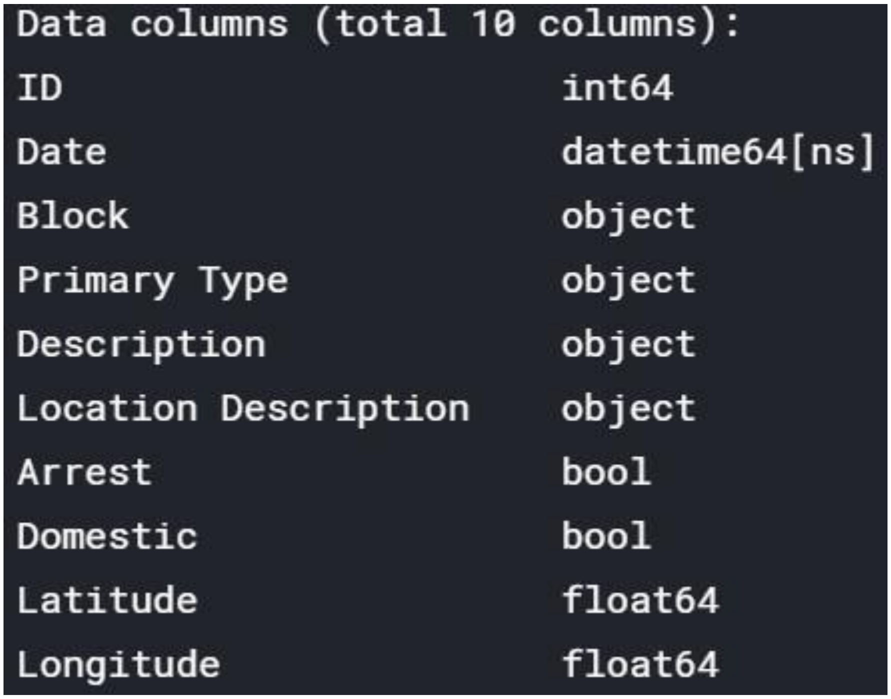
Abstract:

Crime analysis is a systematic approach for identifying and analyzing patterns and trends in Crime. Crimes have a negative effect on any society both socially and economically. Law enforcement bodies face numerous challenges while trying to prevent crimes.

Our project can show regions which have high probability for crime occurrence, visualize crime prone areas, forecast the crimes occurrence and analyze the crime rates over the years. This helps to perform descriptive, predictive, and prescriptive analysis on crime data. With the increasing advent of computerized systems, crime data analysts can help the Law enforcement officers to speed up the process of solving crimes. Instead of focusing on causes of crime occurrence like criminal background of offender, political enmity etc we are focusing mainly on crime primary types of each day.

Table of Contents

Dataset: Chicago Crime (2001-Present)
Total 22 columns
1.6GB
6.5 M records



```
Data columns (total 10 columns):
ID                        int64
Date                      datetime64[ns]
Block                     object
Primary Type              object
Description               object
Location Description      object
Arrest                    bool
Domestic                  bool
Latitude                  float64
Longitude                 float64
```

Introduction:

Crimes are a social nuisance and it has a direct effect on a society. Governments spend lots of money through law enforcement agencies to try and stop crimes from taking place. Today, many law enforcement bodies have large volumes of data related to crimes, which need to be processed to turn into useful information.

Crime data are complex because they have many dimensions and in different formats, e.g., most of them contain string records and narrative records. Due to this diversity, it is difficult to mine them using off the shelf, statistical and machine learning data analytics tools.It is the primary reason for lack of general platform for crime data mining.

We have used string, categorical dataset for our analysis. This dataset reflects reported incidents of crime (with the exception of murders where data exists for each victim) that occurred in the City of Chicago from 2001 to present, minus the most recent seven days. Data is extracted from the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system. In order to protect the privacy of crime victims, addresses are shown at the block level only and specific locations are not identified. The dataset contains more than 6,000,000 records/rows of data and cannot be viewed in full in Microsoft Excel.

To access a list of Chicago Police Department - Illinois Uniform Crime Reporting (IUCR) codes, go to https://catalog.data.gov/dataset/crimes-2001-to-present-398a4

Problem statement:

The problem that this project will try to address can be stated as follows:

We intend to perform time-series analysis of crime rates over a period (2001-present date) using autoregressive integrated moving average (ARIMA model) and other statistical/regression models. Since, the size of data is huge, we plan on implementing it on Databricks.

Using these forecasting models, we aim to predict the crime rate in the coming years along with where or when a crime will be committed. We also aim to analyze various factors causing crime in Chicago. By being able to identify high crime neighborhoods we aim to analyze the social and demographic factors responsible for the same which can help to identify the root cause of violence, so that effective steps can be taken.

