CS9233 Final Project Report

## Professor:

Juliana Freire

**Term:**

2014 Fall

## Team Member:

Shaopeng Zhang, N12312768, [sz1076@nyu.edu](mailto:sz1076@nyu.edu) (Question 1, 2, 3, Preparation, Additional)

Hao Chen, N14239927, [hc1351@nyu.edu](mailto:hc1351@nyu.edu) (Question 2, 4, Preparation, Additional)

Guang Xiong, N17366328, [gx239@nyu.edu](mailto:gx239@nyu.edu) (Question 1, Preparation)

## Presentation Link:

<http://prezi.com/rupla2sv7zqg/?utm_campaign=share&utm_medium=copy>

## Introduction:

We choose **the topic of “Understanding taxi economics”**, first we collect all the data we need, such as the fare data, trip data, weather data and income data. Then we decide to investigate the 4 questions provided under this topic. We will complete the first preparation stage, including which tools we are going to use, what data sets will be used and so on before Nov 17th, and also we will do the first question and play with it. At the second stage, we will do all the problems provided and prepare for the short presentation about the partial results. At last, we should come up with more problems related to this topic and solve them,and prepare for the final presentation.

## Some links:

The S3 Bucket of our Project is:

[zcxfinalproject](https://s3.amazonaws.com/zcxfinalproject/)

The github url is:

[**https://github.com/zsp1987/cs9233\_final\_project**](https://github.com/zsp1987/cs9233_final_project)

## Technologies in the project:

Analysis:

* hadoop streaming with python,
* pig(for additional research)

Visualization:

* google chart
* Excel (only for additional part)

# Part I PREPARATION:

Below are the problems and issues that we encountered during the preparation stage:

1. Data File Access: usually we open csv files through Excel, but since the data file is so large, Excel will exit immediately after we open the file. We solve the problem by using shell command ‘head’ and only list the top 10 line: “head -n 1000 > sample.data”, which is very useful when doing some practice on local machine.
2. Location: the only data fields related to location in the csv files are the latitude and longitude. However we want to find which boroughs each record belongs to. One way we could do is through some geolocation API such as Google Maps API. The problem is that google charge for it and limit the frequency that we can query per day. 2500 per day is of course not suitable for this project. So we decide instead using third party api, we implement the helper Class ourselves. The basic idea is to find the boundary of five boroughs. The northeast and southwest latitude and longitude. If the date fail into the range of the boundary, we roughly think the the trip start with in the geolocation boundary are belong to that borough.

(**Updated: following instructor’s suggestion, we found the right way to solve this problem is using the shape file from** [**zillow**](http://www.zillow.com/howto/api/neighborhood-boundaries.htm)**, and determine the latitude and longitude within the polygon.)**

1. Join: because trip and fare are separate files, but to get the useful information. We must join two files by some common columns. An obvious notice is that both files have medallion, hack\_license, vendor\_id, pickup\_datetime. So these columns could be a bridge connecting two files. Join is an easy step in relational database. However, in hadoop, this need some special technical. the method we using is read both files, but after deciding the columns we need. If some columns miss on one file, we will fill those columns with -1. The idea behind this is that after map process, hadoop will sort data and -1 will always appear at front. And we can distinguish which is which by the position of these -1.
2. Neighbor Income Information: on New York Census website, the data we found is the income median of each zip code area. However to divide them by neighbor, we must find some one-to-one mapping between zip and neighbor. The only information we find is by googling those key word and find a chart from nyc [health department](https://www.health.ny.gov/statistics/cancer/registry/appendix/neighborhoods.htm). One problem is that the neighbor in this chart is a little different from the Zillow shapefile one. So we can not get an exact comparison between NY Census data and data generated by our MapReduce job but approximation.

# Part II IMPLEMENTATION:

## Question 1

## How does a taxi driver’s revenue vary across neighborhoods and how does it correlate with the median household income in the neighborhood?

#### Explanation:

This question should be the most difficult one, since it combines geolocation information from taxi data and household income from NY census, which we accomplished it by two Mapreduce jobs.

MapReduce Job 1:

We want to extract information from joined results of both the taxi fare and trip data. So we choose hacked license as the key because this field appears in both data and we interested in a driver’s revenue and we distinguish two kinds of file by checking whether the length of each line is 14.

The mapper outputs (drv\_lcn, neighbor, fare). In trip file, we get neighbor by indexing pickup geolocation to RTree of New York shapefile. While in fare file, we output field of total\_amount. And we fill up those missing columns of both files by -1 for later use.

The reducer outputs (neighbor, revenue). By checking the position of ‘-1’ we know where the record comes from. We add all the total\_amount of each driver to get the revenue. And also, for each driver, we choose the neighbor with maximal frequency.

MapReduce Job 2:

The second MapReduce job simply groups revenues by its key neighbor and get the median of the revenue list and output (neighbor, median\_driver\_revenue).

#### Some Issues:

After running the MapReduce jobs on hadoop, we found there are a lot of total revenue less than $5000, as figure 1 shown below, which is very unlikely be a normal person’s year income. One reason may be those drivers are only work as part time.

So we revised our second MapReduce job by checking whether the total revenue is more than $5000. If not, the result will not be considered as a valid candidate data.

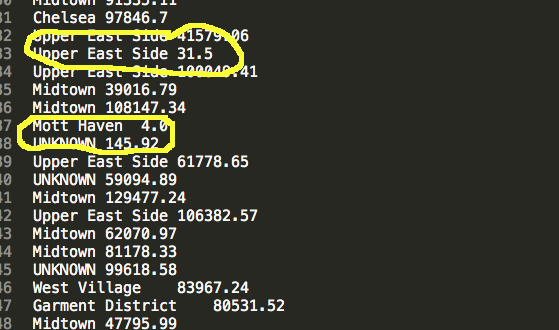


figure 1. Partial Results of Second MapReduce Job

#### Results and Analysis:

Top 3: Taxi Driver Revenue Median:

|  |  |
| --- | --- |
| Borough Park | 101040.36 |
| Williamsburg | 86978.18 |
| Southeast Bronx | 72885.01 |

Top 3: NY Census Household Income Median:

|  |  |
| --- | --- |
| Financial District | 112947 |
| Upper West Side | 78066 |
| Gramercy Park and Murray Hill | 69273 |

