supergroup         68 2019-04-21 12:35 /Mahout/xab
-rw-r--r--   1 jingyi supergroup         60 2019-04-21 12:35 /Mahout/xac
-rw-r--r--   1 jingyi supergroup         74 2019-04-21 12:35 /Mahout/xad
-rw-r--r--   1 jingyi supergroup         74 2019-04-21 12:35 /Mahout/xae
-rw-r--r--   1 jingyi supergroup         82 2019-04-21 12:35 /Mahout/xaf
-rw-r--r--   1 jingyi supergroup        143 2019-04-21 12:35 /Mahout/xag
-rw-r--r--   1 jingyi supergroup        151 2019-04-21 12:35 /Mahout/xah
-rw-r--r--   1 jingyi supergroup         86 2019-04-21 12:35 /Mahout/xai
-rw-r--r--   1 jingyi supergroup         64 2019-04-21 12:35 /Mahout/xaj
-rw-r--r--   1 jingyi supergroup         62 2019-04-21 12:35 /Mahout/xak
-rw-r--r--   1 jingyi supergroup         83 2019-04-21 12:35 /Mahout/xal
-rw-r--r--   1 jingyi supergroup         73 2019-04-21 12:35 /Mahout/xam
-rw-r--r--   1 jingyi supergroup         90 2019-04-21 12:35 /Mahout/xan
-rw-r--r--   1 jingyi supergroup         79 2019-04-21 12:35 /Mahout/xao
-rw-r--r--   1 jingyi supergroup        114 2019-04-21 12:35 /Mahout/xap
```

Then convert data into sequence file using seqdirectory command.



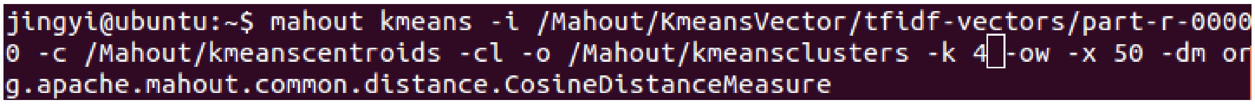
```
jingyi@ubuntu:~$ mahout seqdirectory -i /Mahout/ -o /Mahout/KmeansSeqFile -ow
```

Next convert sequence file to TF-IDF vector using seq2sparse command.



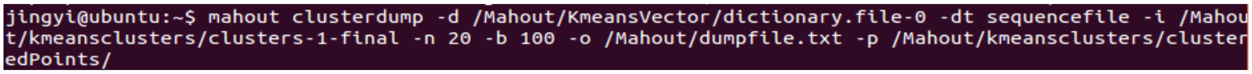
```
jingyi@ubuntu:~$ mahout seq2sparse -i /Mahout/KmeansSeqFile -o /Mahout/KmeansVector -ow
```

Kmean clustering



```
jingyi@ubuntu:~$ mahout kmeans -i /Mahout/KmeansVector/tfidf-vectors/part-r-0000
0 -c /Mahout/kmeanscentroids -cl -o /Mahout/kmeansclusters -k 4 -ow -x 50 -dm or
g.apache.mahout.common.distance.CosineDistanceMeasure
```

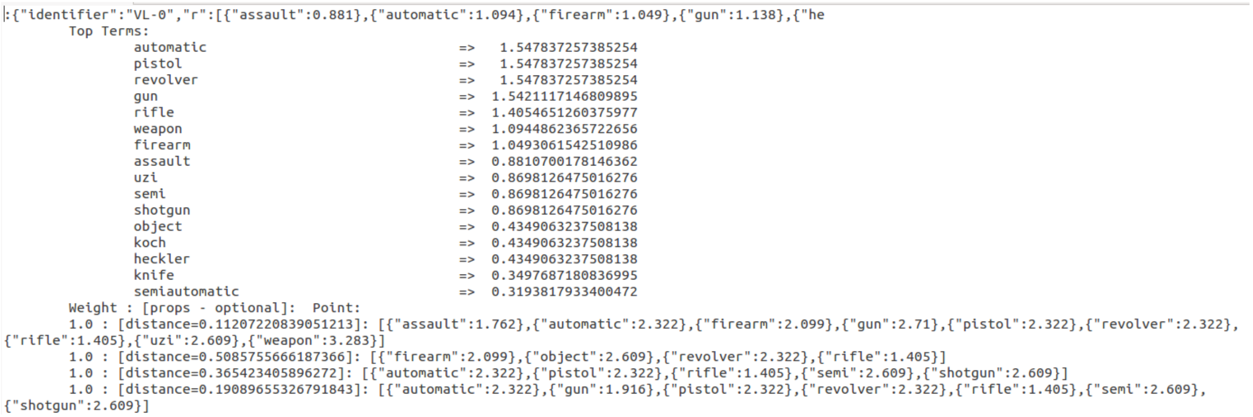
Dump the clusters created into a text file (local file).