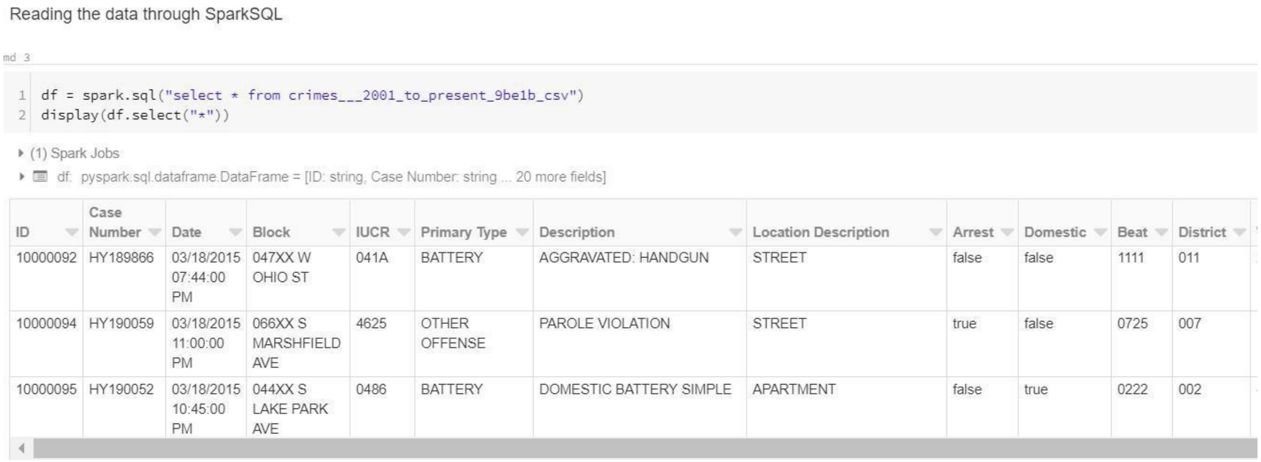
This analysis considers a range of data on Chicago crime along with crime type, geographical, economic and social indicators. All series are measured in annual observations over the period 2001 until present date.

Key questions answered:

- How has crime in Chicago changed across year from 2001 to present
- Are some types of crimes more likely to happen in specific locations or specific time of the day or specific day of the week than other types of crimes?
- Crime count forecasting by using SARIMA (Seasonal time series model) and ARIMA (autoregressive integrated moving average model)
- Predicting crime and arrest rates using Logistic Regression

Data Preparation

Reading data through SparkSQL



Reading the data through SparkSQL

md 3

```
df = spark.sql("select * from crimes___2001_to_present_9be1b_csv")
display(df.select("*"))
```

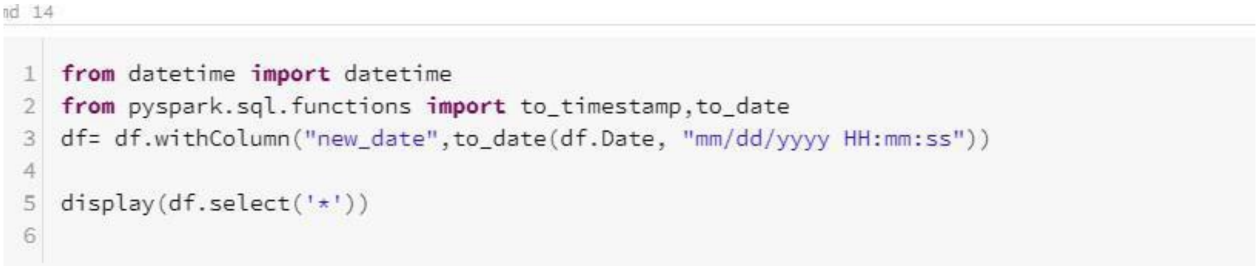
▸ (1) Spark Jobs

▸ df: pyspark.sql.dataframe.DataFrame = [ID: string, Case Number: string ... 20 more fields]

ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	Beat	District
10000092	HY189866	03/18/2015 07:44:00 PM	047XX W OHIO ST	041A	BATTERY	AGGRAVATED: HANDGUN	STREET	false	false	1111	011
10000094	HY190059	03/18/2015 11:00:00 PM	066XX S MARSHFIELD AVE	4625	OTHER OFFENSE	PAROLE VIOLATION	STREET	true	false	0725	007
10000095	HY190052	03/18/2015 10:45:00 PM	044XX S LAKE PARK AVE	0486	BATTERY	DOMESTIC BATTERY SIMPLE	APARTMENT	false	true	0222	002

Data Processing

- Converted date object in string to 'datetime' datatype for further
- Removing null values from the data