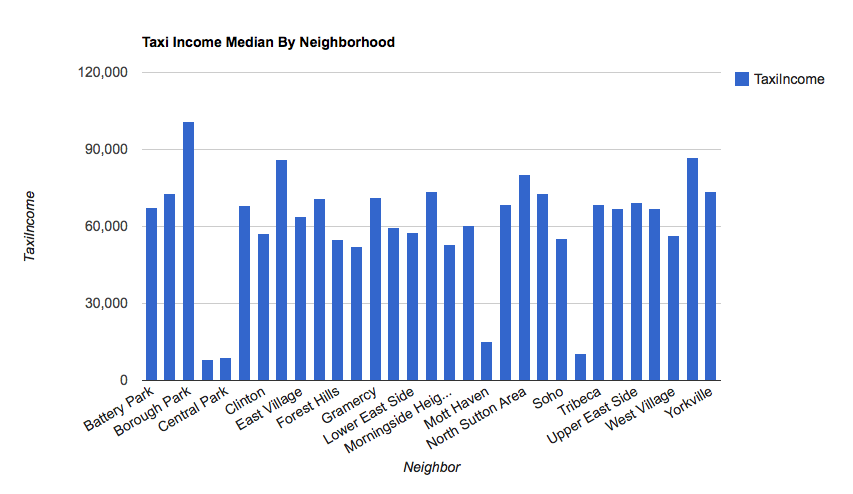


Figure 2a. Taxi Driver Revenue Median by Neighbor

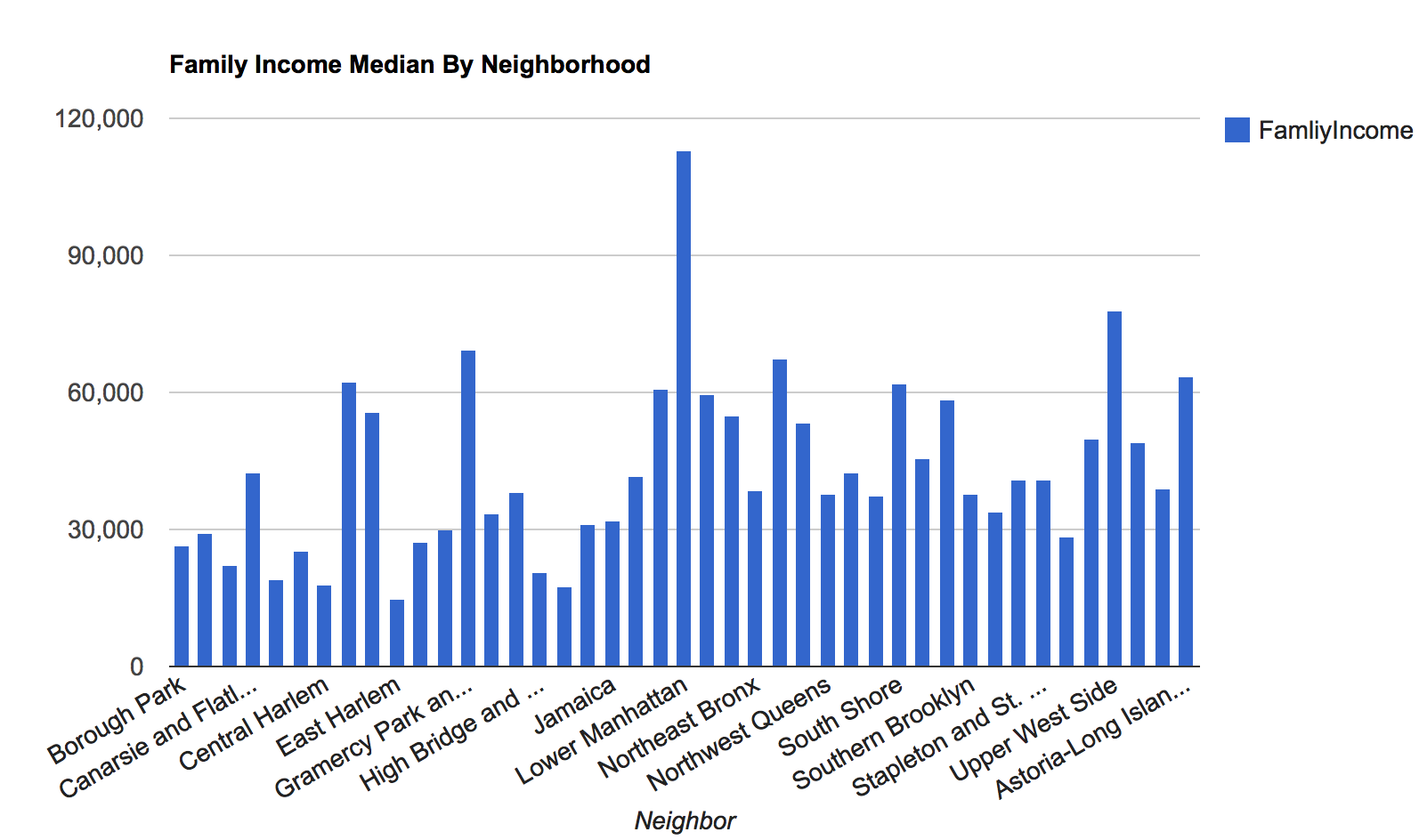


Figure 2b. Household Income Median by Neighbor

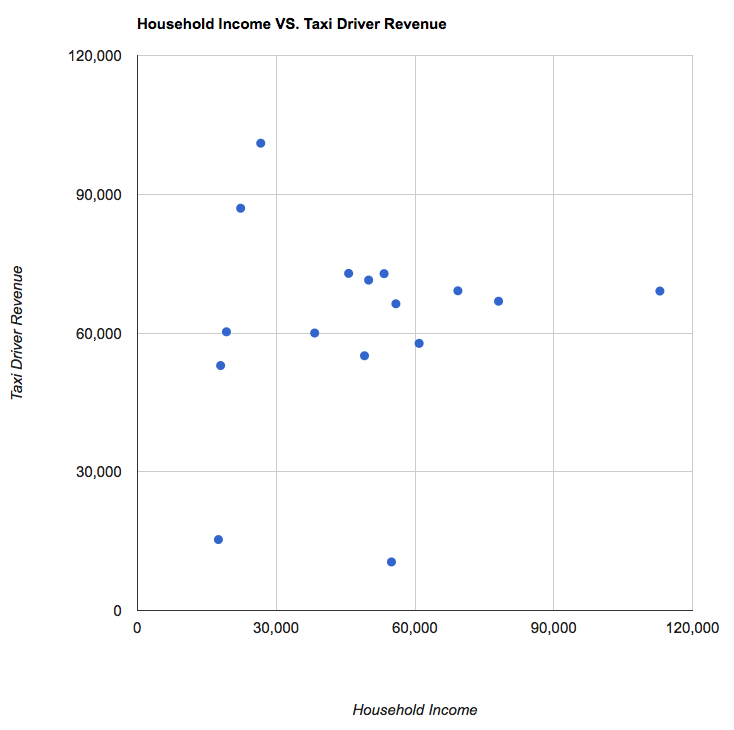


Figure 2c. Taxi Driver Revenue Median by Household Income Median

1. Figure 2c. shows that taxi driver revenue does not correlate with household income revenue, the variance of it is quite small
2. the Average of taxi driver is 60,701. While census income median only have average 34,374, which is much less than the taxi driver. So Taxi driver may be consider as a relative high salary job.
3. Compared to NY Census Household Income, the variance of taxi driver revenue is flatter. Most of them have a similar year income within a certain range. And all of them are around their average value.

#### Running Environment and Performance:

(All AMI in this project are using 3.3.1 if not mentions.)