```
jingyi@ubuntu:~$ mahout clusterdump -d /Mahout/KmeansVector/dictionary.file-0 -dt sequencefile -i /Mahou
t/kmeansclusters/clusters-1-final -n 20 -b 100 -o /Mahout/dumpfile.txt -p /Mahout/kmeansclusters/cluster
edPoints/
```

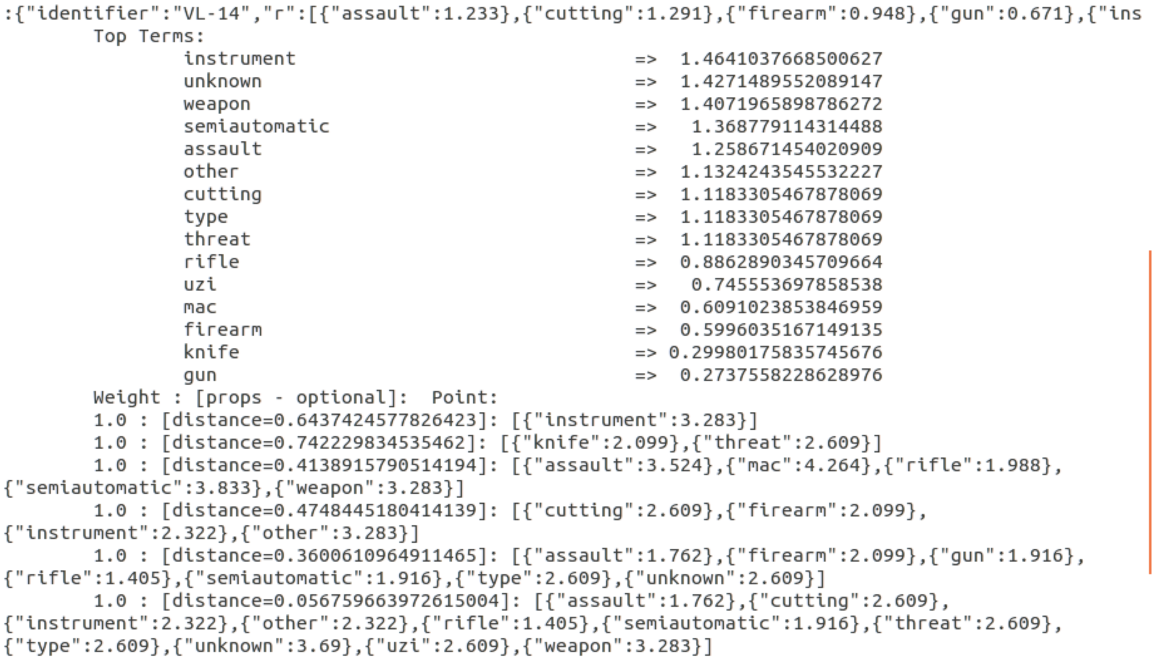
https://mahout.apache.org/users/clustering/cluster-dumper.html

The first cluster:



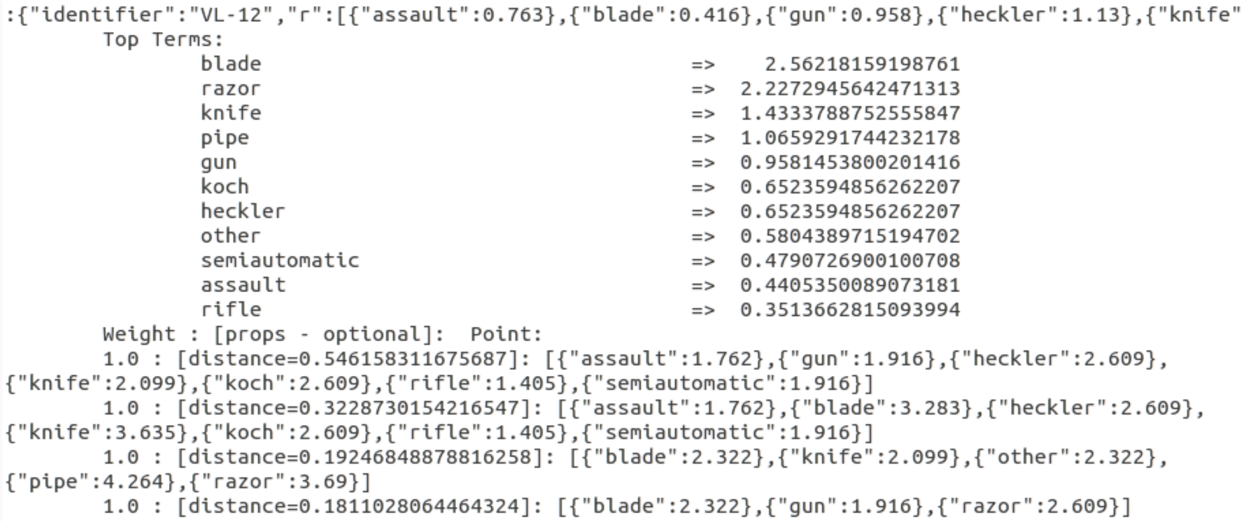
```
:{"identifier":"VL-0","r":[{"assault":0.881},{"automatic":1.094},{"firearm":1.049},{"gun":1.138},{"he
        Top Terms:
                automatic                               =>   1.547837257385254
                pistol                                  =>   1.547837257385254
                revolver                                =>   1.547837257385254
                gun                                     =>  1.5421117146809895
                rifle                                   =>  1.4054651260375977
                weapon                                  =>  1.0944862365722656
                firearm                                 =>  1.0493061542510986
                assault                                 =>  0.8810700178146362
                uzi                                     =>  0.8698126475016276
                semi                                    =>  0.8698126475016276
                shotgun                                 =>  0.8698126475016276
                object                                  =>  0.4349063237508138
                koch                                    =>  0.4349063237508138
                heckler                                 =>  0.4349063237508138
                knife                                   =>  0.3497687180836995
                semiautomatic                           =>  0.3193817933400472
        Weight : [props - optional]:  Point:
        1.0 : [distance=0.11207220839051213]: [{"assault":1.762},{"automatic":2.322},{"firearm":2.099},{"gun":2.71},{"pistol":2.322},{"revolver":2.322},
{"rifle":1.405},{"uzi":2.609},{"weapon":3.283}]
        1.0 : [distance=0.5085755666187366]: [{"firearm":2.099},{"object":2.609},{"revolver":2.322},{"rifle":1.405}]
        1.0 : [distance=0.365423405896272]: [{"automatic":2.322},{"pistol":2.322},{"rifle":1.405},{"semi":2.609},{"shotgun":2.609}]
        1.0 : [distance=0.19089655326791843]: [{"automatic":2.322},{"gun":1.916},{"pistol":2.322},{"revolver":2.322},{"rifle":1.405},{"semi":2.609},
{"shotgun":2.609}]
```

The second cluster:



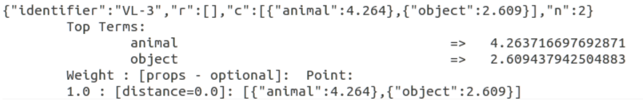
```
:{"identifier":"VL-14","r":[{"assault":1.233},{"cutting":1.291},{"firearm":0.948},{"gun":0.671},{"ins
        Top Terms:
                instrument                              =>  1.4641037668500627
                unknown                                 =>  1.4271489552089147
                weapon                                  =>  1.4071965898786272
                semiautomatic                           =>   1.368779114314488
                assault                                 =>   1.258671454020909
                other                                   =>  1.1324243545532227
                cutting                                 =>  1.1183305467878069
                type                                    =>  1.1183305467878069
                threat                                  =>  1.1183305467878069
                rifle                                   =>  0.8862890345709664
                uzi                                     =>   0.745553697858538
                mac                                     =>  0.6091023853846959
                firearm                                 =>  0.5996035167149135
                knife                                   => 0.29980175835745676
                gun                                     =>  0.2737558228628976
        Weight : [props - optional]:  Point:
        1.0 : [distance=0.6437424577826423]: [{"instrument":3.283}]
        1.0 : [distance=0.742229834535462]: [{"knife":2.099},{"threat":2.609}]
        1.0 : [distance=0.4138915790514194]: [{"assault":3.524},{"mac":4.264},{"rifle":1.988},
{"semiautomatic":3.833},{"weapon":3.283}]
        1.0 : [distance=0.4748445180414139]: [{"cutting":2.609},{"firearm":2.099},
{"instrument":2.322},{"other":3.283}]
        1.0 : [distance=0.3600610964911465]: [{"assault":1.762},{"firearm":2.099},{"gun":1.916},
{"rifle":1.405},{"semiautomatic":1.916},{"type":2.609},{"unknown":2.609}]
        1.0 : [distance=0.056759663972615004]: [{"assault":1.762},{"cutting":2.609},
{"instrument":2.322},{"other":2.322},{"rifle":1.405},{"semiautomatic":1.916},{"threat":2.609},
{"type":2.609},{"unknown":3.69},{"uzi":2.609},{"weapon":3.283}]
```

The third cluster:



```
:{"identifier":"VL-12","r":[{"assault":0.763},{"blade":0.416},{"gun":0.958},{"heckler":1.13},{"knife"
        Top Terms:
                blade                                   =>    2.56218159198761
                razor                                   =>  2.2272945642471313
                knife                                   =>  1.4333788752555847
                pipe                                    =>  1.0659291744232178
                gun                                     =>  0.9581453800201416
                koch                                    =>  0.6523594856262207
                heckler                                 =>  0.6523594856262207
                other                                   =>  0.5804389715194702
                semiautomatic                           =>  0.4790726900100708
                assault                                 =>  0.4405350089073181
                rifle                                   =>  0.3513662815093994
        Weight : [props - optional]:  Point:
        1.0 : [distance=0.546158311675687]: [{"assault":1.762},{"gun":1.916},{"heckler":2.609},
{"knife":2.099},{"koch":2.609},{"rifle":1.405},{"semiautomatic":1.916}]
        1.0 : [distance=0.3228730154216547]: [{"assault":1.762},{"blade":3.283},{"heckler":2.609},
{"knife":3.635},{"koch":2.609},{"rifle":1.405},{"semiautomatic":1.916}]
        1.0 : [distance=0.19246848878816258]: [{"blade":2.322},{"knife":2.099},{"other":2.322},
{"pipe":4.264},{"razor":3.69}]
        1.0 : [distance=0.1811028064464324]: [{"blade":2.322},{"gun":1.916},{"razor":2.609}]
```

The forth cluster:



```
{"identifier":"VL-3","r":[],"c":[{"animal":4.264},{"object":2.609}],"n":2}
        Top Terms:
                animal                                  =>   4.263716697692871
                object                                  =>   2.609437942504883
        Weight : [props - optional]:  Point:
        1.0 : [distance=0.0]: [{"animal":4.264},{"object":2.609}]
```

The deeper meaning and use case of Mahout K-MEANS clustring will be the future work.

# 4. APPENDIX SECTION

## 4.1 MapReduce Java Code

### 4.1.1 MapReduce Filtering Patterns: Distinct Pattern

#### 4.1.1.1 Extract Duplicated data from Arrest dataset

##### 4.1.1.1.1 Distinct Area Arrest

```
package vertical.split;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class DistinctAreaArrest {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "DistinctAreaArrest");
        job.setJarByClass(DistinctAreaArrest.class);
        job.setMapperClass(DistinctAreaArrestMapper.class);
        job.setCombinerClass(DistinctAreaArrestReducer.class);
        job.setReducerClass(DistinctAreaArrestReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(NullWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
    public static class DistinctAreaArrestMapper extends
Mapper<Object,Text,Text,NullWritable>{
        private Text area = new Text();
        public void map(Object key, Text value, Context context) throws
IOException, InterruptedException{
            if(value.toString().contains("Report ID")) {
                return;
            }
            String[] list = value.toString().split(";");
            area.set(list[3]+"\t"+list[4]);
            context.write(area, NullWritable.get());
        }
    }
    public static class DistinctAreaArrestReducer extends
Reducer<Text,NullWritable,Text,NullWritable>{
        public void reduce(Text key, Iterable<NullWritable> values, Context
context) throws IOException, InterruptedException{
            context.write(key,NullWritable.get());
        }
    }
}
```

#### 4.1.1.1.2 Distinct Charge