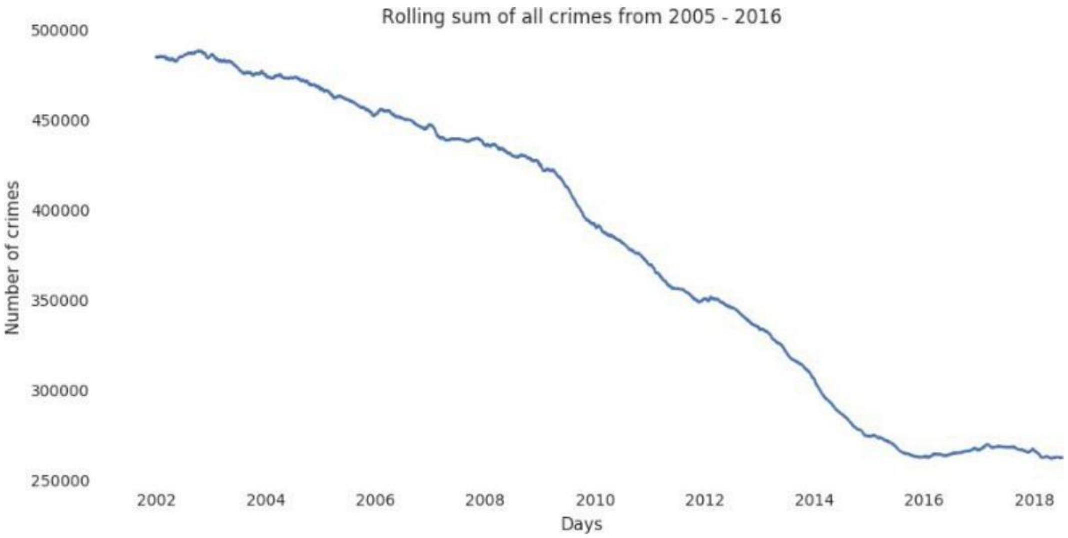


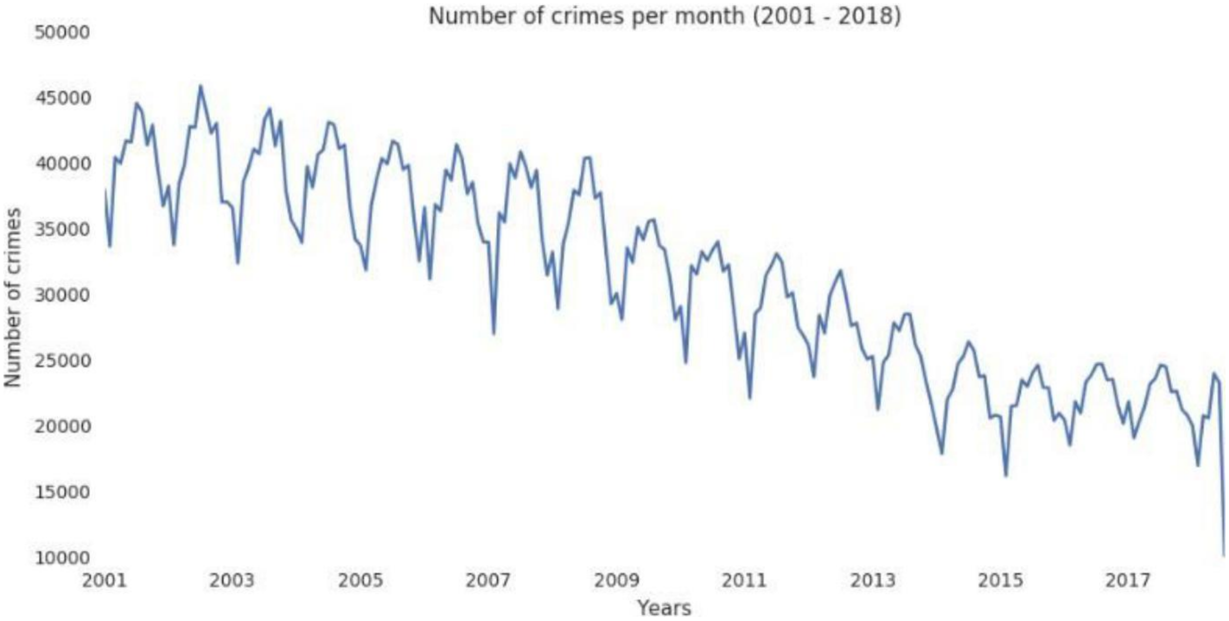
nd 14

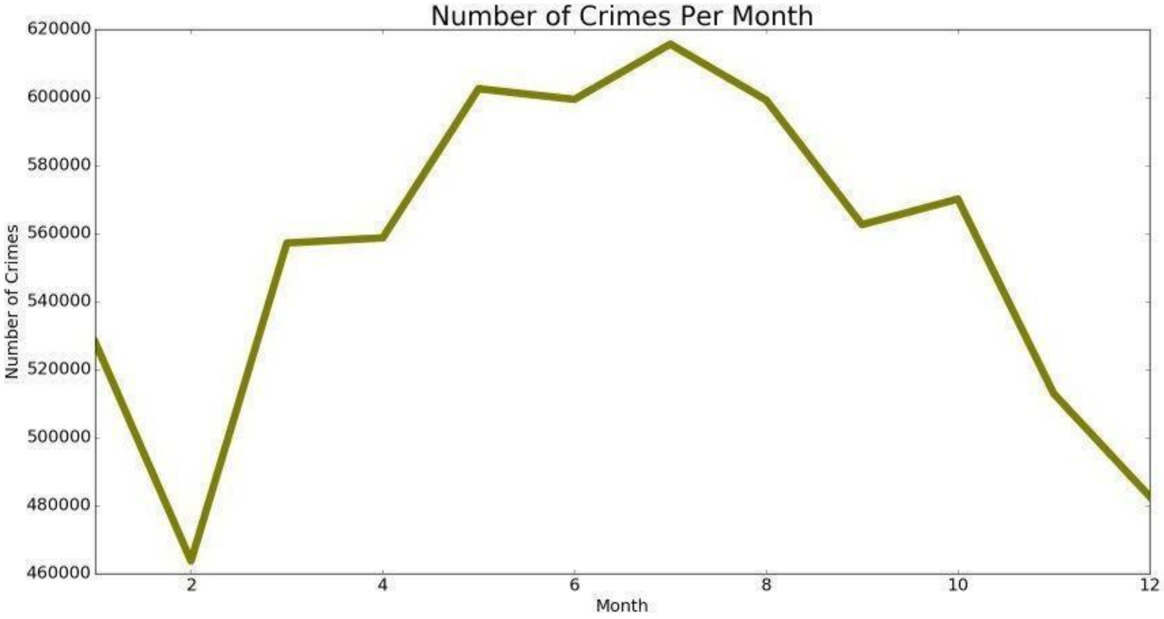
```
from datetime import datetime
from pyspark.sql.functions import to_timestamp,to_date
df= df.withColumn("new_date",to_date(df.Date, "mm/dd/yyyy HH:mm:ss"))