- First MapReduce:

Hadoop streaming job with RTree bootstrap.

* Master: 1 \* m1.medium
* Core: 4 \* m1.medium
* Arguments: -files s3://zcxfinalproject/src/q1map.py,s3://zcxfinalproject/src/q1rd.py,s3://zcxfinalproject/src/shapefile.py,s3://zcxfinalproject/src/ZillowNeighborhoods-NY.prj,s3://zcxfinalproject/src/ZillowNeighborhoods-NY.dbf,s3://zcxfinalproject/src/ZillowNeighborhoods-NY.shp,s3://zcxfinalproject/src/ZillowNeighborhoods-NY.shp.xml,s3://zcxfinalproject/src/ZillowNeighborhoods-NY.shx -mapper q1map.py -reducer q1rd.py -input s3://zcxfinalproject/input/\* -output s3://zcxfinalproject/output/q1j1
* Running Time: 3 hours, 53 minutes

- Second MapReduce:

* Master: 1 \* m1.medium
* Core: 2 \* m1.medium
* Arguments: -files s3://zcxfinalproject/src/q1map2.py,s3://zcxfinalproject/src/q1rd2a.py -mapper q1map2.py -reducer q1rd2a.py -input s3://zcxfinalproject/output/q1j1/part-\* -output s3://zcxfinalproject/output/q1j2a
* Running Time: 11 minutes.

## 

## 

## 

## 

## Question 2

## How does revenue vary over time? Are the months or seasons when taxi companies make more (or less) money?

#### Explanation:

This question the easiest one among four questions. And we really start doing our project with this one as the stepping stone. The simple idea behind this question is to get the summarization of total amount that is group by time.

This time interval can be day, month or others. So we did them both because they could be the intermediated value for other questions. (like question 4)

Running environment:

* Master: m1.large
* Core: m3.xlarge ( This is the first MapReduce we run on the Amazon, so we used the default setting and test the speed).
* Augment: hadoop jar /home/hadoop/contrib/streaming/hadoop-streaming.jar -files s3://zcxfinalproject/src/q2map.py,s3://zcxfinalproject/src/q2rd.py -mapper q2map.py -reducer q2rd.py -input s3://zcxfinalproject/input/trip\_fare\_\* -output s3://zcxfinalproject/output/q2
* Running Time: 43 minutes

Result:

1. for each month, the total amount taxi companies makes are around 200,000,000.
2. Among all the Months, March has the highest value 230,038214 dollars, August have the lowest value 190610276.901 dollars.
3. Among four quarters, the second quarter have the highest values and third quarter have the lowest value.
4. people are more likely to take taxi in spring and fall than summer and winter.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 206177887.411 |  | 1st Quarter | 629242157.384 |
| 2 | 196346621.371 |  | 2nd Quarter | 668507212.036 |
| 3 | 226717648.602 |  | 3rd Quarter | 609631987.663 |
| 4 | 221818042.802 |  | 4th Quarter | 653964005.314 |
| 5 | 230038214.932 |  |  |  |
| 6 | 216650954.302 |  |  |  |
| 7 | 204031309.181 |  |  |  |
| 8 | 190610276.901 |  |  |  |
| 9 | 214990401.581 |  |  |  |
| 10 | 227890393.162 |  |  |  |
| 11 | 214744950.821 |  |  |  |
| 12 | 211328661.331 |  |  |  |

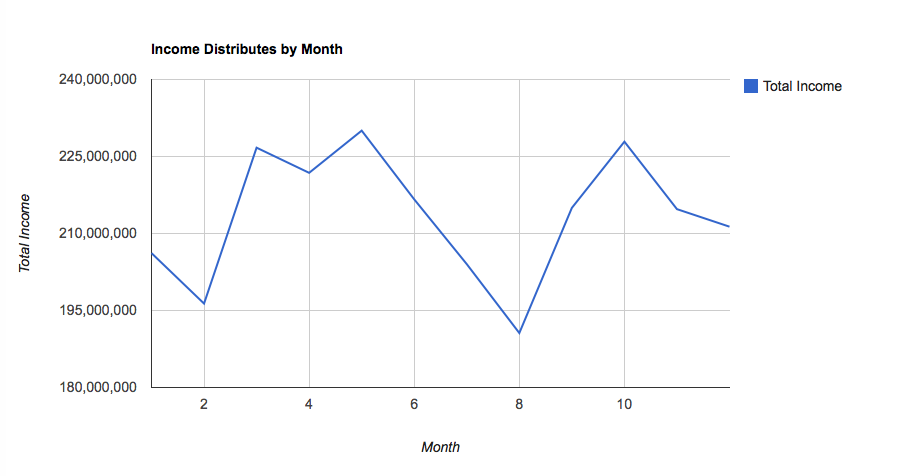


Figure 3.

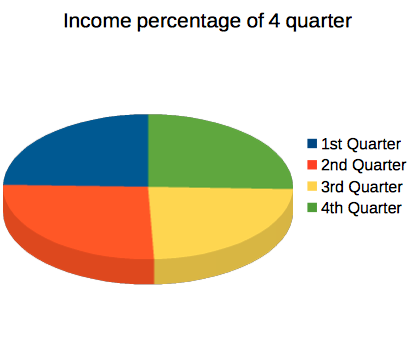


Figure 4.

## **Question 3**

## How long do cab drives ride without passengers? How does this vary over time?

Explanation:

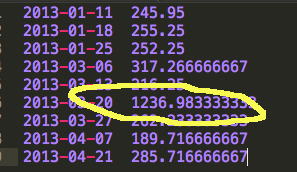
After reading this question, we know that we need information of driver, which is hacked license. and because it also ask to discover the information vary along with time. So pickup\_time should also be included. To calculate the time that a driver drive without passenger. We need two MapReduce to achieve the object.

MapReduce 1:

During Map phase, for each input value, we get its medallion, hacked\_license, and pick\_date(without time) and concate them as the key value. In additional, we output each pair of pickup and drop off time. Because in the reduce phase, the middle result is sorted by key. Therefore we can calculated the total time driver without passenger by add up the time difference between each pickup time and its previous drop off timestamp.

**Interesting notice:**

**By check out the output of first mapreduce, we notice some result have over 720 minutes(About 12 hours).**



This data is very suspicious, because nobody will drive a taxi without passengers over 12 hours a day(half of a day). Otherwise this driver will lose a lot money and must be very unhappy(business is so bad!).

So we check the data, find the reason cause this may due to In some adjacent “pickup-dropoff” pair, the first appear in the morning and the second appears at night. So this driver is very likely be one who work during the night.

Revised MapReduce 1:

So we revised our code a little bit in reduce process. Simply add a restriction that limit the time interval between each ‘pickup-dropoff’ must less than 6 hours. otherwise the interval is consider as driver off-work time.

MapReduce 2:

MapReduce 2 is pretty straight ward. Because question as how data vary by time. So we group up them by day, and calculate the sum of them.