```
package vertical.split;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;


public class DistinctCharge {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "DistinctCharge");
        job.setJarByClass(DistinctCharge.class);
        job.setMapperClass(DistinctChargeMapper.class);
        job.setCombinerClass(DistinctChargeReducer.class);
        job.setReducerClass(DistinctChargeReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(NullWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
    public static class DistinctChargeMapper extends
Mapper<Object,Text,Text,NullWritable>{
        private Text area = new Text();
        public void map(Object key, Text value, Context context) throws
IOException, InterruptedException{
            if(value.toString().contains("Report ID")) {
                return;
            }
            String[] list = value.toString().split(";");
            area.set(list[12]+"\t"+list[13]);
            context.write(area, NullWritable.get());
        }
    }
    public static class DistinctChargeReducer extends
Reducer<Text,NullWritable,Text,NullWritable>{
        public void reduce(Text key, Iterable<NullWritable> values, Context
context) throws IOException, InterruptedException{
            context.write(key,NullWritable.get());
        }
    }
}
```

4.1.1.1.3 Distinct Charge Group

```
package vertical.split;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class DistinctChargeGroup {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "DistinctChargeGroup");
        job.setJarByClass(DistinctAreaArrest.class);
        job.setMapperClass(DistinctChargeGroupMapper.class);
        job.setCombinerClass(DistinctChargeGroupReducer.class);
        job.setReducerClass(DistinctChargeGroupReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(NullWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
    public static class DistinctChargeGroupMapper extends
Mapper<Object,Text,Text,NullWritable>{
        private Text area = new Text();
        public void map(Object key, Text value, Context context) throws
IOException, InterruptedException{
            if(value.toString().contains("Report ID")) {
                return;
            }
            String[] list = value.toString().split(";");
            area.set(list[9]+"\t"+list[10]);
            context.write(area, NullWritable.get());
        }
    }
    public static class DistinctChargeGroupReducer extends
Reducer<Text,NullWritable,Text,NullWritable>{
        public void reduce(Text key, Iterable<NullWritable> values, Context
context) throws IOException, InterruptedException{
            context.write(key,NullWritable.get());
        }
    }
}
```

#### 4.1.1.2 Extract Duplicated data from Crime dataset

#### 4.1.1.2.1 Distinct Area Crime

```
package vertical.split;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class DistinctAreaCrime {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "DistinctAreaCrime");
        job.setJarByClass(DistinctAreaArrest.class);
        job.setMapperClass(DistinctAreaCrimeMapper.class);
        job.setCombinerClass(DistinctAreaCrimeReducer.class);
        job.setReducerClass(DistinctAreaCrimeReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(NullWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
    public static class DistinctAreaCrimeMapper extends
Mapper<Object,Text,Text,NullWritable>{
        private Text area = new Text();
        public void map(Object key, Text value, Context context) throws
IOException, InterruptedException{
            if(value.toString().contains("DR Number")) {
                return;
            }
            String[] list = value.toString().split(";");
            area.set(list[4]+"\t"+list[5]);
            context.write(area, NullWritable.get());
        }
    }
    public static class DistinctAreaCrimeReducer extends
Reducer<Text,NullWritable,Text,NullWritable>{
        public void reduce(Text key, Iterable<NullWritable> values, Context
context) throws IOException, InterruptedException{
            context.write(key,NullWritable.get());
        }
    }
}
```

#### 4.1.1.2.2 Distinct Crime

```
package vertical.split;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class DistinctCrime {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "DistinctCrime");
        job.setJarByClass(DistinctCrime.class);
        job.setMapperClass(DistinctCrimeMapper.class);
        job.setCombinerClass(DistinctCrimeReducer.class);
        job.setReducerClass(DistinctCrimeReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(NullWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
    public static class DistinctCrimeMapper extends
Mapper<Object,Text,Text,NullWritable>{
        private Text area = new Text();
        public void map(Object key, Text value, Context context) throws
IOException, InterruptedException{
            if(value.toString().contains("DR Number")) {
                return;
            }
            String[] list = value.toString().split(";");
            area.set(list[7]+"\t"+list[8]);
            context.write(area, NullWritable.get());
        }
    }
    public static class DistinctCrimeReducer extends
Reducer<Text,NullWritable,Text,NullWritable>{
        public void reduce(Text key, Iterable<NullWritable> values, Context
context) throws IOException, InterruptedException{
            context.write(key,NullWritable.get());
        }
    }
}
```

4.1.1.2.3 Distinct Premise

```
package vertical.split;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class DistinctPremise {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "DistinctPremise");
        job.setJarByClass(DistinctPremise.class);
        job.setMapperClass(DistinctPremiseMapper.class);
        job.setCombinerClass(DistinctPremiseReducer.class);
        job.setReducerClass(DistinctPremiseReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(NullWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
    public static class DistinctPremiseMapper extends
Mapper<Object,Text,Text,NullWritable>{
        private Text area = new Text();
        public void map(Object key, Text value, Context context) throws
IOException, InterruptedException{
            if(value.toString().contains("DR Number")) {
                return;
            }
            String[] list = value.toString().split(";");
            area.set(list[13]+"\t"+list[14]);
            context.write(area, NullWritable.get());
        }
    }
    public static class DistinctPremiseReducer extends
Reducer<Text,NullWritable,Text,NullWritable>{
        public void reduce(Text key, Iterable<NullWritable> values, Context
context) throws IOException, InterruptedException{
            context.write(key,NullWritable.get());
        }
    }
}
```

#### 4.1.1.2.4 Distinct Status

```
package vertical.split;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class DistinctStatus {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "DistinctStatus");
        job.setJarByClass(DistinctAreaArrest.class);
        job.setMapperClass(DistinctStatusMapper.class);
        job.setCombinerClass(DistinctStatusReducer.class);
        job.setReducerClass(DistinctStatusReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(NullWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
    public static class DistinctStatusMapper extends
Mapper<Object,Text,Text,NullWritable>{
        private Text area = new Text();
        public void map(Object key, Text value, Context context) throws
IOException, InterruptedException{
            if(value.toString().contains("DR Number")) {
                return;
            }
            String[] list = value.toString().split(";");
            area.set(list[17]+"\t"+list[18]);
            context.write(area, NullWritable.get());
        }
    }
    public static class DistinctStatusReducer extends
Reducer<Text,NullWritable,Text,NullWritable>{
        public void reduce(Text key, Iterable<NullWritable> values, Context
context) throws IOException, InterruptedException{
            context.write(key,NullWritable.get());
        }
    }
}
```

4.1.1.2.5 Distinct Weapon Used

```
package vertical.split;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class DistinctWeaponUsed {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "DistinctWeaponUsed");
        job.setJarByClass(DistinctWeaponUsed.class);
        job.setMapperClass(DistinctWeaponUsedMapper.class);
        job.setCombinerClass(DistinctWeaponUsedReducer.class);
        job.setReducerClass(DistinctWeaponUsedReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(NullWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
    public static class DistinctWeaponUsedMapper extends
Mapper<Object,Text,Text,NullWritable>{
        private Text area = new Text();
        public void map(Object key, Text value, Context context) throws
IOException, InterruptedException{
            if(value.toString().contains("DR Number")) {
                return;
            }
            String[] list = value.toString().split(";");
```

```
                        area.set(list[15]+"\t"+list[16]);
                        context.write(area, NullWritable.get());
                    }
                }
                public static class DistinctWeaponUsedReducer extends
        Reducer<Text,NullWritable,Text,NullWritable>{
                    public void reduce(Text key, Iterable<NullWritable> values, Context
        context) throws IOException, InterruptedException{
                        context.write(key,NullWritable.get());
                    }
                }
        }
```

4.1.1.3 Clean Main Data and add YEAR & MONTH features for Arrest dataset

```
package vertical.split;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class CleanArrest {
    public static void main(String[] args) throws Exception {
            Configuration conf = new Configuration();
            Job job = Job.getInstance(conf, "CleanArrest");
            job.setJarByClass(CleanArrest.class);
            job.setMapperClass(CleanArrestMapper.class);
            job.setCombinerClass(CleanArrestReducer.class);
            job.setReducerClass(CleanArrestReducer.class);
            job.setOutputKeyClass(Text.class);
            job.setOutputValueClass(NullWritable.class);
            FileInputFormat.addInputPath(job, new Path(args[0]));
            FileOutputFormat.setOutputPath(job, new Path(args[1]));
            System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
    public static class CleanArrestMapper extends Mapper<Object,Text,Text,NullWritable>{
            private Text area = new Text();
```

```
        public void map(Object key, Text value, Context context) throws IOException,
InterruptedException{
            if(value.toString().contains("Report ID")) {
                return;
            }
            String[] list = value.toString().split(";");
            String y = list[1].substring(0,4);
            String m = list[1].substring(5,7);

    area.set(list[0]+";"+list[1]+";"+y+";"+m+";"+list[3]+";"+list[5]+";"+list[6]+";"+list[7]+";"+li
st[8]+";"+list[9]+";"+list[11]+";"+list[12]+";"+list[14]+";"+list[16]);
            context.write(area, NullWritable.get());
        }
    }
    public static class CleanArrestReducer extends
Reducer<Text,NullWritable,Text,NullWritable>{
        public void reduce(Text key, Iterable<NullWritable> values, Context context)
throws IOException, InterruptedException{
            context.write(key,NullWritable.get());
        }
    }
}
```

4.1.1.4 Clean Main Data and add YEAR & MONTH features for Crime dataset