display(df.select('*'))

```

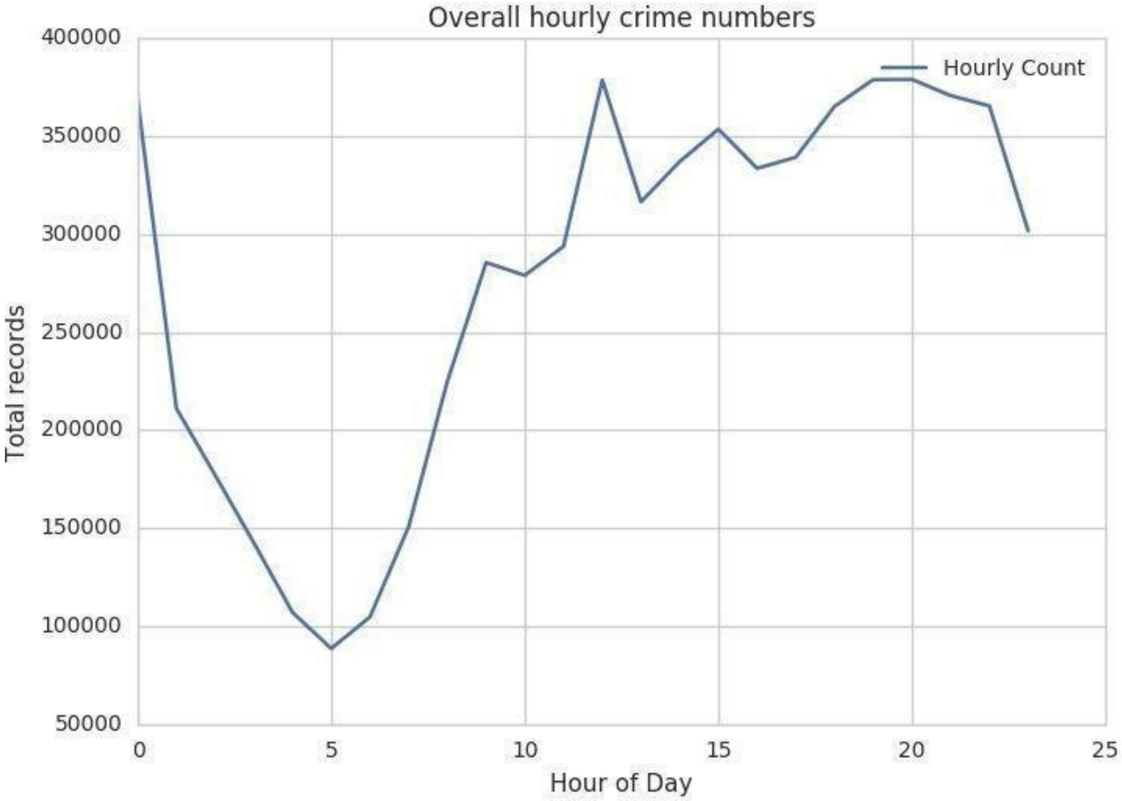
Exploratory data analysis Analysis1

No of crimes per month from 2001 to present


Rolling sum of all crimes from 2005 - 2016
500000
450000
400000
350000
300000
250000
Number of crimes
2002
2004
2006
2008
2010
2012
2014
2016
2018
Days


