International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 07 | July-2018 www.irjet.net p-ISSN: 2395-0072

Predictive Policing in Crime analysis using R

Rohith Mallula[1], Preetham Chowdary[2]

[1]Student, VIT University Vellore [2]Student, VIT University Vellore

---

Abstract - *Predictive Policing alludes to the utilization of mathematical models created by predictive and analytical techniques in law enforcement to distinguish or identify potential criminal activity. Here we learn how to extract data directly from the web and then handle the data (which involves cleaning and re-organizing and processing the crime record data), recover sensitive information through visualizations, make new factor variables from the limited data and build a predictive engine for identifying potential attributes of any future crimes.*

Key Words: **extraction, crime data, heatmap, predictive policing, R programming, visualization**

1. INTRODUCTION

The recent couple of years USA has seen an overall decrease in the crime rate. This can be attributed to the fact that the improvisation happened in law enforcing agencies especially the ones that are aided by computers and the art of data science involved in it. Many law-enforcing agencies have started looking towards data science to convert their huge tons of data into meaningful insights. These new advancements even correlate to the fact that there is tremendous demand for Data Science and Analysis.

Now Predictive Policing is in the horizon of becoming the next big thing in research where the statistical data of crime is being collected and processed for potential insights in the future, like predicting the crime location or timing or vicinity to any nearest public service station etc. All these insights can be obtained by converting the huge data into meaningful inputs for any data analysis tools like Python, R etc. We utilized the Chicago crime dataset for our analysis and predictive model, which is available in their official site as open access in .csv format.

2. DATA EXTRACTION, EXPLORATION AND PREPROCESSING

Our first task is to extract the data directly from the web or from a file locally saved in the workstation. We can use the command **read.csv** to do this. We can view the data that is now stored in the computer memory using **str()** function. This function displays the data as it is saved in the memory of the computer. Often this command can be replaced with **summary()**.

```
> summary(crime.data)
     CASE.                   DATE..OF.OCCURRENCE              BLOCK
 HY572234:     3   1/1/2012 0:01 :    68    008XX N MICHIGAN AVE:     701
 HY576362:     3   9/1/2011 9:00 :    62    001XX N STATE ST     :     582
 HV217424:     3   12/1/2011 9:00:    50    008XX N STATE ST     :     519
 HY341462:     2   3/1/2012 9:00 :    50    0000X W TERMINAL ST  :     484
 HY387816:     2   10/1/2011 9:00:    47    076XX S CICERO AVE   :     470
 HV421725:     2   8/1/2011 9:00 :    47    0000X N STATE ST     :     444
 (Other) :377778   (Other)       :337469    (Other)              :334593
   IUCR            PRIMARY.DESCRIPTION
 486    : 29717   THEFT                :72233
 820    : 28160   BATTERY              :59892
 460    : 20321   NARCOTICS            :37856
 1811   : 20151   CRIMINAL DAMAGE      :36023
 1320   : 16505   BURGLARY             :24827
 610    : 16462   ASSAULT              :19492
 (Other):206477   (Other)              :87470
 LOCATION.DESCRIPTION                  ARREST          DOMESTIC
 STREET                     : 79864    N:245198        N:289939
 RESIDENCE                  : 53826    Y: 92595        Y: 47854
 SIDEWALK                   : 41390
 APARTMENT                  : 40530
 OTHER                      : 11249
 PARKING LOT/GARAGE(NON.RESID.): 10370
 (Other)                    :100564
      BEAT            WARD          FBI.CD       X.COORDINATE
 Min.   : 111   Min.   : 1.00   6      :72233   Min.   :1094469
 1st Qu.: 622   1st Qu.:10.00   08B    :51842   1st Qu.:1152885
```

Fig 1 Preview of data allocated in memory

The data is being stored at a crime incident level. Each and every occurrence has a unique identifier attached to it. For every crime incident there is one observation available in the dataset. There also are cases where there are duplicates available for some incidents. Such observations should be removed from the data. We can using two combinations of functions **subset()** and **duplicated().** There are also a few observations where there is insufficient data i.e. missing data which either need to be substituted or removed. These missing values maybe the coordinates of latitude or longitude which cannot be replaced by any random mathematical logic. So, we remove such observations using **is.na** functions. There are also some illogical signs and values in certain rows like "CASE" is being input as "CASE#". Now this data can be ignored and removed using the same function defining for each and every unwanted variable that is present. The occurrence date will give approximate time and date stamp about when the crime might have happened. We can now view the first few observations to see how data is being stored after these modifications, using **head()** function.

```
>head(crime.data$DATE..OF.OCCURRENCE)
[1] 5/22/2011 18:03 5/22/2011 18:05 5/22/2011 18:05 5/22/2011 18:06 5/22/2011 18:10 5/22/2011
18:10
121027 Levels: 1/1/2012 0:00 1/1/2012 0:01 1/1/2012 0:02 1/1/2012 0:03 1/1/2012 0:05 1/1/2012
0:06 ... 9/9/2011 9:58
```