```
package vertical.split;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class CleanCrime {
    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf, "DistinctAreaCrime");
        job.setJarByClass(DistinctAreaArrest.class);
        job.setMapperClass(CleanCrimeMapper.class);
        job.setCombinerClass(CleanCrimeReducer.class);
        job.setReducerClass(CleanCrimeReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(NullWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
    public static class CleanCrimeMapper extends Mapper<Object,Text,Text,NullWritable>{
        private Text area = new Text();
        public void map(Object key, Text value, Context context) throws IOException,
InterruptedException{
            if(value.toString().contains("DR Number")) {
                return;
            }
            String[] list = value.toString().split(";");
            String y = list[1].substring(0,4);
            String m = list[1].substring(5,7);

    area.set(list[0]+";"+list[1]+";"+y+";"+m+";"+list[2]+";"+list[3]+";"+list[4]+";"+list[6]+";"+li
st[7]+";"+list[10]+";"+list[11]+";"+list[12]+";"+list[13]+";"+list[15]+";"+list[17]+";"+list[20]+";"
+list[23]+";"+list[25]);
            context.write(area, NullWritable.get());
        }
    }
    public static class CleanCrimeReducer extends
Reducer<Text,NullWritable,Text,NullWritable>{
        public void reduce(Text key, Iterable<NullWritable> values, Context context)
throws IOException, InterruptedException{
            context.write(key,NullWritable.get());
        }
    }
}
```

#### 4.1.1.5 Prepare data for Mahout

```
package vertical.split;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;


public class PrepareMahoutData {
  public static void main(String[] args) throws Exception {
      Configuration conf = new Configuration();
      Job job = Job.getInstance(conf, "PrepareMahoutData");
      job.setJarByClass(PrepareMahoutData.class);
      job.setMapperClass(PrepareMahoutDataMapper.class);
      job.setCombinerClass(PrepareMahoutDataReducer.class);
      job.setReducerClass(PrepareMahoutDataReducer.class);
      job.setOutputKeyClass(Text.class);
      job.setOutputValueClass(NullWritable.class);
      FileInputFormat.addInputPath(job, new Path(args[0]));
      FileOutputFormat.setOutputPath(job, new Path(args[1]));
      System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
  public static class PrepareMahoutDataMapper extends
Mapper<Object,Text,Text,NullWritable>{
      private Text area = new Text();
      public void map(Object key, Text value, Context context) throws IOException,
InterruptedException{
          if(value.toString().contains("DR Number")) {
              return;
          }
          String[] list = value.toString().split(";");
          area.set(list[16]);
          context.write(area, NullWritable.get());
      }
  }
  public static class PrepareMahoutDataReducer extends
Reducer<Text,NullWritable,Text,NullWritable>{
      public void reduce(Text key, Iterable<NullWritable> values, Context context)
throws IOException, InterruptedException{
          context.write(key,NullWritable.get());
      }
  }
}
```

### 4.1.2 MapReduce Summarization Patterns: Counter Pattern

#### 4.1.2.1 Count arrest number by Year

```
package finalproject.counter;

import java.io.IOException;
import java.util.Arrays;
import java.util.HashSet;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Counter;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class CounterArrest {
    public static void main(String[] args) throws Exception {
            Configuration conf = new Configuration();
            Job job = Job.getInstance(conf, "CounterArrest");
            job.setJarByClass(CounterArrest.class);
            job.setMapperClass(CounterArrestMapper.class);
            job.setMapOutputKeyClass(NullWritable.class);
            job.setMapOutputValueClass(NullWritable.class);
            FileInputFormat.addInputPath(job, new Path(args[0]));
            FileOutputFormat.setOutputPath(job, new Path(args[1]));
            int code = job.waitForCompletion(true) ? 0 : 1;
            if(code == 0) {
                    for(Counter counter :
job.getCounters().getGroup(CounterArrestMapper.YEAR_COUNTER_GROUP)) {

    System.out.println(counter.getDisplayName()+"\t"+counter.getValue());
                    }
            }
            FileSystem.get(conf).delete(new Path(args[1]),true);
            System.exit(code);
    }
    public static class CounterArrestMapper extends Mapper<Object, Text, NullWritable,
NullWritable>{
            public static final String YEAR_COUNTER_GROUP = "Year";
            public static final String UNKNOWN_COUNTER="Unknown";
            public static final String NULL_OR_EMPTY_COUNTER = "Null or Empty";
            private String[] YEAR = new String[] {

    "2010","2011","2012","2013","2014","2015","2016","2017","2018","2019"
            };
```

```
      private HashSet<String> YEARSet = new HashSet<String>(Arrays.asList(YEAR));
      public void map(Object key, Text value, Context context) throws IOException,
InterruptedException{
          String[] line = value.toString().split(";");
          String year = line[1].substring(0,4);
          if (year != null && !year.isEmpty()) {
              if (YEARSet.contains(year)) {
                  context.getCounter(YEAR_COUNTER_GROUP,
year).increment(1);
              }else {

  context.getCounter(YEAR_COUNTER_GROUP,UNKNOWN_COUNTER).increment(1);
              }
          }else {

  context.getCounter(YEAR_COUNTER_GROUP,NULL_OR_EMPTY_COUNTER).increment(1
);
          }
      }
  }
}
```

### 4.1.2.2 Count crime reported number by Year

```
package finalproject.counter;

import java.io.IOException;
import java.util.Arrays;
import java.util.HashSet;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Counter;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class CounterCrime {
  public static void main(String[] args) throws Exception {
      Configuration conf = new Configuration();
      Job job = Job.getInstance(conf, "CounterCrime");
      job.setJarByClass(CounterCrime.class);
      job.setMapperClass(CounterCrimeMapper.class);
      job.setMapOutputKeyClass(NullWritable.class);
      job.setMapOutputValueClass(NullWritable.class);
      FileInputFormat.addInputPath(job, new Path(args[0]));
      FileOutputFormat.setOutputPath(job, new Path(args[1]));
      int code = job.waitForCompletion(true) ? 0 : 1;
      if(code == 0) {
          for(Counter counter :
job.getCounters().getGroup(CounterCrimeMapper.YEAR_COUNTER_GROUP)) {

  System.out.println(counter.getDisplayName()+"\t"+counter.getValue());
          }
      }
      FileSystem.get(conf).delete(new Path(args[1]),true);
      System.exit(code);
  }
  public static class CounterCrimeMapper extends Mapper<Object, Text, NullWritable,
NullWritable>{
      public static final String YEAR_COUNTER_GROUP = "Year";
      public static final String UNKNOWN_COUNTER="Unknown";
      public static final String NULL_OR_EMPTY_COUNTER = "Null or Empty";
      private String[] YEAR = new String[] {

  "2010","2011","2012","2013","2014","2015","2016","2017","2018","2019"
      };
      private HashSet<String> YEARSet = new HashSet<String>(Arrays.asList(YEAR));
      public void map(Object key, Text value, Context context) throws IOException,
InterruptedException{
          String[] line = value.toString().split(";");
          String year = line[1].substring(0,4);
          if (year != null && !year.isEmpty()) {
              if (YEARSet.contains(year)) {
                  context.getCounter(YEAR_COUNTER_GROUP,
year).increment(1);
              }else {

  context.getCounter(YEAR_COUNTER_GROUP,UNKNOWN_COUNTER).increment(1);
              }
          }else {

  context.getCounter(YEAR_COUNTER_GROUP,NULL_OR_EMPTY_COUNTER).increment(1
);
          }
      }
```

```
    }
}
```

### 4.1.3 MapReduce Organization Patterns: Partitioning Pattern

#### 4.1.3.1 Split Arrest dataset by Year

```
package finalproject.partition;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Partitioner;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class PartiArrest {
    public static void main(String[] args) throws Exception {
            Configuration conf = new Configuration();
            Job job = Job.getInstance(conf, "PartiArrest");
            job.setJarByClass(PartiArrest.class);

            job.setMapperClass(PartiArrestMapper.class);
            job.setReducerClass(PartiArrestReducer.class);
            job.setOutputKeyClass(Text.class);
            job.setOutputValueClass(NullWritable.class);

            job.setPartitionerClass(PartiPartitioner.class);
            job.setNumReduceTasks(3);

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

            System.exit(job.waitForCompletion(true) ? 0 : 1);
            }

            public static class PartiArrestMapper extends Mapper<Object, Text, Text,
NullWritable> {
                    public void map(Object key, Text value, Context context) throws
IOException, InterruptedException {
                            context.write(value,NullWritable.get());
```

```
        }
    }
    public static class PartiPartitioner extends Partitioner<Text, NullWritable> {