Number of crimes per month (2001 - 2018)
50000
45000
40000
35000
30000
25000
20000
15000
10000
Number of crimes
2001
2003
2005
2007
2009
2011
2013
2015
2017
Years

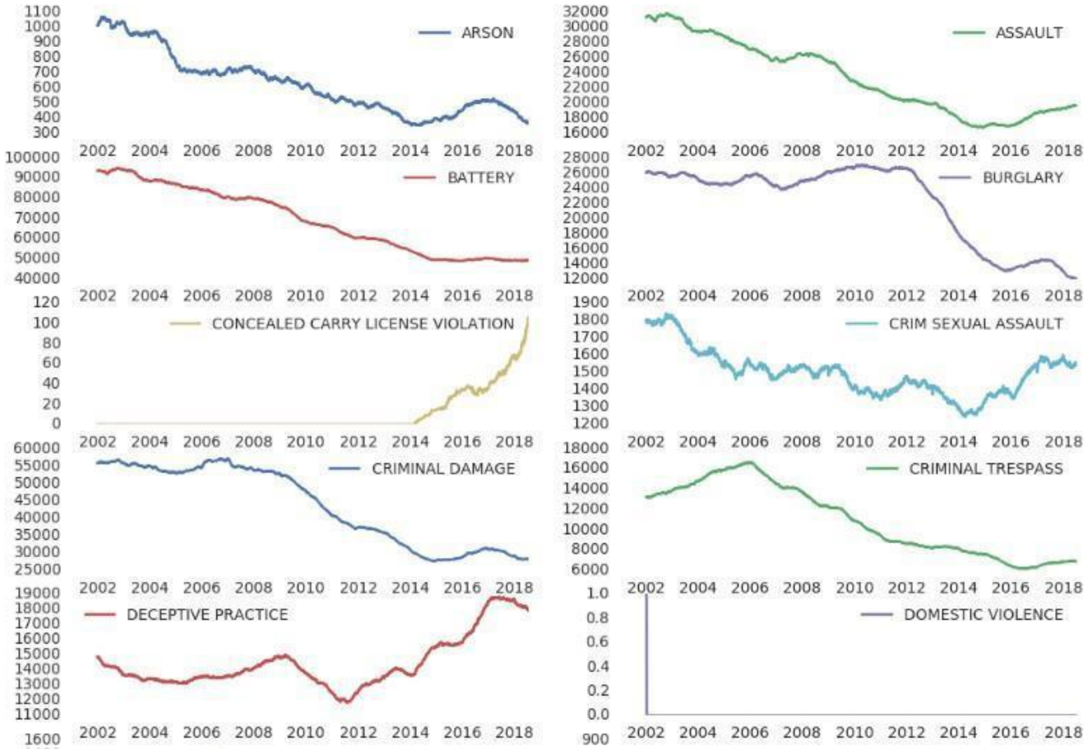

Number of Crimes Per Month
Number of Crimes
Month
620000
600000
580000
560000
540000
520000
500000
480000
460000
2
4
6
8
10
12

