**Assignment 4 – Group 4**

As columns in .csv file are separated by “,”, so in order to avoid of incorrect separation, we first dealt with the raw dataset. We exchanged the “,” in the “Crimes-small.csv” file with “$” sign. Then when we use LOAD “Crimes-small.csv” file USING PigStorage(‘,’), it will be better. But for “Crimes\_2001\_to\_present.csv” we didn’t replace ‘,’ because the dataset is too large. when we try to open it, it just showed part of the data.

1. Which are historically the most dangerous neighborhoods in the City of Chicago, and how dangerous are they?

When it comes to neighborhoods, we used ‘Community Area’ as the unit to be analyzed. Then we define dangerous with two different ways. The first way is that, we consider “Arrest = true”. Another way is considering the total crimes happened in a certain Community Area.

1)

grunt> A = LOAD 'Crimes\_2001\_to\_present.csv' USING PigStorage(',') AS (ID: chararray, CaseNumber: chararray, Date: chararray, Block: chararray, IUCR: int, PrimaryType: chararray, Description: chararray, LocationDescription: chararray, Arrest: charArray, Domestic: chararray, Beat: int, District: int, Ward: int, CommunityArea: int, FBICode: chararray, XCoordinate: double, YCoordinate: double, Year: int);

grunt> B = FILTER A BY Arrest == 'true';

grunt> C = GROUP B BY CommunityArea;

grunt> D = FOREACH C GENERATE group, COUNT(B.Arrest) AS Mycount;

grunt> E = ORDER D BY Mycount DESC;

grunt> DUMP E

The result for the Crimes\_small.csv is stored in “Q1-1-OutputForLargeDataset.txt”. The most dangerous neighborhood is Community 25. And when used small dataset, the number of crimes is 126928. In the large dataset, the output gives a value as “(,178578)”. This could be because the dataset is not pre-processes or there could be null values in community areas.

2)

grunt> A = LOAD 'Crimes\_2001\_to\_present.csv' USING PigStorage(',') AS (ID: chararray, CaseNumber: chararray, Date: chararray, Block: chararray, IUCR: int, PrimaryType: chararray, Description: chararray, LocationDescription: chararray, Arrest: charArray, Domestic: chararray, Beat: int, District: int, Ward: int, CommunityArea: int, FBICode: chararray, XCoordinate: double, YCoordinate: double, Year: int);

grunt> B = GROUP A BY CommunityArea;

grunt> C = FOREACH B GENERATE group, COUNT(A.CommunityArea) AS Mycount; grunt> D = ORDER C BY Mycount DESC;

grunt> DUMP D

The result for the Crimes\_small.csv is stored in “Q1-2-OutputForLargeDataset.txt”. The most dangerous neighborhood is also Community 25. And the number of crimes is 304612.

2. For each crime type, in which months does the crime happen most frequently?

For this question, we first extract “Month” from “Date”

grunt> A = LOAD 'Crimes\_2001\_to\_present.csv’ USING PigStorage(',') AS (ID:chararray,CaseNumber:chararray,Date:chararray,Block:chararray,IUCR:int,PrimaryType:chararray,Description:chararray, LocationDescription:chararray, Arrest:chararray, Domestic:chararray);

grunt> B = FILTER A BY Arrest == 'true' OR Arrest == 'false';

grunt> C = FOREACH B GENERATE PrimaryType,SUBSTRING(Date,0,INDEXOF(Date,'/',0)) AS Month;

grunt> D = GROUP C BY (PrimaryType,Month);

grunt> E = FOREACH D GENERATE FLATTEN($0) AS (PrimaryType, Month),COUNT(C) AS cnt;

grunt> F = GROUP E BY PrimaryType;

grunt> G = FOREACH F GENERATE group, MAX(E.cnt);

grunt> H = JOIN G BY ($0,$1), E BY ($0,$2);

grunt> I = FOREACH H GENERATE $0, $3, $4;

**Output:**

(ARSON,07,902)

(THEFT,07,108150)

(ASSAULT,07,31003)

(BATTERY,07,96213)

(ROBBERY,07,19252)

(BURGLARY,08,30550)

(GAMBLING,08,1914)

(HOMICIDE,07,743)

(STALKING,06,229)

(NARCOTICS,03,55157)

