CSCI-B 565 DATA MINING
Project Work (For Extra Credit)
Morning Class
Computer Science Core
Fall 2013
Indiana University

Debpriya Seal
debseal@indiana.edu

December 20, 2013

All the work herein is solely mine.

UFO Data Analysis

Data Cleansing

After listening Prof. Memo mention that storing data in realtional format, fetch you some benefits. And after looking at the project , I thought pushing the data to the This time, I am using **SQLite3** to store the given json file into a relational table. In that i have created a table wiht below attribute:

```
CREATE TABLE IF NOT EXISTS UFO_DATA
                           (sighted_at  TEXT NOT NULL,
                           reported_at TEXT NOT NULL,
                           city    TEXT   NULL,
                           state    TEXT   NULL,
                           country    TEXT  NULL,
                           shape       TEXT,
                           duration    INT,
                           description TEXT);
```

Below are the transformation I did:

1. I breaked the location, into 3 different fields:
    a. City
    b. State
    c. Country
2. Converted sighted_at and reported_at to **date** format.

3. Moreover, I converted the **duration into a single unit of seconds**. As that way we can measure them. However, for few records with duration mentioned as ”1 & 1/2 hour”, i was not able to convert.

This took me most of the time, as handling all the possible scenarios was a big problem for me. Much against my belief, the data **does** talk about sitings even outside US of A.

On further analysis data, I found that many records are not in good state. Below were the reasons:

1. *sighted_at* date found ”0000”, which is not a proper date format. For example:
```
"InvalidDate","{"sighted_at": "0000","reported_at": "19951218","location": "Chattanooga"," TN",
"shape": ","duration":," "description": "He called seeking information regarding the UFO
incident over McMinnville","TN"," on 07JA95.""}"
```
2. Few records were not in json format. Like, the end paranthesis were missing etc. For example:
```
"InvalidJSONFormat","{"sighted_at"": "20041022","reported_at": "20041022","
 "location": " High Point (rural)","  & ","  "shape": " unknown"," "duration": "45 seconds?",
"description": ""Very fast and turned. Not a meteor.Sitting at a stop sign on country
 road and facing south we both saw object moving very fast at about 45 degree
angle from horizon.Object had bright light on front with lesser light behind it.
First thought was meteor due to inc"
```

I redirected such records to a seperate file named ***ufo_awesome.bad***. The .bad file is a csv file with below:

```
        <ReasonOfBad>,<The correpsonding record.>
```

I have attached the same for your reference.

Data Analysis

- Are the majority of witnesses male or female?
  Answer: There was no easy way to figure this out. Hence, I used the description to find the male or female count. I agree this is not the best way to do this. But given the time constraint, I could not do better. Below are the snapshot from my Java Code ran on **SQLite3**:

  ```
  SELECT COUNT(*) AS COUNT FROM UFO_DATA
  WHERE LOWER(DESCRIPTION) LIKE "% woman %" OR
  LOWER(DESCRIPTION) LIKE "% girl %" OR
  LOWER(DESCRIPTION) LIKE "% female %" OR
  LOWER(DESCRIPTION) LIKE "% fe-male %";
  SQL successfully executed
  Fe-male Count = 555

  SELECT COUNT(*) AS COUNT FROM UFO_DATA
  WHERE LOWER(DESCRIPTION) LIKE "% man %" OR
  LOWER(DESCRIPTION) LIKE "% boy %" OR
  LOWER(DESCRIPTION) LIKE "% male %";
  SQL successfully executed
  Male Count = 942
  ```

- Are the siting in the U.S. located to a particular region or time?
  Answer:

Location Analysis:
The reason, I bifurcated the location field into city, state and country. Was to make life easy now. I just grouped by the State to get a count at State level. Below are the snapshot from my Java Code ran on **SQLite3**:

```
SELECT STATE, COUNT(STATE) AS COUNT FROM UFO_DATA GROUP BY STATE ORDER BY COUNT DESC ;
SQL successfully executed
CA     :    3492
null    :     2717
WA     :    1921
TX     :    1263
FL     :    1129
NY     :    1070
AZ     :    1036
IL     :    814
OH     :    795
OR     :    792
PA     :    757
MI     :    682
ON     :    588
CO     :    581
MO     :    546
BC     :    510
WI     :    477
NJ     :    467
NC     :    466
IN     :    432
VA     :    403
TN     :    389
GA     :    386
MA     :    386
NV     :    376
MN     :    340
NM     :    295
AR     :    271
KY     :    269
MD     :    262
UT     :    261
OK     :    258
CT     :    254
IA     :    224
KS     :    216
SC     :    210
ME     :    202
AL     :    201
LA     :    191
ID     :    175
MT     :    172
WV     :    156
NH     :    151
AB     :    142
MS     :    140
```

```
NE        :        133
AK        :        124
PQ        :        100
HI        :        99
WY        :        94
RI        :        69
VT        :        61
MB        :        55
ND        :        55
SD        :        54
NS        :        50
DE        :        49
DC        :        41
SA        :        37
PR        :        36
SK        :        31
NB        :        28
QC        :        24
NT        :        10
NF        :        9
PE        :        8
YK        :        7
YT        :        2
VI        :        1
```

Now the null are the ones,for which state information was missing.However, you would find few states here which are not part of USA and that would be because this data consist of data from other part of world as well.

Time Analysis I started initially with Month. And below is the percentage i got.

```
SELECT M.Month as Month, CAST(M.Count AS REAL)/C.COUNT*100  AS Count FROM
(
        SELECT
                1 AS X,
                COUNT(*) AS COUNT
        FROM UFO_DATA
) AS C
JOIN
(
        SELECT
                1 AS X,
                strftime('%m',date(sighted_at)) AS Month,
                COUNT(*) AS Count
        FROM UFO_DATA
        GROUP BY 1,strftime('%m',date(sighted_at))
) AS M
ON C.X=M.X
ORDER BY COUNT DESC;
```

```
08    =    11.52 %
07    =    11.51 %
06    =    11.06 %
09    =    9.55 %
10    =    8.62 %
11    =    8.23 %
05    =    7.03 %
03    =    6.91 %
01    =    6.59 %
04    =    6.48 %
12    =    6.30 %
02    =    6.18 %
```

My second idea was to look for **Years** in particular. And below are my percentage for year.

```
SELECT M.Year as Year, CAST(M.Count AS REAL)/C.COUNT*100  AS Count FROM
(
    SELECT
        1 AS X,
        COUNT(*) AS COUNT
    FROM UFO_DATA
) AS C
JOIN
(
    SELECT
        1 AS X,
        strftime('%Y',date(sighted_at)) AS Year,
        COUNT(*) AS Count
    FROM UFO_DATA
    GROUP BY 1,strftime('%Y',date(sighted_at))
) AS M
ON C.X=M.X
ORDER BY COUNT DESC;
```

```
2003    =    14.3213750802368 %
2002    =    11.9855930390129 %
2001    =    11.3009057841809 %
2004    =    10.5413308608516 %
1999    =    10.1561942800086 %
2000    =    9.85664360601954 %
1998    =    6.08373154553884 %
1995    =    4.33991869338849 %
1997    =    3.76934598102846 %
1996    =    2.72448470151915 %
1994    =    0.930746737037301 %
1993    =    0.709649810997789 %
1978    =    0.634762142500535 %
1975    =    0.591969189073533 %
1989    =    0.54561015619428 %
1976    =    0.53847799728978 %
1990    =    0.517081520576278 %
1992    =    0.502817202767278 %
1974    =    0.481420726053777 %
```

```
1991        =        0.477854646601526 %
1977        =        0.474288567149276 %
1980        =        0.470722487697026 %
1973        =        0.427929534270024 %
1987        =        0.424363454817773 %
1979        =        0.417231295913273 %
1988        =        0.410099137008773 %
1984        =        0.388702660295271 %
1982        =        0.385136580843021 %
1968        =        0.381570501390771 %
1966        =        0.36374010412952 %
1986        =        0.36374010412952 %
1985        =        0.36017402467727 %
1965        =        0.345909706868269 %
1967        =        0.335211468511518 %
1972        =        0.328079309607018 %
1983        =        0.310248912345767 %
1981        =        0.306682832893517 %
1970        =        0.278154197275515 %
1969        =        0.267455958918765 %
1971        =        0.231795164396263 %
1964        =        0.156907495899009 %
1963        =        0.149775336994508 %
1957        =        0.146209257542258 %
1960        =        0.146209257542258 %
1952        =        0.117680621924256 %
1954        =        0.117680621924256 %
1962        =        0.114114542472006 %
1961        =        0.0927180657585051 %
1958        =        0.0891519863062549 %
1947        =        0.0784537479495043 %
1959        =        0.0784537479495043 %
1956        =        0.0713215890450039 %
1953        =        0.0641894301405035 %
1955        =        0.0606233506882533 %
1951        =        0.0392268739747522 %
1949        =        0.0320947150702518 %
1945        =        0.0249625561657514 %
1950        =        0.0249625561657514 %
1944        =        0.0213964767135012 %
1943        =        0.0142643178090008 %
1946        =        0.0106982383567506 %
1948        =        0.0106982383567506 %
1942        =        0.00713215890450039 %
1860        =        0.0035660794522502 %
1865        =        0.0035660794522502 %
1906        =        0.0035660794522502 %
1910        =        0.0035660794522502 %
1916        =        0.0035660794522502 %
1920        =        0.0035660794522502 %
1929        =        0.0035660794522502 %
1930        =        0.0035660794522502 %
1931        =        0.0035660794522502 %
1935        =        0.0035660794522502 %
1936        =        0.0035660794522502 %
1937        =        0.0035660794522502 %
1939        =        0.0035660794522502 %
1941        =        0.0035660794522502 %
```