Fig 2 Preview of data allocated in memory after changes

R compiler didn't identify date as an object but is instead treated it like a factor variable. We modify this using **as.POSIXlt()** function. After using this function R identifies the date as an actual date instead of a simple variable. Now R compiler will be able to identify that the data stored in the columns are actually date and the time-stamps. We can even simplify the data by separating the date and time-stamps as individual variables. This is done using the **times()** function which can be accessed from the **chron** library after installing a separate **chron** package(// **install.packages(chron)** //) . The rate at which crimes happen is not necessarily steady and consistent all through the day. In some time-intervals there may be more while sometimes it can be very less. In order to analyze this, we can divide the 24hr day into four six-hour time intervals starting from midnight. Then we need to match these times intervals to each timestamp in the data. **Cut()** functions helps us in doing this operation. By doing so we can easily identify whether crimes or happening more in the day or the night. Now with this minimal data we can draw many conclusions and find many patterns in the happenings of crime.

We can find which day of the week and which day of the month using the date of crime incidence. Simply creating two variables like **crime.data$day**, **crime.data$month** and use the functions **weekdays()** and **months().** The data has nearly 31 different types and not all are mutually exclusive. So, we can merge two or more similar groups into one so our simplifications and analysis becomes a bit easier.

```
> table(crime.data$PRIMARY.DESCRIPTION)
              ARSON                          ASSAULT                        BATTERY
                465                            19484                          59855
           BURGLARY              CRIM SEXUAL ASSAULT                CRIMINAL DAMAGE
              24807                             1295                          35995
  CRIMINAL TRESPASS               DECEPTIVE PRACTICE                       GAMBLING
               8338                            11772                            759
           HOMICIDE   INTERFERE  WITH  PUBLIC  OFFICER  INTERFERENCE WITH PUBLIC OFFICER
                474                              168                           1022
       INTIMIDATION                       KIDNAPPING            LIQUOR LAW VIOLATION
                161                              245                            618
MOTOR VEHICLE THEFT                        NARCOTICS                    NON-CRIMINAL
              17704                            37828                              2
          OBSCENITY       OFFENSE INVOLVING CHILDREN       OTHER NARCOTIC VIOLATION
                 29                             2095                              5
      OTHER OFFENSE                    OTHER OFFENSE                   PROSTITUTION
              18323                                2                           2449
   PUBLIC INDECENCY            PUBLIC PEACE VIOLATION                        ROBBERY
                 16                             2957                          13389
        SEX OFFENSE                         STALKING                          THEFT
                994                              186                          72115
  WEAPONS VIOLATION
               3892
```

Fig 3 Preview of data in memory after cleaning

```
>table(crime.data$crime)
ARSON  ASSAULT  BATTERY  BURGLARY  DAMAGE  DRUG   FRAUD     HOMICIDE
  465    19484    59855     24807   35995  37833  11772          474
MVT    NONVIO   OTHER    ROBBERY      SEX  THEFT  TRESPASS       VIO
17704     5918    18325     13389     4738  72115   8338        6232
```

Fig 4 Preview of data in memory after re-arranging

2. DATA VISUALIZATIONS

In order to gain valuable insights, data visualization is a very effective way. Through visualization we can observe and understand the patterns and trends of the crime happenings.

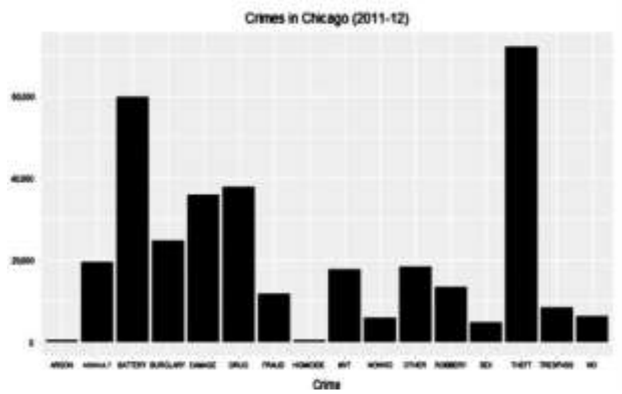

Fig 5 Frequency of crimes in Chicago (2011-12)

The above plot is made using a simple **qplot()** function which is typically the same as **plot()** function.

We can also plot how different crimes happen during different times of the day or even in a week.