(OBSCENITY,09,24)

(OBSCENITY,10,24)

(OBSCENITY,03,24)

(RITUALISM,09,3)

(RITUALISM,03,3)

(RITUALISM,08,3)

(KIDNAPPING,05,533)

(KIDNAPPING,06,533)

(SEX OFFENSE,07,2047)

(INTIMIDATION,07,294)

(NON-CRIMINAL,05,5)

(PROSTITUTION,01,5807)

(OTHER OFFENSE,03,30272)

(NON - CRIMINAL,08,1)

(OTHER OFFENSE ,06,1)

(OTHER OFFENSE ,02,1)

(OTHER OFFENSE ,04,1)

(OTHER OFFENSE ,03,1)

(OTHER OFFENSE ,07,1)

(CRIMINAL DAMAGE,07,58253)

(PUBLIC INDECENCY,08,15)

(CRIMINAL TRESPASS,08,13683)

(DOMESTIC VIOLENCE,01,1)

(WEAPONS VIOLATION,08,4626)

(DECEPTIVE PRACTICE,08,14812)

(CRIM SEXUAL ASSAULT,07,1861)

(MOTOR VEHICLE THEFT,07,20408)

(LIQUOR LAW VIOLATION,06,1236)

(PUBLIC PEACE VIOLATION,07,3363)

(OTHER NARCOTIC VIOLATION,05,14)

(OFFENSE INVOLVING CHILDREN,01,3322)

(OFFENSES INVOLVING CHILDREN,09,62)

(INTERFERE WITH PUBLIC OFFICER,07,609)

(INTERFERENCE WITH PUBLIC OFFICER,08,388)

(NON-CRIMINAL (SUBJECT SPECIFIED),09,1)

(NON-CRIMINAL (SUBJECT SPECIFIED),12,1)

3. For each crime type, what is the crime growth between the years 2001 and 2012?

In this question, we consider Growth = (Number of crimes in next year – Number of crimes in current year)/Number of crimes in current year. Take 2001 and 2002 as an example, Growth = (crimes in 2002 – crimes in 2001)/(crimes in 2001)

for other years, we just need to change filter conditions

grunt> F = FILTER E BY Year==2001;

grunt> G = FILTER E BY Year==2002;

to ther years.

**Query:**

grunt> A = LOAD 'Crimes\_2001\_to\_present.csv' USING PigStorage(',') AS (ID: chararray, CaseNumber: chararray, Date: chararray, Block: chararray, IUCR: int, PrimaryType: chararray, Description: chararray, LocationDescription: chararray, Arrest: charArray, Domestic: chararray, Beat: int, District: int, Ward: int, CommunityArea: int, FBICode: chararray, XCoordinate: double, YCoordinate: double, Year: int);

grunt> B = FILTER A BY Arrest == 'true' OR Arrest == 'false';

grunt> C = FOREACH B GENERATE PrimaryType,Year;

grunt> D = GROUP C BY (PrimaryType,Year);

grunt> E = FOREACH D GENERATE FLATTEN($0) AS (PrimaryType,Year), COUNT(C) AS cnt;

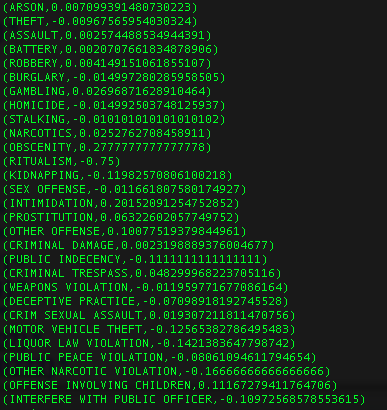
grunt> F = FILTER E BY Year==2001;

grunt> G = FILTER E BY Year==2002;

grunt> H = JOIN F BY $0,G BY $0;

grunt> I = FOREACH H GENERATE $0,(DOUBLE)(((DOUBLE)$5 - (DOUBLE)$2)/(DOUBLE)$2);

**Output:**



4. For each crime type, at what time periods of a day does the crime happen most frequently, and how frequent is it?

For this question, we first extract the Hour form “Date”, and then divided 24 hours into 4 periods: Moring (6 <= Morning < 12), Afternoon (12<= Afternoon < 18), Evening (18<= Evening < 24), and Night (0 <= Night < 6).