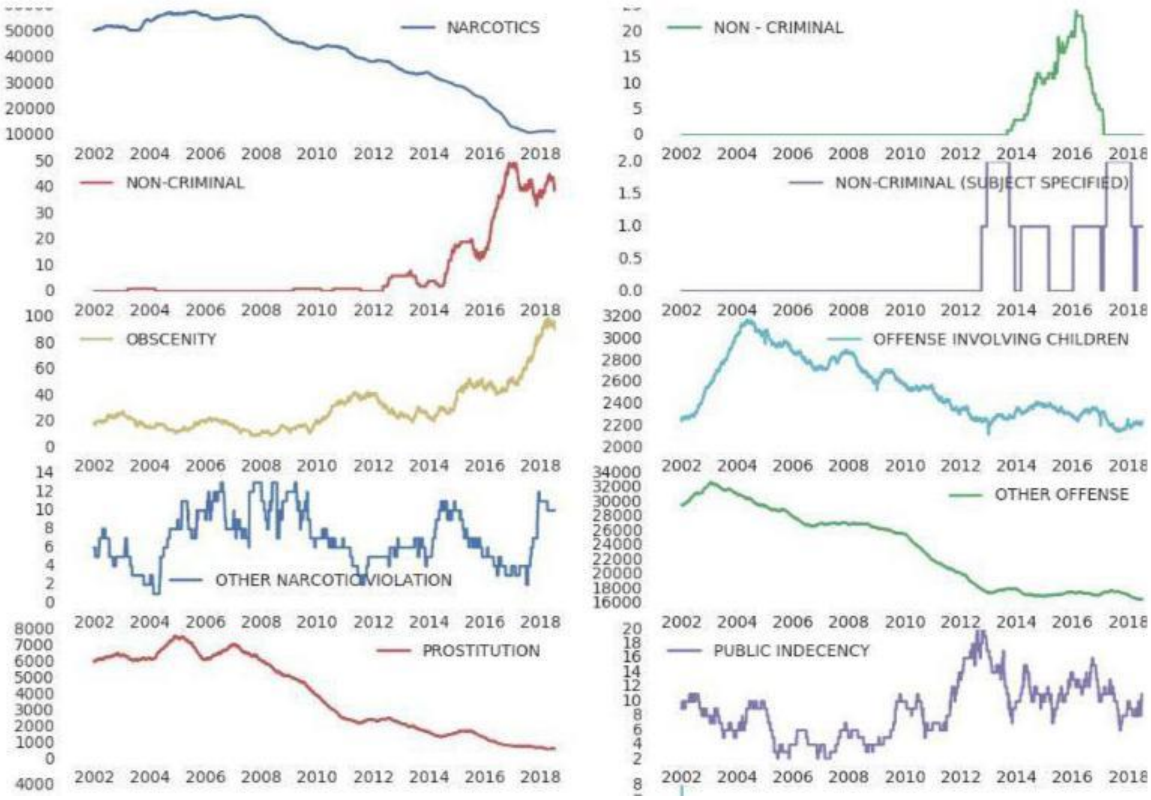

Overall hourly crime numbers
Hourly Count
Total records
Hour of Day
400000
350000
300000
250000
200000
150000
100000
50000
0
5
10
15
20
25

1. These graphs show a clear "periodic" pattern in the crimes over many years
2. The crime rates show a decreasing trend over a period
3. Research has shown that more police officers, using smarter investigative techniques, has helped bring down crime

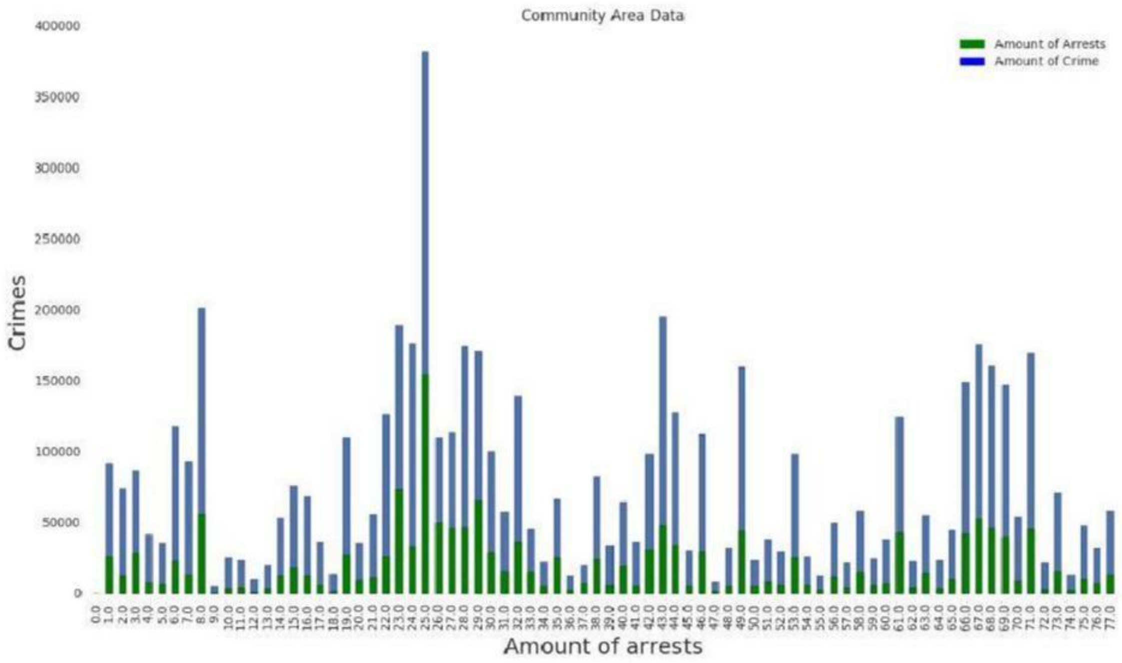
Analysis 2:

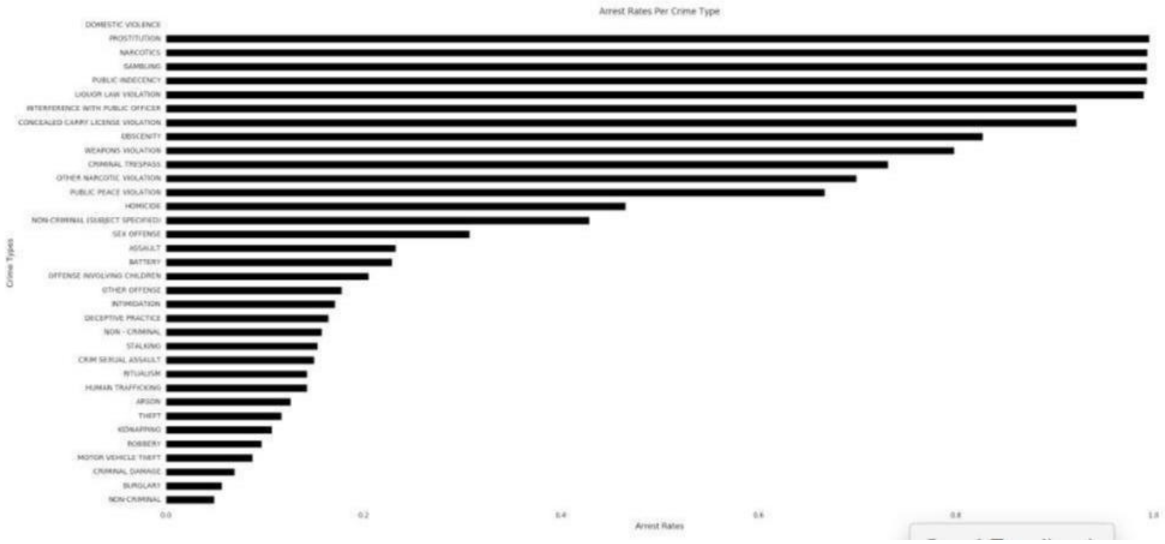
Potential causes of crime





1. The most frequent crime is theft
2. Concealed carry lease violation, Non-criminal, obscenity was non-existent until 2008 but are now increasing from 2010 onwards
3. Arson, Assault, Burglary, Criminal damage and narcotics crime rate is declining
4. Weapon Violation has increased compared to last year





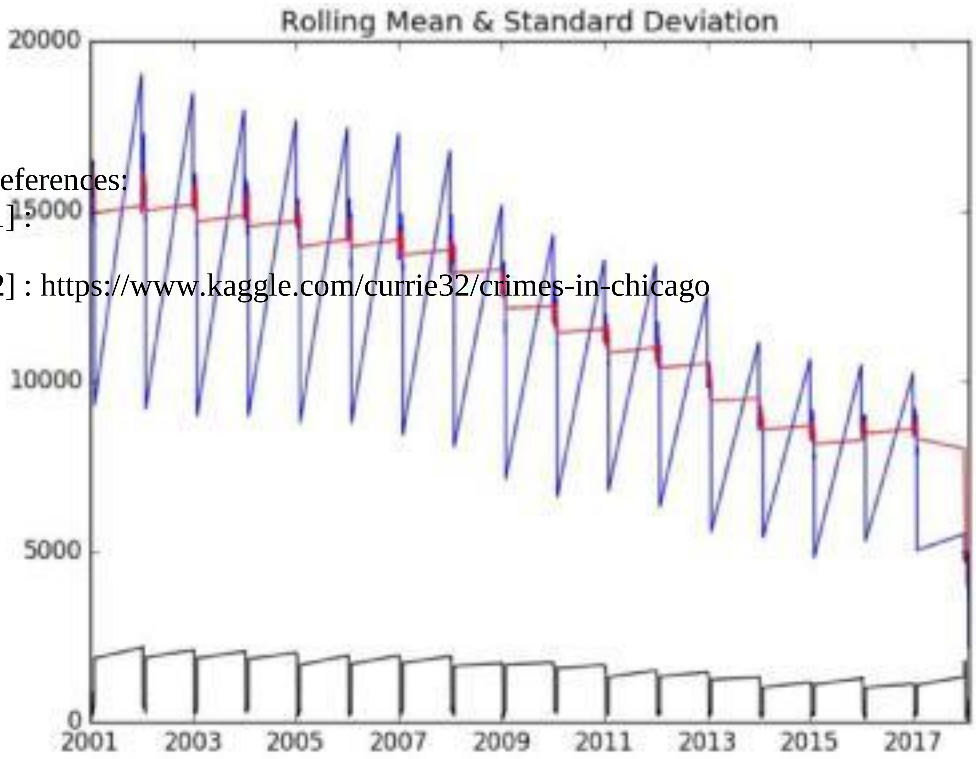
The no of arrests is less than the number of crimes committed. Maximum arrests were for Narcotics and prostitution

Analysis 3: Time series analysis and forecasting for Crime rates using ARIMA model