Running environment:

- MapReduce 1:

* Master: 1 \* m1.medium
* Core: 4 \* m1.large
* Arguments: -files s3://zcxfinalproject/src/q3map.py,s3://zcxfinalproject/src/q3rd.py -mapper q3map.py -reducer q3rd.py -input s3://zcxfinalproject/input/trip\_data\* -output s3://zcxfinalproject/output/q3j1
* Action on failure: Continue
* Running Time: 3 hours, 29 minutes

- MapReduce 2:

* Master: 1 \* m1.medium
* Core: 2 \* m1.medium
* Arguements: -files s3://zcxfinalproject/src/q3map2.py,s3://zcxfinalproject/src/q3rd2.py -mapper q3map2.py -reducer q3rd2.py -input s3://zcxfinalproject/output/q3j1/part-\* -output s3://zcxfinalproject
* Running Time: 8 minutes

Result:

Surprised to us, We thought the time would be a very randomized data. However, in the result seem data is distributed within some area.

1. From figure 5, we can see for each driver, most of them have an average without passenger time are 250 minutes(around 4 hours) per day and 6,000,000 minutes per year.
2. May and october have the lowest average time. Meanwhile, combine with figure 5, we found that this two month also have a high total income.

So Can we guess that driver will earn more when they have less without passenger period.

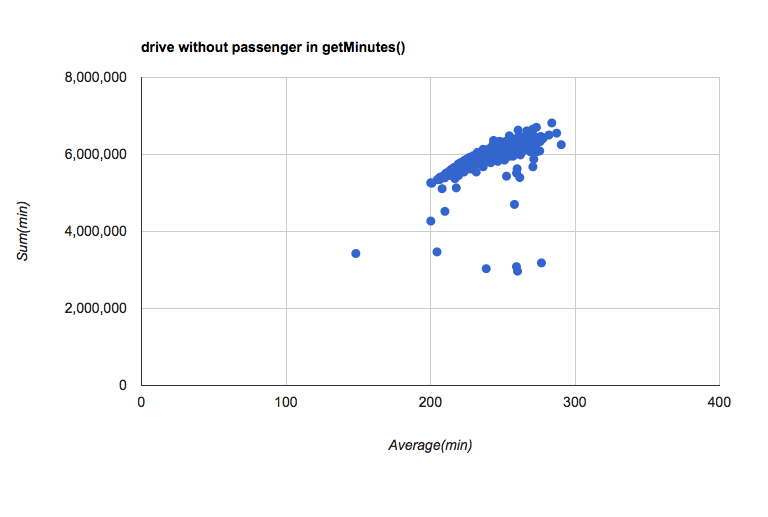


Figure 5

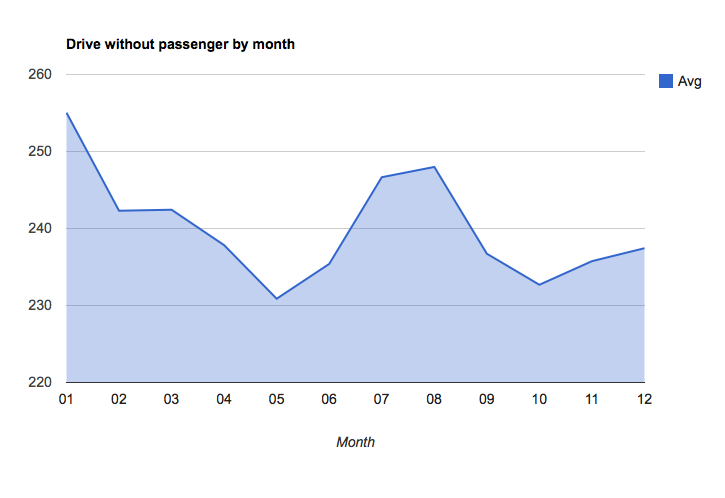


Figure 6.

## **Question 4**

## **Are revenues affected during major events? E.g., parades, presidential visits, storms**

Explanation:

This question actually is not about how to use mapreduce to handle large data,it focus more on the results that the data reveals,and for our specific question,it is how the events affect the taxi revenues.

So at first,we just write a mapreduce to produces the certain result of some input events with its date.And after that,we realize there is no need to run the mapreduce every time to find the data about the event.The easiest way is to just run the mapreduce once and use the results to analyse the event.So at last,we only write one mapreduce to get the result of the revenue for each day in 2013.

There are two methods we try to analyse this data:

(a)what event will cause the revenue increase?

I choose to use the additional weather data to see whether the rain will increase the revenue.In common sense,when rain comes,people who don’t have umbrella will take the taxi,and this should increase the revenue.

The weather data is such a problem for us,because a lot of databases do not have the certain csv format,the one which provides this csv format,QCLCD,always inaccessible when we try to get inside,don’t know why,but it always shows the inner error on the website.The weather data is small,and to manipulate this data set,there is no need to use mapreduce,just read all the files and grab the date and precipitation of water fields,make them into the format,which can be used into the google graph.

Then we check to see if the rainfall really affects the taxi revenues.

After examine the normal rainfall,we start to looking for the Tropical-Storm happened in 2013,whose name is Andrea.The date shown in Wiki is June 07,08.And we just grab that period of time of data and shown in the graph to analyse.

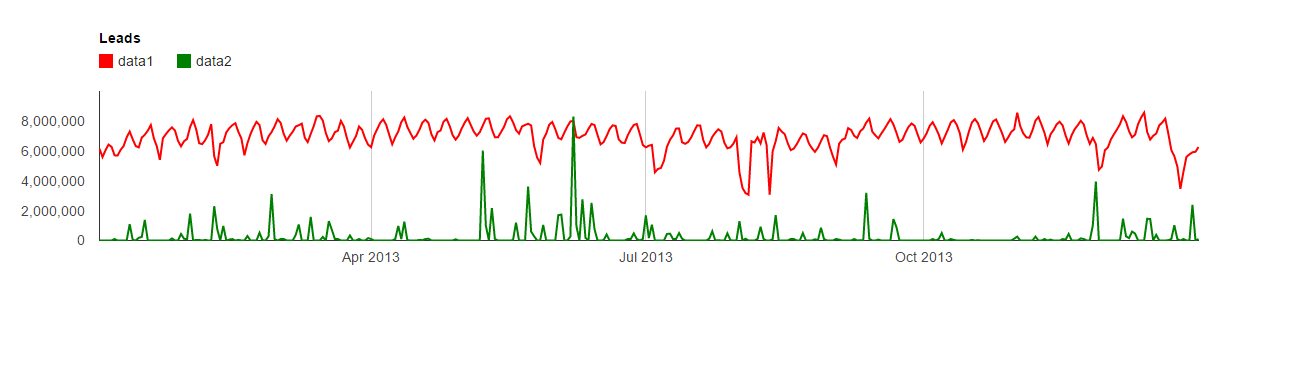
(b)what event will cause the revenue decrease?

We choose to look for the holidays,which we think they will affect the revenues.So the simplest and fastest method is to make a anaotation graph for each day in 2013,and anaotate the important points shown in the graph.