        @Override
        public int getPartition(Text key, NullWritable value, int numPartitions) {
            String[] line = key.toString().split(";");
            String y = line[1].substring(0,4);
            int year = Integer.parseInt(y.toString());
            int partition = 0;
            if(numPartitions == 0) {
                partition = 0;
            }
            if(year<2013) {
                partition = 0;
            }
            else if(2013<=year && year<= 2015) {
                partition = 1 % numPartitions;
            }
            else {
                partition = 2 % numPartitions;
            }
            return partition;
        }

    }
    public static class PartiArrestReducer extends Reducer<Text, Text, Text,
NullWritable> {
        public void reduce(Text key, Iterable<Text> values, Context context)
                throws IOException, InterruptedException {
            context.write(key, NullWritable.get());
        }
    }
}
```

4.1.3.2 Split Crime dataset by Year

```
package finalproject.partition;

import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Partitioner;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class PartiCrime {
  public static void main(String[] args) throws Exception {
  Configuration conf = new Configuration();
  Job job = Job.getInstance(conf, "PartiCrime");
  job.setJarByClass(PartiCrime.class);

  job.setMapperClass(PartiCrimeMapper.class);
  job.setReducerClass(PartiCrimeReducer.class);
  job.setOutputKeyClass(Text.class);
  job.setOutputValueClass(NullWritable.class);

  job.setPartitionerClass(PartiPartitioner.class);
  job.setNumReduceTasks(3);

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

  System.exit(job.waitForCompletion(true) ? 0 : 1);
  }

  public static class PartiCrimeMapper extends Mapper<Object, Text, Text, NullWritable> {
      public void map(Object key, Text value, Context context) throws IOException,
InterruptedException {
          context.write(value,NullWritable.get());
      }
  }
  public static class PartiPartitioner extends Partitioner<Text, NullWritable> {

      @Override
      public int getPartition(Text key, NullWritable value, int numPartitions) {
          String[] line = key.toString().split(";");
          String y = line[1].substring(0,4);
          int year = Integer.parseInt(y.toString());
          int partition = 0;
          if(numPartitions == 0) {
              partition = 0;
          }
          if(year<2013) {
              partition = 0;
          }
          else if(2013<=year && year<= 2015) {
              partition = 1 % numPartitions;
          }
          else {
              partition = 2 % numPartitions;
          }
          return partition;
      }

  }
  public static class PartiCrimeReducer extends Reducer<Text, Text, Text, NullWritable> {
      public void reduce(Text key, Iterable<Text> values, Context context)
              throws IOException, InterruptedException {
          context.write(key, NullWritable.get());
      }
  }
}
```

4.2 Pig Script Code

4.2.1 All the script Code for dataset arrest2013-2015

```
otable = LOAD '/FinalProject/PartiArrest/part-r-00001' USING PigStorage(';') AS
(ReportID:long,ArrestDate:chararray,Year:chararray,Month:chararray,AreaID:long,Reporting
District:long,Age:int,SexCode:chararray,DescentCode:chararray,ChargeGroupCode:chararray
,ArrestTypeCode:chararray,Charge:chararray,Address:chararray,Location:chararray);
------------------------------------------------------------------------------------------
/*sorted area by the incidence of Arrest in 2016-2019*/
/*Count Arrest number in each areas 2016-2019.*/
groupArea = GROUP otable BY (AreaID);
count = FOREACH groupArea GENERATE group, COUNT(otable) AS sum;

/*left join with dataset that has areas' detail*/
areadetail = LOAD '/FinalProject/DistinctAreaArrest/part-r-00000' AS
(AreaID:long,AreaName:chararray);
joindata1 = JOIN count BY $0 LEFT OUTER, areadetail BY $0;
joindata = FOREACH joindata1 GENERATE $0,$1,$3;

/*sorted by number*/
sorted = ORDER joindata BY sum DESC;

STORE sorted INTO '/FinalProject/PigOut/SortedArrestArea1315';
------------------------------------------------------------------------------------------
/*sorted Arrest type */
groupType = GROUP otable BY (ArrestTypeCode);
```

```
typeCount = FOREACH groupType GENERATE group, COUNT(otable) AS sum;

sortedType = ORDER typeCount BY sum DESC;
STORE sortedType INTO '/FinalProject/PigOut/SortedArrestType1315';
-------------------------------------------------------------------------------------------
/*proportion of 2 genders being Arrested*/
groupGender = Group otable BY (SexCode);
genderCount = FOREACH groupGender GENERATE group, COUNT(otable) AS sum;
temp = GROUP genderCount ALL;
gendersum = FOREACH temp GENERATE SUM(genderCount.sum) AS total;
temp2 = FOREACH genderCount GENERATE $0,
$1,ROUND_TO((sum/(double)gendersum.total)*100,2) AS perc;
result = FOREACH temp2 GENERATE $0, $1,CONCAT((chararray)perc,'%');

STORE result INTO '/FinalProject/PigOut/ArrestGenderProportion1315';
-------------------------------------------------------------------------------------------
/*month ratio of Arrest*/
groupMonth = Group otable BY (Month);
monthCount = FOREACH groupMonth GENERATE group, COUNT(otable) AS sum;
monthtemp = GROUP monthCount ALL;
monthsum = FOREACH monthtemp GENERATE SUM(monthCount.sum) AS total;
monthtemp2 = FOREACH monthCount GENERATE
$0,$1,ROUND_TO((sum/(double)monthsum.total)*100,2) AS perc;

result = FOREACH monthtemp2 GENERATE $0,$1,CONCAT((chararray)perc,'%');

STORE result INTO '/FinalProject/PigOut/ArrestMonthProportion1315';
-------------------------------------------------------------------------------------------
/*Age ratio of being Arrest*/
groupAge = GROUP otable BY (Age);
ageCount = FOREACH groupAge GENERATE group, COUNT(otable) AS sum;
agetemp = GROUP ageCount ALL;
agesum = FOREACH agetemp GENERATE SUM(ageCount.sum) AS total;
agetemp2 = FOREACH ageCount GENERATE
$0,$1,ROUND_TO((sum/(double)agesum.total)*100,4) AS perc;

result = FOREACH agetemp2 GENERATE $0,$1,CONCAT((chararray)perc,'%');

STORE result INTO '/FinalProject/PigOut/ArrestAgeProportion1315';
```

4.2.2 All the script Code for dataset arrest2016-2019

```
otable = LOAD '/FinalProject/PartiArrest/part-r-00002' USING PigStorage(';') AS
(ReportID:long,ArrestDate:chararray,Year:chararray,Month:chararray,AreaID:long,Reporting
District:long,Age:int,SexCode:chararray,DescentCode:chararray,ChargeGroupCode:chararray
,ArrestTypeCode:chararray,Charge:chararray,Address:chararray,Location:chararray);
-----------------------------------------------------------------------------------------------
/*sorted area by the incidence of Arrest in 2016-2019*/
/*Count Arrest number in each areas 2016-2019.*/
groupArea = GROUP otable BY (AreaID);
count = FOREACH groupArea GENERATE group, COUNT(otable) AS sum;

/*left join with dataset that has areas' detail*/
areadetail = LOAD '/FinalProject/DistinctAreaArrest/part-r-00000' AS
(AreaID:long,AreaName:chararray);
joindata1 = JOIN count BY $0 LEFT OUTER, areadetail BY $0;
joindata = FOREACH joindata1 GENERATE $0,$1,$3;

/*sorted by number*/
sorted = ORDER joindata BY sum DESC;

STORE sorted INTO '/FinalProject/PigOut/SortedArrestArea';
-----------------------------------------------------------------------------------------------
/*sorted Arrest type */
groupType = GROUP otable BY (ArrestTypeCode);
typeCount = FOREACH groupType GENERATE group, COUNT(otable) AS sum;

sortedType = ORDER typeCount BY sum DESC;
STORE sortedType INTO '/FinalProject/PigOut/SortedArrestType';
-----------------------------------------------------------------------------------------------
/*proportion of 2 genders being Arrested*/
groupGender = Group otable BY (SexCode);
genderCount = FOREACH groupGender GENERATE group, COUNT(otable) AS sum;
temp = GROUP genderCount ALL;
gendersum = FOREACH temp GENERATE SUM(genderCount.sum) AS total;
temp2 = FOREACH genderCount GENERATE $0,
$1,ROUND_TO((sum/(double)gendersum.total)*100,2) AS perc;
result = FOREACH temp2 GENERATE $0, $1,CONCAT((chararray)perc,'%');

STORE result INTO '/FinalProject/PigOut/ArrestGenderProportion';
-----------------------------------------------------------------------------------------------
/*month ratio of Arrest*/
groupMonth = Group otable BY (Month);
monthCount = FOREACH groupMonth GENERATE group, COUNT(otable) AS sum;
monthtemp = GROUP monthCount ALL;
monthsum = FOREACH monthtemp GENERATE SUM(monthCount.sum) AS total;
monthtemp2 = FOREACH monthCount GENERATE
$0,$1,ROUND_TO((sum/(double)monthsum.total)*100,2) AS perc;
```