Lastly, I looked into date and found a interesting pattern. **That most of the citings are on 15th.**

```
SELECT M.Date as Date , CAST(M.Count AS REAL)/C.COUNT*100  AS Count FROM
(
        SELECT
                1 AS X,
                COUNT(*) AS COUNT
        FROM UFO_DATA
) AS C
JOIN
(
        SELECT
                1 AS X,
                strftime('%d',date(sighted_at)) AS Date,
                COUNT(*) AS Count
        FROM UFO_DATA
        GROUP BY 1,strftime('%d',date(sighted_at))
) AS M
ON C.X=M.X
ORDER BY COUNT DESC;
```

```
15        =        10.1918550745311 %
01        =        7.88816774837743 %
20        =        4.02966978104272 %
10        =        3.74794950431496 %
16        =        3.41630411525569 %
12        =        3.16311247414592 %
13        =        3.11318736181442 %
07        =        3.08822480564867 %
11        =        3.02760145496042 %
04        =        3.00977105769917 %
25        =        2.97411026317666 %
17        =        2.96341202481991 %
14        =        2.94914770701091 %
23        =        2.91348691248841 %
19        =        2.83859924399116 %
05        =        2.8136366878254 %
09        =        2.73161686042365 %
18        =        2.73161686042365 %
28        =        2.7209186220669 %
08        =        2.70308822480565 %
06        =        2.6995221453534 %
22        =        2.6638613508309 %
26        =        2.57470936452464 %
24        =        2.56401112616789 %
30        =        2.54618072890664 %
03        =        2.53191641109764 %
21        =        2.53191641109764 %
02        =        2.46772698095714 %
27        =        2.39640539191213 %
29        =        2.22879965765637 %
31        =        1.77947364667285 %
```