Then we try to check whether the weekends will have less revenues than the weekdays.We use the two line graph to represent them,every week has one average value of weekdays and one average value of weekends.

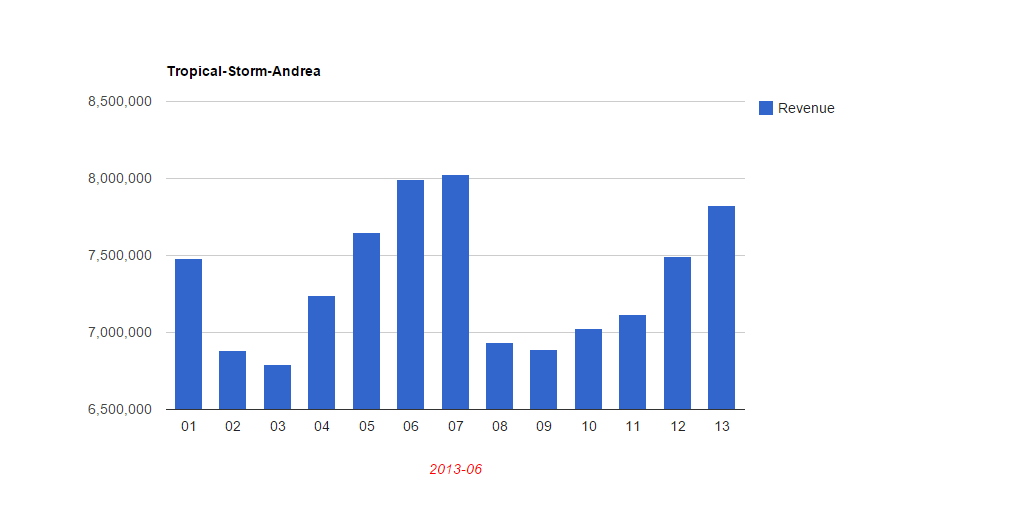
Result:

For the how rain affects the revenues:



From this graph,we just find a disappointing fact,which is it seems that the rain does not have such a huge effect to the taxi revenue,there is no certain relation about the rain and the revenue.So we try to find why this will happen,and then we realize there are subsititute ways for people to use when there comes raining,people can still use subways,buses,in this case,we will prefer to take subway if we are not in a hurry.So rain is not a big factor for the revenue,but we think it must have impact to the revenue,so we try to analyse the Tropical-Storm-Andrea(which is the highest point of rainfall in the above graph).

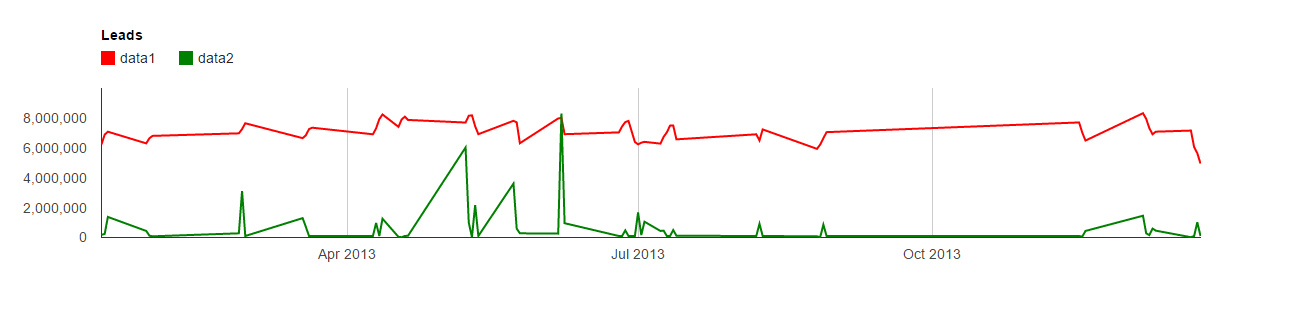
For the Tropical-Storm-Andrea:



From this graph,we can get a lot of information.First we know that the storm is on 07 and 08,there is a huge revenue decrease between them.We can know that in the weather data,the year’s highest rainfall is on 07th,and the rainfall is larger on 08th than on 06th,but there is huge decrease between 06 and 08,which indicates the heavy rain in 07th does impact a lot of the revenue.The reason will be people prefer to stay at home than go out for the storm,so on 08th,when the storm was still there,fewer people took taxis.The reason why there is a high revenue on 07th is that all the traffic,including taxi,subway,bus are fully used to avoid the storm,and the storm causes some flooding on the street,makes the next few days in a low level revenue until the road is clear.

From this result,I can get that the heavy rain(storm) actually does affect the revenue in a decreasing way,but this pattern is not shown from the first graph,so I assume there is still some conditions about the rain I am missing.When to analyse this graph,maybe the duration of the rain is the missing part.

For the further investigation:



From the graph,we choose some data which the rain lasts at least 3 days,and this time,it reveals the fact we are supposed to find out.Although this graph is a bit of unobvious,because some data are not shown in this graph,only the points which has the rainfall shown.So we don’t need to consider the part whcih is a smooth line.

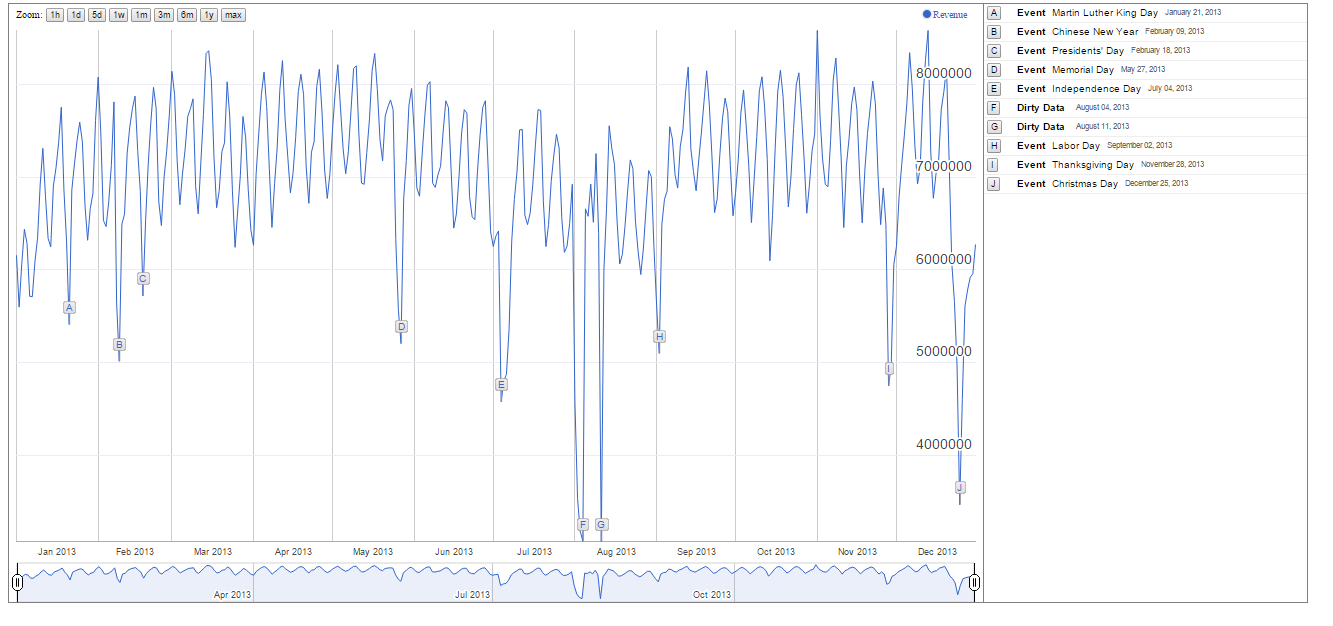
The important facts we can get are:

(a)There are three points which are the highest points of the rainfall shows when the rainfall comes to a certain level of amount,it will decrease the taxi revenue.We can find the corresponding revenues decrease after this point.