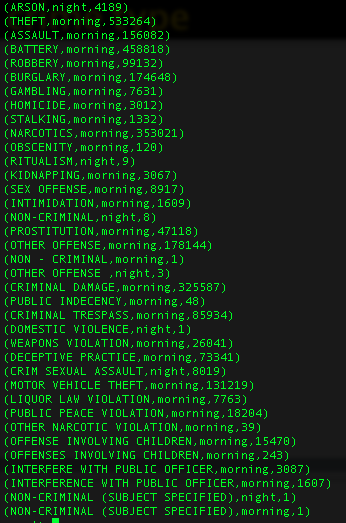
grunt> A = LOAD 'Crimes\_2001\_to\_present.csv' USING PigStorage(',') AS (ID: chararray, CaseNumber: chararray, Date: chararray, Block: chararray, IUCR: int, PrimaryType: chararray, Description: chararray, LocationDescription: chararray, Arrest: charArray, Domestic: chararray, Beat: int, District: int, Ward: int, CommunityArea: int, FBICode: chararray, XCoordinate: double, YCoordinate: double, Year: int);

grunt> B = FILTER A BY Arrest==‘true’ OR Arrest==‘false’; grunt> C = FOREACH B GENERATE PrimaryType,(INT) SUBSTRING(Date, INDEXOF(Date,' ',1),INDEXOF(Date,':',0)) AS Hour;

grunt> D1 = FILTER C BY Hour >= 6 AND Hour < 12; grunt> Morning = FOREACH D1 GENERATE $0, $1, 'morning' AS TimeOfDay; grunt> D2 = FILTER C BY Hour >= 12 AND Hour < 18; grunt> Afternoon = FOREACH D2 GENERATE $0, $1, 'afternoon' AS TimeOfDay; grunt> D3 = FILTER C BY Hour >= 18 AND Hour < 24; grunt> Evening = FOREACH D3 GENERATE $0, $1, 'evening' AS TimeOfDay; grunt> D4 = FILTER C BY Hour >= 0 AND Hour < 6; grunt> Night = FOREACH D4 GENERATE $0, $1, 'night' AS TimeOfDay; grunt> E = UNION Morning, Afternoon, Evening, Night; grunt> F = GROUP E BY (PrimaryType, TimeOfDay); grunt> G = FOREACH F GENERATE FLATTEN($0) AS (PrimaryType, TimeOfDay),COUNT(E) AS cnt;

grunt> H = GROUP G BY PrimaryType; grunt> I = FOREACH H GENERATE group, MAX(G.cnt); grunt> J = JOIN I BY ($0, $1), G BY ($0, $2); grunt> K = FOREACH J GENERATE $0,$3,$4;

**Output**



5. For what crime type is it the most difficult to arrest the criminal?

Difficult to arrest means Arrest == false

grunt> bt = LOAD 'Crimes\_2001\_to\_present.csv' using PigStorage(',') AS (ID:chararray,CaseNumber:chararray,Date:chararray,Block:chararray,IUCR:chararray,PrimaryType:chararray,Description:chararray,LocationDescription:chararray,Arrest:chararray);

grunt> btfAf = FILTER bt BYArrest=='false';

grunt> btfAfgP = GROUP btfAf BY PrimaryType;

grunt> btfAfgPCount = FOREACH btfAfgP GENERATE $0,COUNT(btfAf);

grunt> btgP = GROUP bt BY PrimaryType;

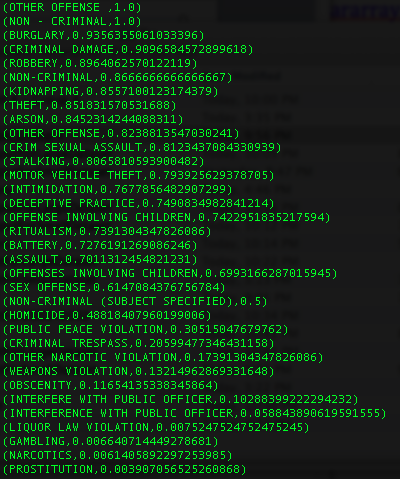
grunt> btgPCount = FOREACH btgP GENERATE $0,COUNT(bt);

grunt> btFinalJOIN = JOIN btgPCount BY $0, btfAfgPCount BY $0;

grunt> btFinal = FOREACH btFinalJOIN GENERATE $0,(DOUBLE)((DOUBLE)$3/(DOUBLE)$1);

grunt> btFinalSorted = ORDER btFinal BY $1 DESC;

**Output**



6. For one of the questions above (your choice), write a user-defined filter (UDF) that gives the same result you obtained before?