- Its often claimed (popularly) that sitings are most common on Tuesday orWednesday. Is this consistent with the data?
  Answer: No, from my analysis this is not conistent with the data we are provided with. Below are mine statistics:

  ```
  SELECT CAST(M.SundaySight AS REAL)/C.COUNT*100  AS SundaySight FROM
  (
          SELECT 1 AS X, COUNT(*) AS COUNT
          FROM UFO_DATA
  ) AS C
  JOIN
  (
          SELECT 1 AS X, COUNT(1) AS SundaySight
          FROM UFO_DATA
          WHERE CAST(strftime('%w',date(sighted_at)) AS INTEGER)= 0
  ) AS M
  ON C.X=M.X;

  % of sighting on Sunday = 15.23 %

  SELECT CAST(M.MondaySight AS REAL)/C.COUNT*100  AS MondaySight FROM
  (
          SELECT 1 AS X, COUNT(*) AS COUNT
          FROM UFO_DATA
  ) AS C
  JOIN
  (
          SELECT 1 AS X, COUNT(1) AS MondaySight
          FROM UFO_DATA
          WHERE CAST(strftime('%w',date(sighted_at)) AS INTEGER)= 1
  ) AS M
  ON C.X=M.X;

  % of sighting on Monday = 12.99 %


  SELECT CAST(M.TuesdaySight AS REAL)/C.COUNT*100  AS TuesdaySight FROM
  (
          SELECT 1 AS X, COUNT(*) AS COUNT
          FROM UFO_DATA
  ) AS C
  JOIN
  (
          SELECT 1 AS X, COUNT(1) AS TuesdaySight
          FROM UFO_DATA
          WHERE CAST(strftime('%w',date(sighted_at)) AS INTEGER)= 2
  ) AS M
  ON C.X=M.X;

  % of sighting on Tuesday = 14.16 %
  ```

```
SELECT CAST(M.WednesdaySight AS REAL)/C.COUNT*100  AS WednesdaySight FROM
(
        SELECT 1 AS X, COUNT(*) AS COUNT
        FROM UFO_DATA
) AS C
JOIN
(
        SELECT 1 AS X, COUNT(1) AS WednesdaySight
        FROM UFO_DATA
        WHERE CAST(strftime('%w',date(sighted_at)) AS INTEGER)= 3
) AS M
ON C.X=M.X;
```

% of sighting on Wednesday = 13.61 %

```
SELECT CAST(M.ThursdaySight AS REAL)/C.COUNT*100  AS ThursdaySight FROM
(
        SELECT 1 AS X, COUNT(*) AS COUNT
        FROM UFO_DATA
) AS C
JOIN
(
        SELECT 1 AS X, COUNT(1) AS ThursdaySight
        FROM UFO_DATA
        WHERE CAST(strftime('%w',date(sighted_at)) AS INTEGER)= 4
) AS M
ON C.X=M.X;
```

% of sighting on Thursday = 14.20 %

```
SELECT CAST(M.FridaySight AS REAL)/C.COUNT*100  AS  FridaySight FROM
(
        SELECT 1 AS X, COUNT(*) AS COUNT
        FROM UFO_DATA
) AS C
JOIN
(
        SELECT 1 AS X, COUNT(1) AS FridaySight
        FROM UFO_DATA
        WHERE CAST(strftime('%w',date(sighted_at)) AS INTEGER)= 5
) AS M
ON C.X=M.X;
```

% of sighting on Friday = 14.17 %

```
SELECT CAST(M.SaturdaySight AS REAL)/C.COUNT*100  AS  SaturdaySight FROM
(
        SELECT 1 AS X, COUNT(*) AS COUNT
        FROM UFO_DATA
) AS C
JOIN
(
        SELECT 1 AS X, COUNT(1) AS SaturdaySight
        FROM UFO_DATA
        WHERE CAST(strftime('%w',date(sighted_at)) AS INTEGER)= 6
) AS M
ON C.X=M.X;
```

% of sighting on Saturday = 14.17 %

Challenges

- Inserting DATE into SQLite3. The issue was that the Java DATE and SQLIte3 DATE does not go along. Event though the INSERT was executing successfully. All the dates in tabel was getting set as "1969-12-31"
- SQLite does not have any data type as DATE, it store them in the primitive TEXT, REAL etc. format only.
- Bifurcate location into city, state and country.

External Libraries used

- sqlite-jdbc-3.7.15-M1 : To write given dataset into relational database.
- opencsv-2.3 : To write a file in a csv format.
- json-simple-1.1.1 : To parse a json file.