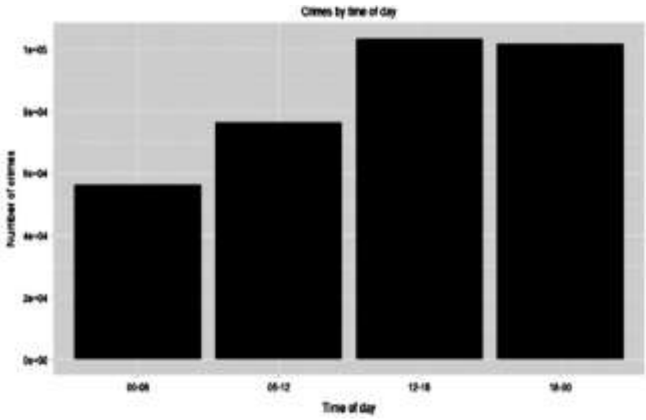

Fig 6 Distribution of crimes by day

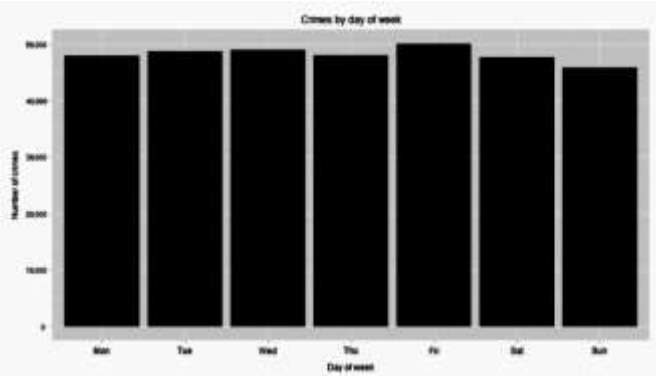

Fig 7 Distribution of crimes by week

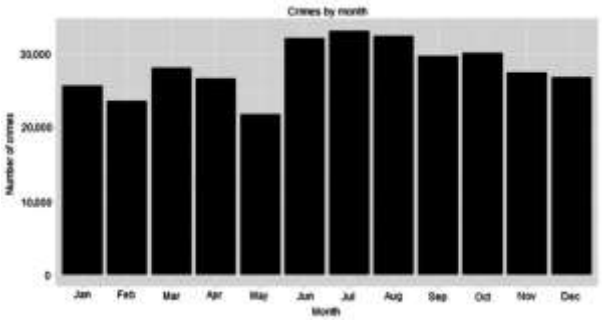

Fig 8 Distribution of crimes by month

Now we plot heatmaps to accurately find at what time, which type of crime is happening. We plot these heatmaps in similar way like the bar charts for day, week and month.

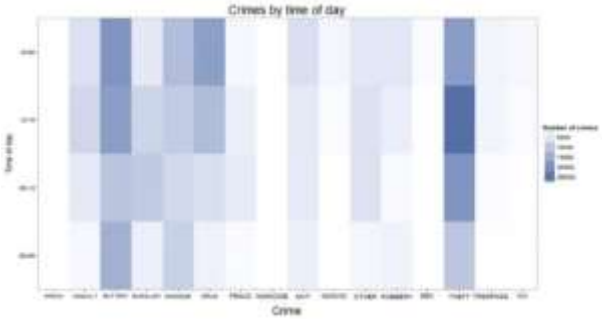

Fig 9 Heatmap of crimes by day

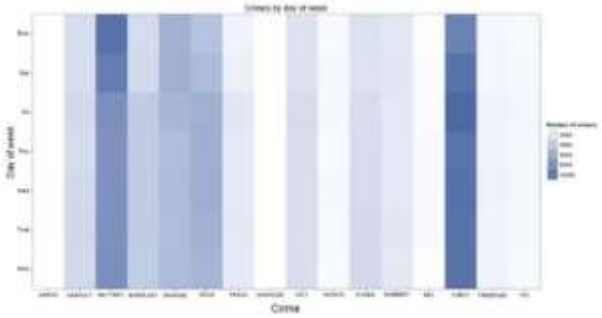

Fig 10 Heatmap of crimes by week

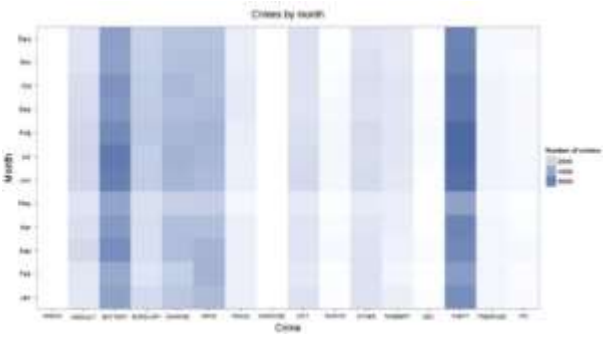

Fig 11 Heatmap of crimes by month

Before plotting the data on map, we need to manipulate the data a bit more and get the counts of crime for each and every location in Chicago. Till now we didn't consider the spatial element in analysis i.e. where there is high crime rate and where there is low crime rate. We can do this by using the **ddply()** function from the **plyr** library (**//install.packages(plyr)//**). This function has the capability to take a part of the data frame and apply the function only to that part and even gives the result in the form of a data frame. Now in order to convert the shape file (map of Chicago) into data frame we use **ggplot2**'s **fortify.SpatialPolygons() function** and store it into a variable by any name say breeze.shp .