(b)When the rain lasts at least three days,and the rain is not so heavy,the revenue will increase during this period.Almost every part,the rainfall under certain level shows this relation.

So the result will be that rainfall is not the major factor affects the taxi revenue,but it still influences the revenue under certain characteristics:when the rain is heavy enough,it decreases the revenue,when the rain is not so heavy,but it last long,at least three days,it will increase the revenue.

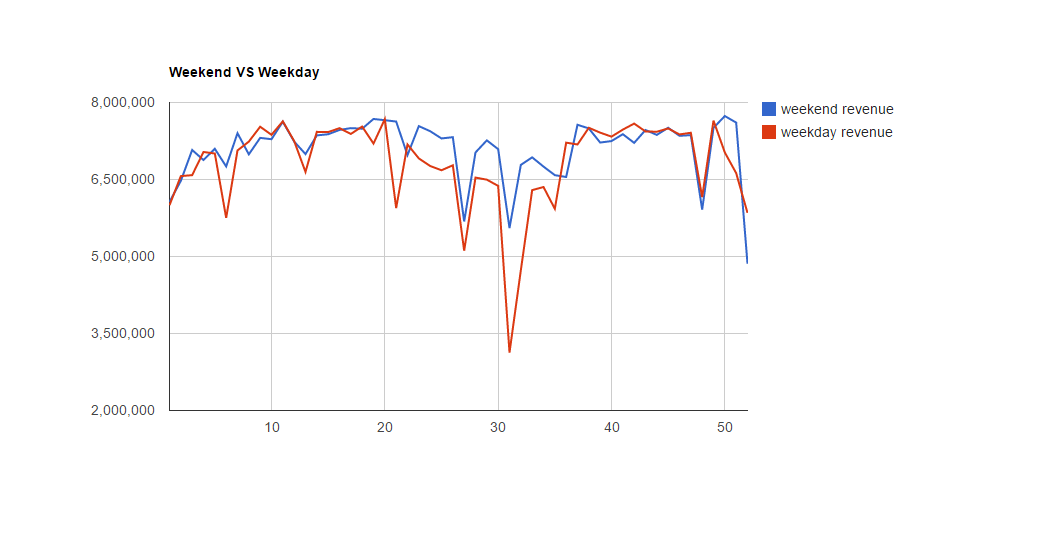
For the year’s revenue:



This is an annotation graph,which can be annotated and can be zoomed.This is the most interesting and direct part.This blue line shows the revenue of each day in 2013.It is obvious that we interested in a few of points that in a low level below 6000000.They are holidays,especially,they are the national wide holidays.People do not work,so they don’t need the taxi so much,so the revenue is low,also maybe some of the taxi drivers enjoy the holidays as well.But we can find there is only one holiday is not national,but it still decreases the revenue a lot.It is the Chinese New Year,Feb 09,2013,this maybe because the large amount of Chinese people living in New York,also indicate that Chinese people contributes a lot to the New York taxi revenue.Another interesting fact from this graph is that we can find out some patterns about the effect.There are two parts look the same in the graph,one is at E point,and the other is at I point.(if it is not clear in the graph,you can use the annotation.html in github,in the graph directory).

The only thing we find about this pattern is that they are both holidays on Thursday,so this make us to think whether if the weekends will affect the revenue,because weekends are also holidays for a lot of people.

For the weekdays VS weekends:



We use the average of the weekdays and average of the weekends to represent these two lines.The index shows it is the Nth week in the year.

This graph does not show what we want to find,it seems there is no such a big difference between the average of the weekdays and the average of the weekends.It may expain why there is such a pattern,the two lines act similiarly at that time period.But we can still get something,we can combine these two graphs,and find that it will take almost a week to make the revenue erases the impact of the holiday.Things are not changed so quickly in reality,the impact of the holiday will continue at least a week.We can find the continuing decreasing of the revenue a few days before the holiday,showing that people also prepare to welcome the holiday.

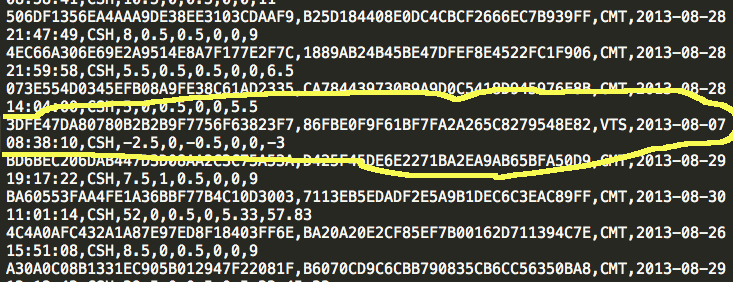
PART III MORE:

During the project, some data appears significantly be wrong. During the project, we just ignore them as dirty data because we need tolerant these data.

But after we finish our questions, we think these data are interesting and may tell us some truth behind the story. So we do a little bit research with them.

Observation

1. The first kind of wrong data is the total income is negative,



We really don’t know what cause this. Should it be that taxi company own passenger money? There is no way to explain such data.

1. Another kind data is that when we calculate

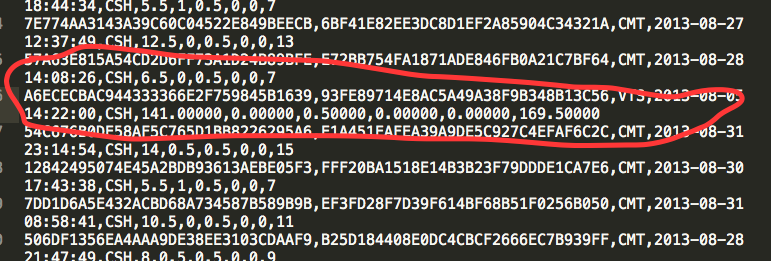
**total\_amount - tolls\_amount - tip\_amount - mta\_tax - surcharge - fare\_amount,**

instead **0**, the result is **some positive value**.

like the blow one is 169.5 - 0 -0 - 0.5 - 0 -141 = 28.

This must means something. An easy explanation is that driver overcharge customer money.