```
result = FOREACH monthtemp2 GENERATE $0,$1,CONCAT((chararray)perc,'%');

STORE result INTO '/FinalProject/PigOut/ArrestMonthProportion';
------------------------------------------------------------------------------------------
/*Age ratio of being Arrest*/
groupAge = GROUP otable BY (Age);
ageCount = FOREACH groupAge GENERATE group, COUNT(otable) AS sum;
agetemp = GROUP ageCount ALL;
agesum = FOREACH agetemp GENERATE SUM(ageCount.sum) AS total;
agetemp2 = FOREACH ageCount GENERATE
$0,$1,ROUND_TO((sum/(double)agesum.total)*100,4) AS perc;

result = FOREACH agetemp2 GENERATE $0,$1,CONCAT((chararray)perc,'%');

STORE result INTO '/FinalProject/PigOut/ArrestAgeProportion';
```

4.2.3 All the script Code for dataset crime2013-2015

```
otable = LOAD '/FinalProject/PartiCrime/part-r-00001' USING PigStorage(';') AS
(DRNumber:long,DateReported:chararray,Year:chararray,Month:chararray,DateOccurred:chararray,TimeOccurred:long,
AreaID:long,ReportingDistrict:long,CrimeCode:long,VictimAge:int,VictimSex:chararray,
VictimDescent:chararray,PremiseCode:long,WeaponUsed:long,StatusCode:chararray,CrimeCode2:long,Address:chararray,Location:chararray);
------------------------------------------------------------------------------------------
/*sorted area by the incidence of Arrest in 2016-2019*/
/*Count Arrest number in each areas 2016-2019.*/
groupArea = GROUP otable BY (AreaID);
count = FOREACH groupArea GENERATE group, COUNT(otable) AS sum;

/*left join with dataset that has areas' detail*/
areadetail = LOAD '/FinalProject/DistinctAreaCrime/part-r-00000' AS
(AreaID:long,AreaName:chararray);
joindata1 = JOIN count BY $0 LEFT OUTER, areadetail BY $0;
joindata = FOREACH joindata1 GENERATE $0,$1,$3;

/*sorted by number*/
sorted = ORDER joindata BY sum DESC;

STORE sorted INTO '/FinalProject/PigOut/SortedCrimeArea1315';
------------------------------------------------------------------------------------------
/*Top 10 crime Types occur most grequently each year*/
crimeCodeDetail = LOAD '/FinalProject/DistinctCrime/part-r-00000' AS
(CrimeCode:long,Describe:chararray);
```

```
groupCrimeCode = GROUP otable BY (Year,CrimeCode);
crimeCount = FOREACH groupCrimeCode GENERATE group.Year,group.CrimeCode,
COUNT(otable) AS sum;

joindata1 = JOIN crimeCount BY $1 LEFT OUTER, crimeCodeDetail BY $0;
joinresult = FOREACH joindata1 GENERATE $0,$1,$2,$4;

groupCountCrime = GROUP joinresult BY $0;
resultCrimeCode = FOREACH groupCountCrime {
    sorted = ORDER joinresult BY $2 DESC;
    lim = LIMIT sorted 10;
    GENERATE FLATTEN(lim);
}

STORE resultCrimeCode INTO '/FinalProject/PigOut/SortedCrimeCode1315';
-----------------------------------------------------------------------------------------
/*month ratio of Crime*/
groupMonth = Group otable BY (Month);
monthCount = FOREACH groupMonth GENERATE group, COUNT(otable) AS sum;
monthtemp = GROUP monthCount ALL;
monthsum = FOREACH monthtemp GENERATE SUM(monthCount.sum) AS total;
monthtemp2 = FOREACH monthCount GENERATE
$0,$1,ROUND_TO((sum/(double)monthsum.total)*100,2) AS perc;

result = FOREACH monthtemp2 GENERATE $0,$1,CONCAT((chararray)perc,'%');

STORE result INTO '/FinalProject/PigOut/CrimeMonthProportion1315';
-----------------------------------------------------------------------------------------
/*Victim Age ratio*/
groupVicAge = Group otable BY (VictimAge);
ageCount = FOREACH groupVicAge GENERATE group, COUNT(otable) AS sum;
agetemp = GROUP ageCount ALL;
agesum = FOREACH agetemp GENERATE SUM(ageCount.sum) AS total;
agetemp2 = FOREACH ageCount GENERATE
$0,$1,ROUND_TO((sum/(double)agesum.total)*100,4) AS perc;

result = FOREACH agetemp2 GENERATE $0,$1,CONCAT((chararray)perc,'%');

STORE result INTO '/FinalProject/PigOut/CrimeVicAgeProportion1315';
-----------------------------------------------------------------------------------------
/*Proportion of victim genders */
groupGender = Group otable BY (VictimSex);
genderCount = FOREACH groupGender GENERATE group, COUNT(otable) AS sum;
temp = GROUP genderCount ALL;
```

```
gendersum = FOREACH temp GENERATE SUM(genderCount.sum) AS total;
temp2 = FOREACH genderCount GENERATE $0,
$1,ROUND_TO((sum/(double)gendersum.total)*100,2) AS perc;
result = FOREACH temp2 GENERATE $0, $1,CONCAT((chararray)perc,'%');

STORE result INTO '/FinalProject/PigOut/CrimeVicGenderProportion1315';

------------------------------------------------------------------------------------------
/*Proportion and detail of Weapons used of crime*/
weapondetail = LOAD '/FinalProject/DistinctWeaponUsed/part-r-00000' AS
(WeaponUsed:long,WeaponDescribed:chararray);
groupWeapon = Group otable BY (WeaponUsed);
WeaponCount = FOREACH groupWeapon GENERATE group, COUNT(otable) AS sum;
Weapontemp = GROUP WeaponCount ALL;
Weaponsum = FOREACH Weapontemp GENERATE SUM(WeaponCount.sum) AS total;
Weapontemp2 = FOREACH WeaponCount GENERATE
$0,$1,ROUND_TO((sum/(double)Weaponsum.total)*100,4) AS perc;
result = FOREACH Weapontemp2 GENERATE $0,$1,CONCAT((chararray)perc,'%');

joindata1 = JOIN result BY $0 LEFT OUTER, weapondetail BY $0;
joindata = FOREACH joindata1 GENERATE $0,$1,$2,$4;

sorted = ORDER joindata BY $1 DESC;
STORE sorted INTO '/FinalProject/PigOut/CrimeWeaponProportion1315';

------------------------------------------------------------------------------------------
/*Different Status proportion of Crime*/
statusdetail = LOAD '/FinalProject/DistinctStatus/part-r-00000' AS
(StatusCode:chararray,StatusDescribed:chararray);
groupStatus = Group otable BY (StatusCode);
StatusCount = FOREACH groupStatus GENERATE group, COUNT(otable) AS sum;
Statustemp = GROUP StatusCount ALL;
Statussum = FOREACH Statustemp GENERATE SUM(StatusCount.sum) AS total;
Statustemp2 = FOREACH StatusCount GENERATE
$0,$1,ROUND_TO((sum/(double)Statussum.total)*100,4) AS perc;
result = FOREACH Statustemp2 GENERATE $0,$1,CONCAT((chararray)perc,'%');

joindata1 = JOIN result BY $0 LEFT OUTER, statusdetail BY $0;
joindata = FOREACH joindata1 GENERATE $0,$1,$2,$4;

sorted = ORDER joindata BY $1 DESC;
STORE sorted INTO '/FinalProject/PigOut/CrimeStatusProportion1315';
------------------------------------------------------------------------------------------
/*Descent*/
```

```
groupDescent = Group otable BY (VictimDescent);
descentCount = FOREACH groupDescent GENERATE group, COUNT(otable) AS sum;
temp = GROUP descentCount ALL;
descentsum = FOREACH temp GENERATE SUM(descentCount.sum) AS total;
temp2 = FOREACH descentCount GENERATE $0,
$1,ROUND_TO((sum/(double)descentsum.total)*100,2) AS perc;
result = FOREACH temp2 GENERATE $0, $1,CONCAT((chararray)perc,'%');

STORE result INTO '/FinalProject/PigOut/CrimeVicDescentProportion1315';
```

4.2.4 All the script Code for dataset crime2016-2019

```
otable = LOAD '/FinalProject/PartiCrime/part-r-00002' USING PigStorage(';') AS
(DRNumber:long,DateReported:chararray,Year:chararray,Month:chararray,DateOccurred:ch
ararray,TimeOccurred:long,
AreaID:long,ReportingDistrict:long,CrimeCode:long,VictimAge:int,VictimSex:chararray,
VictimDescent:chararray,PremiseCode:long,WeaponUsed:long,StatusCode:chararray,CrimeC
ode2:long,Address:chararray,Location:chararray);
------------------------------------------------------------------------------------------
/*sorted area by the incidence of Arrest in 2016-2019*/
/*Count Arrest number in each areas 2016-2019.*/
groupArea = GROUP otable BY (AreaID);
count = FOREACH groupArea GENERATE group, COUNT(otable) AS sum;