Fig 12 Crimes in Chicago on 22 May 2011

'+' symbol represents the police stations and the dots indicate the type of crime as mentioned in the figure.

3. MODELING

For our goal of predictive policing model, we need to construct a model that is fairly sized predefined geographical area for a fairly long predefined time interval. To serve this purpose we build a multivariate regression model that has negative binomial distribution for errors.

So to make this model we initially create a dataset which has all possible combinations of breeze and different time slots. We can use the function **expand.grid()** with variable breeze and dates inside it. For our ease we sort the dataframe by breeze and assign it to a temporary variable. Therefore, the total number of crimes that happen in a breeze on a particular day can be obtained by taking the aggregate of breeze and dates. Now we overlap the aggregated crime data with the temporary dataset that we earlier created to obtain our final modeling data set. One of the important indicator of criminal activities can be the criminal activities in the past. In order to calculate the criminal histories we take the function **ave()** and add a customized input function that helps us in simplification by supplying a single argument. **past.Days()** is also being added for the calculation of number crimes per single breeze. Only using the variable of past arrests is not sufficient enough because the whole output will be highly correlated with the past crimes variable.

Thus we have created a predicator system and we can also check the authenticity of this model by taking a simple correlation between the dependent and independent variable. We use the **psych** library that has the **cor()** function to perform that action**.**

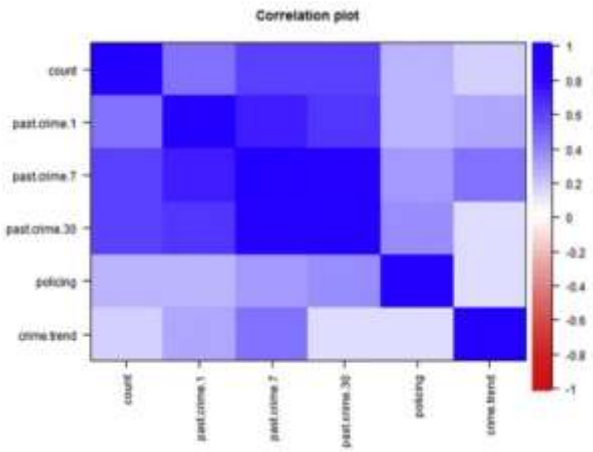

Fig 13 Correlation Matrix Plot

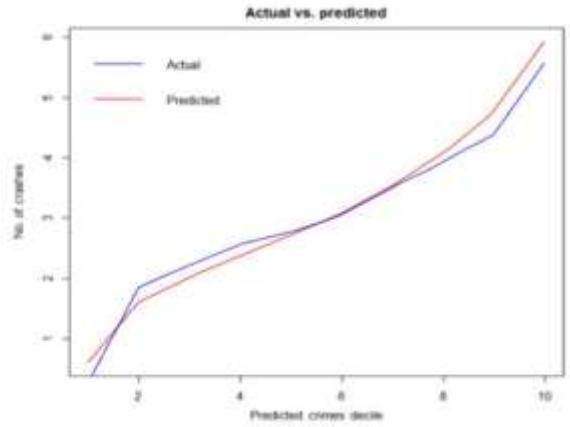

Fig 13 Actual vs Predicted (when tested with sample dataset)

CONCLUSION

We built a model that gives us the expected number of crimes that can happen in Chicago inn each day in a given time interval. This model can be built for any city or state in the world provided that there is data to clean and analyze. However, this model can be further improvised by adding a spatial dimension to it so that it can predict the crimes happening in a particular location at a particular time interval. And even crimes like murder and assault etc. need special attention so a predictor system can be built to predict these crime happenings to a pinpoint location.

REFERENCES

- Kass, Robert E. "Nonlinear Regression Analysis and its Applications." Journal of the American Statistical Association, vol. 85, no. 410, 1990, p. 594+. Academic OneFile, Accessed 8 July 2018.
- Bo Pang and Lillian Lee (2008), "Opinion Mining and Sentiment Analysis", Foundations and Trends® in Information Retrieval: Vol. 2: No. 1–2, pp 1-135
- Usama Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, From Data Mining to Knowledge Discovery in Databases, AI Magazine.
- Brent E Turvey, Criminal Profiling: An Introduction to Behavioral Evidence Analysis