So we decide to find out these “Bad Driver”



## Section 1. Negative Income

Explanation:

After tried python stream, we decided use some different technologies to implement our logic. Pig is very simple if you are familiar with SQL and quickly to implement.

Step we solve this problem is following:

1. read data.
2. clean up the first line.
3. filter data by some restrictions.
4. generate new data by select the field that we need and add some facility columns.
5. group the data by what we are interesting.
6. output the final result.

The steps is pretty straight forward. So in our code, we add filter in step 3 for checking the if the total amount is negative. And only keep that line if so.

In steps 4, we will add a columns only have value 1 for counting purpose.

In steps 5, we decide that we need group either by Date or By drivers.

Running environment:

(with pig 0.12.0)

* Master: 1\* m1.medium
* Core: 3 \* m1.medium
* Arguments: s3://elasticmapreduce/libs/pig/pig-script --run-pig-script --pig-versions 0.12.0 --args -f s3://zcxfinalproject/src/q5j2.pig -p INPUT=s3://zcxfinalproject/input/\* -p OUTPUT=s3://zcxfinalproject/output/
* Running Time: 1 hour, 2 minutes

Result

1. I**t’s pretty interesting that all negative income happens at August and almost happens every day. So is that means during this month taxi system have some problems**
2. Compare to the result of Question 2, August is also the month that taxi company has lowest total income. And these dirty data obviously have some effect on that result.
3. hack license start with AA77 have the highest frequency which is 64 times. Ofter driver appear have problem less often.

## Top three:

By Date

|  |  |  |
| --- | --- | --- |
| 8/11/13 | 173 | -1809.46 |
| 8/30/13 | 158 | -1916 |
| 8/22/13 | 157 | -1481.98 |

By Drivers

|  |  |  |
| --- | --- | --- |
| AA77C06C2C6FAA2B3823E3D691C42F95 | 64 | -686 |
| DA77E53CAC669E67B42EA3430A9D4E36 | 37 | -196 |
| E6CA8455605FB3AB7A5C164E56BE174A | 26 | -150 |

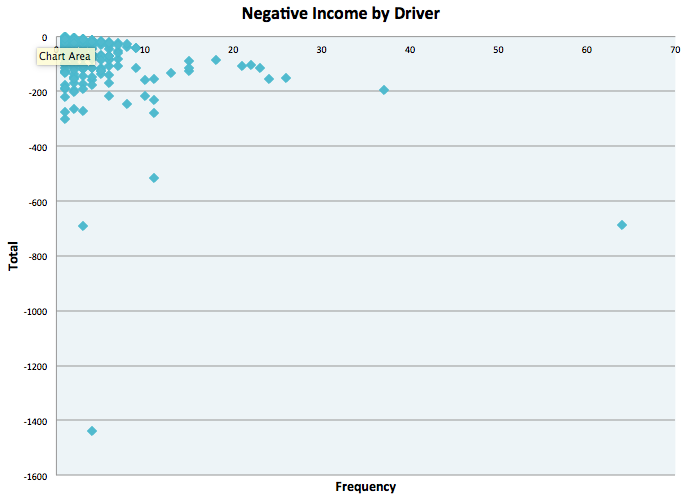


Figure A1.

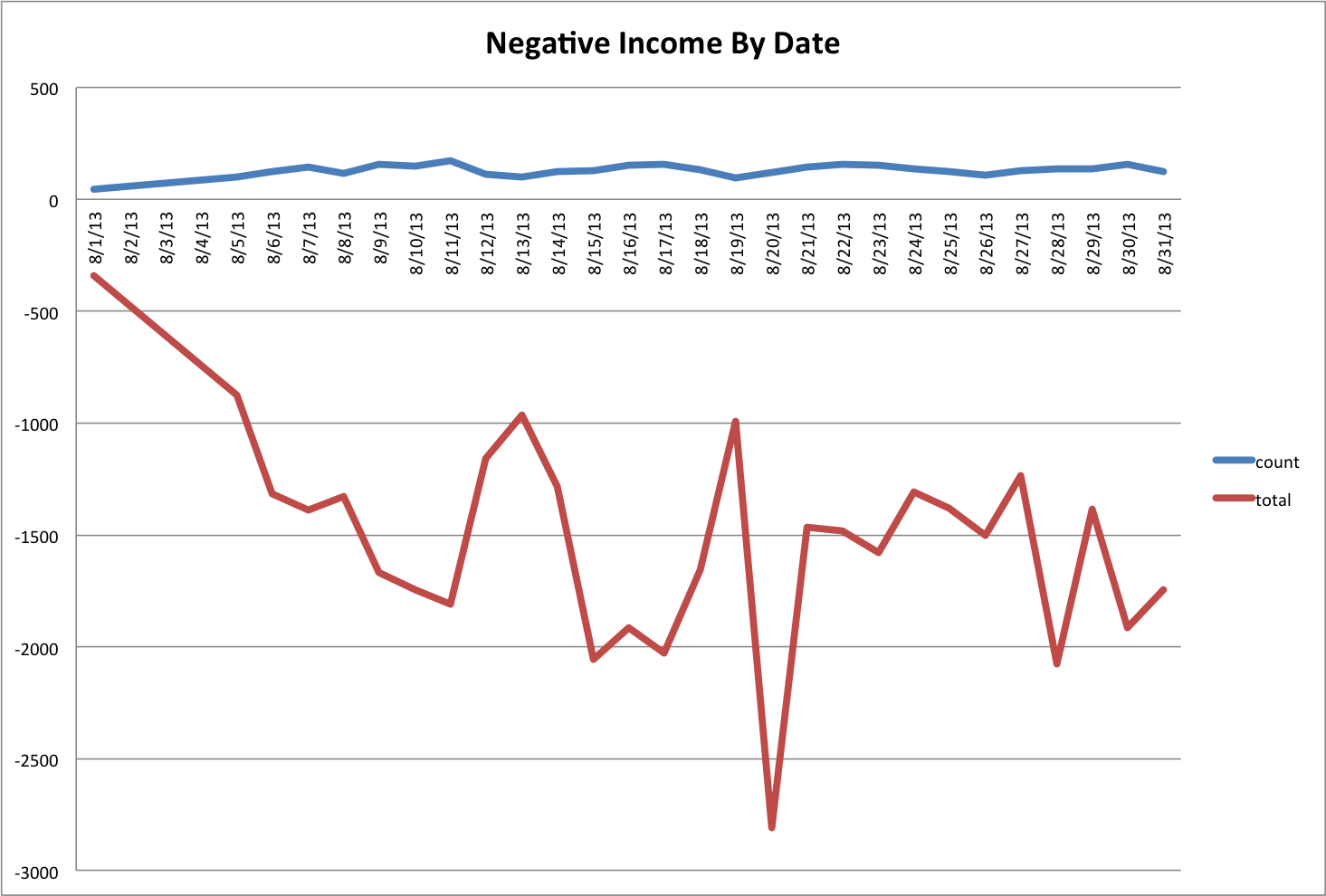


Figure A2.