/*left join with dataset that has areas' detail*/
areadetail = LOAD '/FinalProject/DistinctAreaCrime/part-r-00000' AS
(AreaID:long,AreaName:chararray);
joindata1 = JOIN count BY $0 LEFT OUTER, areadetail BY $0;
joindata = FOREACH joindata1 GENERATE $0,$1,$3;

/*sorted by number*/
sorted = ORDER joindata BY sum DESC;

STORE sorted INTO '/FinalProject/PigOut/SortedCrimeArea';
------------------------------------------------------------------------------------------
/*Top 10 crime Types occur most grequently each year*/
crimeCodeDetail = LOAD '/FinalProject/DistinctCrime/part-r-00000' AS
(CrimeCode:long,Describe:chararray);
groupCrimeCode = GROUP otable BY (Year,CrimeCode);
crimeCount = FOREACH groupCrimeCode GENERATE group.Year,group.CrimeCode,
COUNT(otable) AS sum;

joindata1 = JOIN crimeCount BY $1 LEFT OUTER, crimeCodeDetail BY $0;
joinresult = FOREACH joindata1 GENERATE $0,$1,$2,$4;
```

```
groupCountCrime = GROUP joinresult BY $0;
resultCrimeCode = FOREACH groupCountCrime {
    sorted = ORDER joinresult BY $2 DESC;
    lim = LIMIT sorted 10;
    GENERATE FLATTEN(lim);
}

STORE resultCrimeCode INTO '/FinalProject/PigOut/SortedCrimeCode';
------------------------------------------------------------------------------------------
/*month ratio of Crime*/
groupMonth = Group otable BY (Month);
monthCount = FOREACH groupMonth GENERATE group, COUNT(otable) AS sum;
monthtemp = GROUP monthCount ALL;
monthsum = FOREACH monthtemp GENERATE SUM(monthCount.sum) AS total;
monthtemp2 = FOREACH monthCount GENERATE
$0,$1,ROUND_TO((sum/(double)monthsum.total)*100,2) AS perc;

result = FOREACH monthtemp2 GENERATE $0,$1,CONCAT((chararray)perc,'%');

STORE result INTO '/FinalProject/PigOut/CrimeMonthProportion';
------------------------------------------------------------------------------------------
/*Victim Age ratio*/
groupVicAge = Group otable BY (VictimAge);
ageCount = FOREACH groupVicAge GENERATE group, COUNT(otable) AS sum;
agetemp = GROUP ageCount ALL;
agesum = FOREACH agetemp GENERATE SUM(ageCount.sum) AS total;
agetemp2 = FOREACH ageCount GENERATE
$0,$1,ROUND_TO((sum/(double)agesum.total)*100,4) AS perc;

result = FOREACH agetemp2 GENERATE $0,$1,CONCAT((chararray)perc,'%');

STORE result INTO '/FinalProject/PigOut/CrimeVicAgeProportion';
------------------------------------------------------------------------------------------
/*Proportion of victim genders */
groupGender = Group otable BY (VictimSex);
genderCount = FOREACH groupGender GENERATE group, COUNT(otable) AS sum;
temp = GROUP genderCount ALL;
gendersum = FOREACH temp GENERATE SUM(genderCount.sum) AS total;
temp2 = FOREACH genderCount GENERATE $0,
$1,ROUND_TO((sum/(double)gendersum.total)*100,2) AS perc;
result = FOREACH temp2 GENERATE $0, $1,CONCAT((chararray)perc,'%');

STORE result INTO '/FinalProject/PigOut/CrimeVicGenderProportion';
```

```
-----------------------------------------------------------------------------------------------
/*Proportion and detail of Weapons used of crime*/
weapondetail = LOAD '/FinalProject/DistinctWeaponUsed/part-r-00000' AS
(WeaponUsed:long,WeaponDescribed:chararray);
groupWeapon = Group otable BY (WeaponUsed);
WeaponCount = FOREACH groupWeapon GENERATE group, COUNT(otable) AS sum;
Weapontemp = GROUP WeaponCount ALL;
Weaponsum = FOREACH Weapontemp GENERATE SUM(WeaponCount.sum) AS total;
Weapontemp2 = FOREACH WeaponCount GENERATE
$0,$1,ROUND_TO((sum/(double)Weaponsum.total)*100,4) AS perc;
result = FOREACH Weapontemp2 GENERATE $0,$1,CONCAT((chararray)perc,'%');

joindata1 = JOIN result BY $0 LEFT OUTER, weapondetail BY $0;
joindata = FOREACH joindata1 GENERATE $0,$1,$2,$4;

sorted = ORDER joindata BY $1 DESC;
STORE sorted INTO '/FinalProject/PigOut/CrimeWeaponProportion';

-----------------------------------------------------------------------------------------------
/*Different Status proportion of Crime*/
statusdetail = LOAD '/FinalProject/DistinctStatus/part-r-00000' AS
(StatusCode:chararray,StatusDescribed:chararray);
groupStatus = Group otable BY (StatusCode);
StatusCount = FOREACH groupStatus GENERATE group, COUNT(otable) AS sum;
Statustemp = GROUP StatusCount ALL;
Statussum = FOREACH Statustemp GENERATE SUM(StatusCount.sum) AS total;
Statustemp2 = FOREACH StatusCount GENERATE
$0,$1,ROUND_TO((sum/(double)Statussum.total)*100,4) AS perc;
result = FOREACH Statustemp2 GENERATE $0,$1,CONCAT((chararray)perc,'%');

joindata1 = JOIN result BY $0 LEFT OUTER, statusdetail BY $0;
joindata = FOREACH joindata1 GENERATE $0,$1,$2,$4;

sorted = ORDER joindata BY $1 DESC;
STORE sorted INTO '/FinalProject/PigOut/CrimeStatusProportion';
-----------------------------------------------------------------------------------------------
/*Descent*/
groupDescent = Group otable BY (VictimDescent);
descentCount = FOREACH groupDescent GENERATE group, COUNT(otable) AS sum;
temp = GROUP descentCount ALL;
descentsum = FOREACH temp GENERATE SUM(descentCount.sum) AS total;
temp2 = FOREACH descentCount GENERATE $0,
$1,ROUND_TO((sum/(double)descentsum.total)*100,2) AS perc;
```

```
result = FOREACH temp2 GENERATE $0, $1,CONCAT((chararray)perc,'%');

STORE result INTO '/FinalProject/PigOut/CrimeVicDescentProportion';
```

### 4.3 Hive SQL Code

#### 4.3.1 All the LOAD DATA CODE

```
CREATE TABLE arrest1619 (ReportID INT,ArrestDate STRING,Year STRING,Month STRING,
AreaID INT,ReportingDistrict INT,Age INT,SexCode STRING,DescentCode STRING,
ChargeGroupCode STRING,ArrestTypeCode STRING,Charge STRING,Address STRING,Location
STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ";"
STORED AS TEXTFILE;

LOAD DATA INPATH "/FinalProject/PartiArrest/part-r-00002" INTO TABLE arrest1619;

CREATE TABLE arrest1315 (ReportID INT,ArrestDate STRING,Year STRING,Month
STRING,AreaID INT,ReportingDistrict INT,Age INT,SexCode STRING,DescentCode
STRING,ChargeGroupCode STRING,ArrestTypeCode STRING,Charge STRING,Address
STRING,Location STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ";"
STORED AS TEXTFILE;

LOAD DATA INPATH "/FinalProject/PartiArrest/part-r-00001" INTO TABLE arrest1315;

CREATE TABLE crime1619 (DRNumber INT,DateReported STRING,Year STRING,Month
STRING,DateOccurred STRING,TimeOccurred INT,AreaID INT,ReportingDistrict
INT,CrimeCode INT,VictimAge INT,VictimSex STRING,VictimDescent STRING,PremiseCode
INT,WeaponUsed INT,StatusCode STRING,CrimeCode2 INT,Address STRING,Location
STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ";"
STORED AS TEXTFILE;

LOAD DATA INPATH "/FinalProject/PartiCrime/part-r-00002" INTO TABLE crime1619;

CREATE TABLE crime1315 (DRNumber INT,DateReported STRING,Year STRING,Month
STRING,DateOccurred STRING,TimeOccurred INT,AreaID INT,ReportingDistrict
INT,CrimeCode INT,VictimAge INT,VictimSex STRING,VictimDescent STRING,PremiseCode
INT,WeaponUsed INT,StatusCode STRING,CrimeCode2 INT,Address STRING,Location
STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ";"
STORED AS TEXTFILE;