We wrote a UDF in java to filter the headers from the dataset. The class name is FilterHeader, and we made a jar file called filter.jar. (find it in the folder)

We used the UDF to answer Q3

Finally we got the same result with Q3

**Query:**

**grunt> REGISTER filter.jar**

grunt> A = LOAD 'Crimes\_2001\_to\_present.csv' USING PigStorage(',') AS (ID: chararray, CaseNumber: chararray, Date: chararray, Block: chararray, IUCR: int, PrimaryType: chararray, Description: chararray, LocationDescription: chararray, Arrest: charArray, Domestic: chararray, Beat: int, District: int, Ward: int, CommunityArea: int, FBICode: chararray, XCoordinate: double, YCoordinate: double, Year: int);

**grunt> B = FILTER A BY FilterHeader(Arrest);**

grunt> C = FOREACH B GENERATE PrimaryType,Year;

grunt> D = GROUP C BY (PrimaryType,Year);

grunt> E = FOREACH D GENERATE FLATTEN($0) AS (PrimaryType,Year), COUNT(C) AS cnt;

grunt> F = FILTER E BY Year==2001;

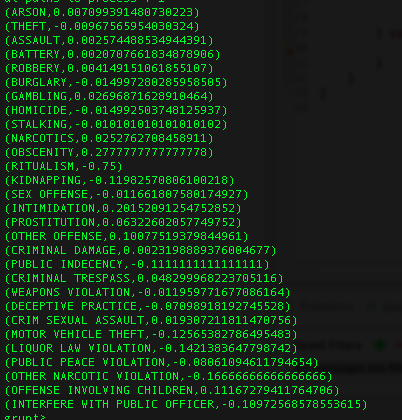
grunt> G = FILTER E BY Year==2002;

grunt> H = JOIN F BY $0,G BY $0;

grunt> I = FOREACH H GENERATE $0,(DOUBLE)(((DOUBLE)$5 - (DOUBLE)$2)/(DOUBLE)$2);

grunt> dump I

**Output**



7. Import the data into either Oracle or MySQL (whichever happens to be on your machine), and answer one of the questions above (your choice). Compare the time taken to answer the query.

We chose the 2rd question, and used MicroSoft SQL Server. We used both the “ Crimes-small.csv” dataset and “Crimes\_2001\_to\_present.csv”

We used SQL Server Integration Services to load the data into the database, and then queried the database.

Time to run the small dataset using SQL Server is almost 0 second, while time using Pig is 57 seconds.

Time to run the large dataset using SQL Server is 1 minute and 15 seconds, while time using Pig is 1 minute and 53 seconds.

According to the analysis, Pig doesn’t show advantage in “time”. But the time comparison may not be so accurate, because the two methods are not ran in the same platform. there might be some errors.

**The query we used:**

SELECT T2.PT2,Mon

FROM

(

(SELECT [Primary Type] PT1,MAX(Cnt1) AS Cnt1

FROM( SELECT [Primary Type], SUBSTRING(Date,0,CHARINDEX('/',Date,0)) AS Mon, COUNT(1) AS Cnt1

FROM [Hadoop\_ChicagoCrime].[dbo].[Chicago\_Crime]

GROUP BY [Primary Type], SUBSTRING(Date,0,CHARINDEX('/',Date,0))) AS T1

GROUP BY [Primary Type])

LEFTOUTER JOIN

(SELECT [Primary Type] PT2, SUBSTRING(Date,0,CHARINDEX('/',Date,0)) AS Mon, COUNT(1) AS Cnt

FROM [Hadoop\_ChicagoCrime].[dbo].[Chicago\_Crime]

GROUP BY [Primary Type], SUBSTRING(Date,0,CHARINDEX('/',Date,0))) AS T2

ON Cnt = Cnt1 AND PT1=PT2

)

ORDER BY PT2

ORDER BY PT2

**Output:**