## **Section 2. Overcharged**

Explanation:

The idea solve this problem is almost the same as the negative income. The only difference is that instead checking whether total\_amount is negative, we need to compare total\_amount with tolls\_amount + tip\_amount + mta\_tax + surcharge + fare\_amount. We only keep those difference bigger than **5 dollars**.

Also we think group by driver or by date is a good choice to analysis data.

In this analysis, because the condition and process is almost the same. So we also want to compare the performance of different running environment. By change instance type from moderate m1.medium. We change them to m3.xlarge and m1.large. We can see the running time significant to reduce to half of the previous one even using less instance.

Running environment:

* Master: 1\* m3.xlarge
* Core: 2 \* m1.large
* Arguments: s3://elasticmapreduce/libs/pig/pig-script --run-pig-script --pig-versions 0.12.0 --args -f s3://zcxfinalproject/src/q5j2.pig -p INPUT=s3://zcxfinalproject/input/\* -p OUTPUT=s3://zcxfinalproject/output/
* Running Time: 39 minutes

Result

1. Jan, Feb, Mar and Apr has no such error, the overcharge problem starts from May.
2. Aug 14, and Aug 5 have a honorable overcharge. And this two overage may due to driver 6649 and E4F9
3. Two huge overcharge in Aug make this month incredible high. Other month only have a slightly increase.
4. Driver usually make 1 or 2 overcharge between 5-15 dollars.

**Top Three:**

By Date

|  |  |  |
| --- | --- | --- |
| Date | Overcharge | Count |
| 8/14/13 | 686819.72 | 8 |
| 8/5/13 | 541468.56 | 4 |
| 8/31/13 | 1241.27 | 2 |

By Driver

|  |  |  |
| --- | --- | --- |
| Driver | count | Overcharge |
| 664927CDE376A32789BA48BF55DFB7E3 | 1 | 685874.1 |
| E4F99C9ABE9861F18BCD38BC63D007A9 | 1 | 541422.56 |
| B54ABD47DC3B5E1CCE11759127350BFE | 1 | 1234.52 |



Figure A3.

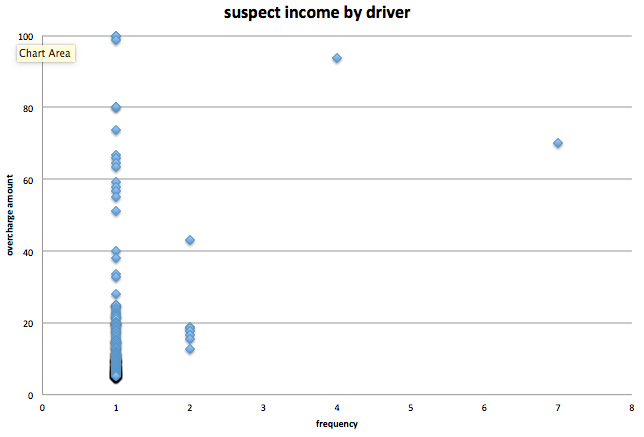


Figure A4.

# Part IV ISSUE

## Issues 1:

When we using python hadoop streaming. We found a very strange problem. By following the example given by Lecturer. We first implement our MapReduce logic code with python iterator.tools package. And the code runs pretty well on our local machine even with the really data. But when we create hadoop python streaming job on the EMR. The job always terminated at the beginning reduce process and complain with the IO Exception with broken pipe (As shown below). As the After do a little research, we found some other developers also complains the similar problem when they using python streaming. We try to solve this problem by the following step.

1. One reason may cause this problem is the bad data, so when python read as columns will null value and cannot convert to a right type. It will wait for the std.in and cause the broken pipe. We try to fixed by using python exception handling and the error still here.
2. So we rewrite the MapReduce by using the plain code. Which is using the variable to continuously record the current key and tracking the key changing. Once we found key changes, we put our logic there.

We guess this error may cause by the memory overflow by the python version on the AWS.

2014-12-02 15:55:56,387 INFO [main] org.apache.hadoop.streaming.PipeMapRed: R/W/S=2800000/0/0 in:121739=2800000/23 [rec/s] out:0=0/23 [rec/s]  
2014-12-02 15:55:57,869 INFO [Thread-38] org.apache.hadoop.streaming.PipeMapRed: MRErrorThread done  
2014-12-02 15:55:57,869 WARN [main] org.apache.hadoop.streaming.PipeMapRed: java.io.IOException: Broken pipe  
2014-12-02 15:55:57,878 INFO [main] org.apache.hadoop.streaming.PipeMapRed: PipeMapRed failed!  
java.lang.RuntimeException: PipeMapRed.waitOutputThreads(): subprocess failed with code 143  
 at org.apache.hadoop.streaming.PipeMapRed.waitOutputThreads(PipeMapRed.java:330)  
 at org.apache.hadoop.streaming.PipeMapRed.mapRedFinished(PipeMapRed.java:543)  
 at org.apache.hadoop.streaming.PipeReducer.reduce(PipeReducer.java:128)  
 at org.apache.hadoop.mapred.ReduceTask.runOldReducer(ReduceTask.java:445)  
 at org.apache.hadoop.mapred.ReduceTask.run(ReduceTask.java:393)  
 at org.apache.hadoop.mapred.YarnChild$2.run(YarnChild.java:167)

## **Issue 2:**

After we switch from hadoop streaming to pig, we found the configuration is a little different. Before, we only need to specify the input folder as the s3 bucket input folder in the steaming step setup page. Amazon hadoop will read these files and pipe them into our python script files.

But In the pig, we found that if we only setup s3 in job steps. The pig cannot read them direct from s3 folders. Instead, it will complains that it cannot find out any files. After checking the log, we find that during pig process in Amazon, even you include s3 folder in your step setup pages. You still need explicitly write the whole s3 url in your pig script. Otherwise, Amazon hadoop pig will read hdfs in the local instance instead of Amazon. The behavior is the same in **LOAD & STORE** commands. (I have run pig process on Amazon, after all step completed, I found nothing in S3 folder. At last, I realized they store in local HDFS and gone when I terminate this machines).

INFO org.apache.hadoop.mapreduce.lib.jobcontrol.ControlledJob - PigLatin:q5j1.pig got an error while submitting   
org.apache.pig.backend.executionengine.ExecException: ERROR 2118: Input Pattern hdfs://172.31.10.183:9000/trip\_fare\_\*.csv matches 0 files  
 at org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.PigInputFormat.getSplits(PigInputFormat.java:288)  
 at org.apache.hadoop.mapreduce.JobSubmitter.writeNewSplits(JobSubmitter.java:493)  
 at org.apache.hadoop.mapreduce.JobSubmitter.writeSplits(JobSubmitter.java:510)

**Issue 3:**

When we try to use the google chart to make graph,we encounter a problem about the Date type.When we deal with the date,if we don’t change our original date,the date shown in the graph will be a month later.For example,if we want to show 2013,January in the chart,we need to make our data’s date field into 2012,December,it will automatically add one month and display it in the graph.