LOAD DATA INPATH "/FinalProject/PartiCrime/part-r-00001" INTO TABLE crime1315;

CREATE TABLE sortArrestArea (AreaID INT,AreaCount INT,Name STRING)
PARTITIONED BY (tl STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY "\t";

LOAD DATA INPATH "/FinalProject/PigOut/SortedArrestArea/part-r-00000" INTO TABLE
sortArrestArea PARTITION (tl="1619");
LOAD DATA INPATH "/FinalProject/PigOut/SortedArrestArea1315/part-r-00000" INTO
TABLE sortArrestArea PARTITION (tl="1315");

CREATE TABLE sortArrestType (TypeID STRING,TypeCount INT)
PARTITIONED BY (tl STRING)
```

```
ROW FORMAT DELIMITED FIELDS TERMINATED BY "\t";

LOAD DATA INPATH "/FinalProject/PigOut/SortedArrestType/part-r-00000" INTO TABLE
sortArrestType PARTITION (tl="1619");
LOAD DATA INPATH "/FinalProject/PigOut/SortedArrestType1315/part-r-00000" INTO
TABLE sortArrestType PARTITION (tl="1315");

CREATE TABLE ArrestMonthProportion (Month STRING,MonthCount INT,Perc STRING)
PARTITIONED BY (tl STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY "\t";

LOAD DATA INPATH "/FinalProject/PigOut/ArrestMonthProportion/part-m-00000" INTO
TABLE ArrestMonthProportion PARTITION (tl="1619");
LOAD DATA INPATH "/FinalProject/PigOut/ArrestMonthProportion1315/part-m-00000"
INTO TABLE ArrestMonthProportion PARTITION (tl="1315");

CREATE TABLE ArrestGenderProportion (Gender STRING,GenderCount INT,Perc STRING)
PARTITIONED BY (tl STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY "\t";

LOAD DATA INPATH "/FinalProject/PigOut/ArrestGenderProportion/part-m-00000" INTO
TABLE ArrestGenderProportion PARTITION (tl="1619");
LOAD DATA INPATH "/FinalProject/PigOut/ArrestGenderProportion1315/part-m-00000"
INTO TABLE ArrestGenderProportion PARTITION (tl="1315");

CREATE TABLE ArrestAgeProportion (Age INT,AgeCount INT,Perc STRING)
PARTITIONED BY (tl STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY "\t";

LOAD DATA INPATH "/FinalProject/PigOut/ArrestAgeProportion/part-m-00000" INTO
TABLE ArrestAgeProportion PARTITION (tl="1619");
LOAD DATA INPATH "/FinalProject/PigOut/ArrestAgeProportion1315/part-m-00000" INTO
TABLE ArrestAgeProportion PARTITION (tl="1315");

CREATE TABLE SortCrimeCode (Year String,CrimeCode INT,CrimeCount INT,Detail
STRING)
PARTITIONED BY (tl STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY "\t";

LOAD DATA INPATH "/FinalProject/PigOut/SortedCrimeCode/part-r-00000" INTO TABLE
SortCrimeCode PARTITION (tl="1619");
LOAD DATA INPATH "/FinalProject/PigOut/SortedCrimeCode1315/part-r-00000" INTO
TABLE SortCrimeCode PARTITION (tl="1315");

CREATE TABLE SortCrimeArea (AreaID INT,AreaCount INT,Name STRING)
PARTITIONED BY (tl STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY "\t";

LOAD DATA INPATH "/FinalProject/PigOut/SortedCrimeArea/part-r-00000" INTO TABLE
SortCrimeArea PARTITION (tl="1619");
LOAD DATA INPATH "/FinalProject/PigOut/SortedCrimeArea1315/part-r-00000" INTO
TABLE SortCrimeArea PARTITION (tl="1315");

CREATE TABLE CrimeWeaponProportion (WeaponCode INT,WeaponCount INT,Perc
STRING,detail STRING)
PARTITIONED BY (tl STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY "\t";
```

```
LOAD DATA INPATH "/FinalProject/PigOut/CrimeWeaponProportion/part-r-00000" INTO
TABLE CrimeWeaponProportion PARTITION (tl="1619");
LOAD DATA INPATH "/FinalProject/PigOut/CrimeWeaponProportion1315/part-r-00000"
INTO TABLE CrimeWeaponProportion PARTITION (tl="1315");

CREATE TABLE CrimeVicGenderProportion (Gender STRING,GenderCount INT,Perc STRING)
PARTITIONED BY (tl STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY "\t";

LOAD DATA INPATH "/FinalProject/PigOut/CrimeVicGenderProportion/part-m-00000" INTO
TABLE CrimeVicGenderProportion PARTITION (tl="1619");
LOAD DATA INPATH "/FinalProject/PigOut/CrimeVicGenderProportion1315/part-m-00000"
INTO TABLE CrimeVicGenderProportion PARTITION (tl="1315");

CREATE TABLE CrimeVicDescentProportion (Descent STRING,DescentCount INT,Perc
STRING)
PARTITIONED BY (tl STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY "\t";

LOAD DATA INPATH "/FinalProject/PigOut/CrimeVicDescentProportion/part-m-00000"
INTO TABLE CrimeVicDescentProportion PARTITION (tl="1619");
LOAD DATA INPATH "/FinalProject/PigOut/CrimeVicDescentProportion1315/part-m-00000"
INTO TABLE CrimeVicDescentProportion PARTITION (tl="1315");

CREATE TABLE CrimeVicAgeProportion (Age INT,AgeCount INT,Perc STRING)
PARTITIONED BY (tl STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY "\t";

LOAD DATA INPATH "/FinalProject/PigOut/CrimeVicAgeProportion/part-m-00000" INTO
TABLE CrimeVicAgeProportion PARTITION (tl="1619");
LOAD DATA INPATH "/FinalProject/PigOut/CrimeVicAgeProportion1315/part-m-00000"
INTO TABLE CrimeVicAgeProportion PARTITION (tl="1315");

CREATE TABLE CrimeStatusProportion (Status STRING,StatusCount INT,detail STRING)
PARTITIONED BY (tl STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY "\t";

LOAD DATA INPATH "/FinalProject/PigOut/CrimeStatusProportion/part-r-00000" INTO
TABLE CrimeStatusProportion PARTITION (tl="1619");
LOAD DATA INPATH "/FinalProject/PigOut/CrimeStatusProportion1315/part-r-00000"
INTO TABLE CrimeStatusProportion PARTITION (tl="1315");

CREATE TABLE CrimeMonthProportion (Month INT,MonthCount INT,Perc STRING)
PARTITIONED BY (tl STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY "\t";

LOAD DATA INPATH "/FinalProject/PigOut/CrimeMonthProportion/part-m-00000" INTO
TABLE CrimeMonthProportion PARTITION (tl="1619");
LOAD DATA INPATH "/FinalProject/PigOut/CrimeMonthProportion1315/part-m-00000" INTO
TABLE CrimeMonthProportion PARTITION (tl="1315");
```

### 4.3.2 SQL queries

```
SELECT * FROM CrimeVicAgeProportion WHERE Age <= 14;
SELECT * FROM ArrestAgeProportion WHERE Age <= 14 AND tl="1619";
SELECT * FROM sortArrestType WHERE tl="1619";
SELECT Age, SexCode,DescentCode,ArrestTypeCode FROM arrest1619 WHERE Age == 0 AND
tl = "1619";
```

### 4.4 Mahout command

#### 4.4.1 Split data and store data on HDFS

```
split –l 5 /home/Jingyi/Desktop/Finalproject/test/part-r-00000

hadoop fs –copyFromLocal –f /home/Jingyi/Desktop/Finalproject/test/MahoutData/*
/Mahout/
```

#### 4.4.2 Convert data into sequence file

```
mahout seqdirectory –i /Mahout/ -o /Mahout/KmeanSeqFile -ow
```

#### 4.4.3 Convert data into TF-IDF vector

```
mahout seq2sparse –i /Mahout/KmeansSeqFile –o /Mahout/KmeansVector -ow
```

#### 4.4.4 Kmean clustering

```
mahout kmeans –i /Mahout/KmeansVector/tfidf-vector/part-r-00000 –c
/Mahout/kmeanscentoids –cl –o /Mahout/kmeansclusters –k 4 –ow –x 50 –dm
org.apache.mahout.common.distance.CosineDistanceMeasure
```

#### 4.4.5 Dump the clusters created into a text file

```
mahout clusterdump –d /Mahout/KmeansVector/dictionary.file-0 –dt sequencefile –i
/Mahout/kmeansclusters/clusters-1-final –n 20 –b 100 –o /Mahout/dumpfile.txt –p
/Mahout/kmeansclusters/clusteredPoints/
```