1. Removed Null values from dataset
2. Counted the number of crimes by date

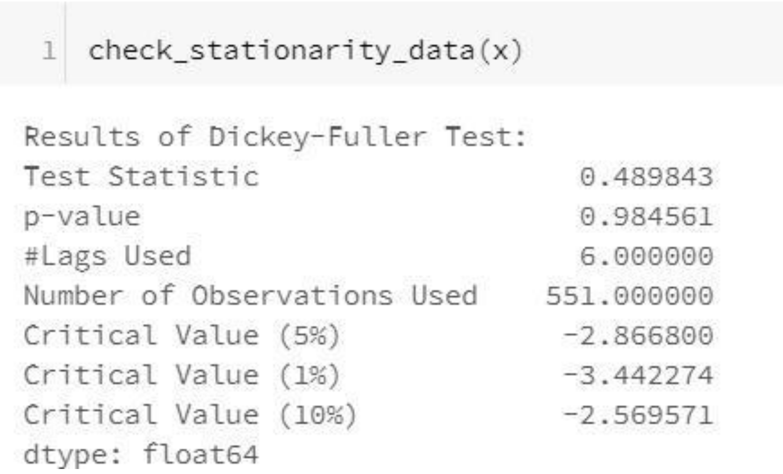
3. Perform Dickey-Fuller Test to check if the given crime time series data is stationary by plotting the rolling mean and standard deviation

4. Get p-value and t-statistics



References:
[1] :

[2] : https://www.kaggle.com/currie32/crimes-in-chicago



```
check_stationarity_data(x)
```

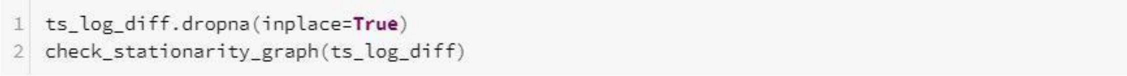
```
Results of Dickey-Fuller Test:
Test Statistic                   0.489843
p-value                          0.984561
#Lags Used                       6.000000
Number of Observations Used    551.000000
Critical Value (5%)             -2.866800
Critical Value (1%)             -3.442274
Critical Value (10%)            -2.569571
dtype: float64
```

We are getting p-value as 0.9 meaning our crime time series data is not stationary

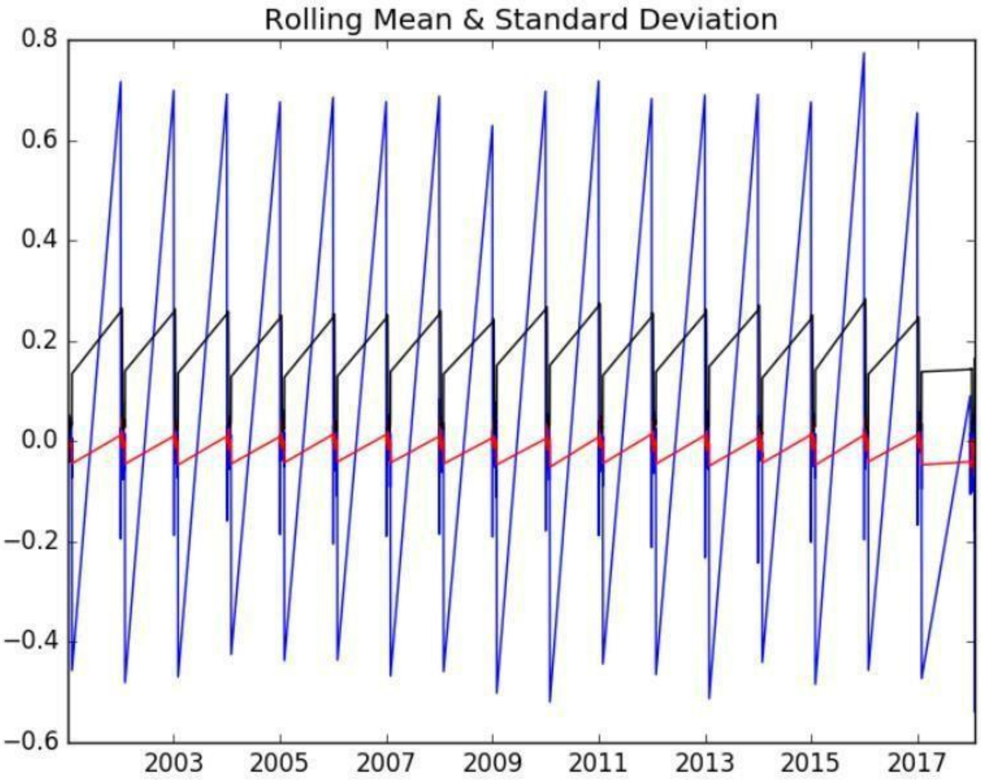
We will make the time series stationary by performing the following steps:

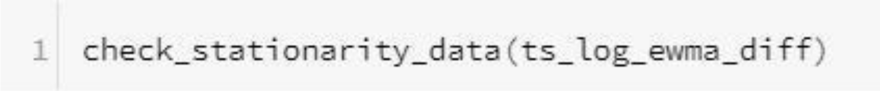
1. Plot data using log value of crimes count and plot moving avg crime count

2. Subtract moving avg crime count from log value of crime count. Plot time series log moving average difference

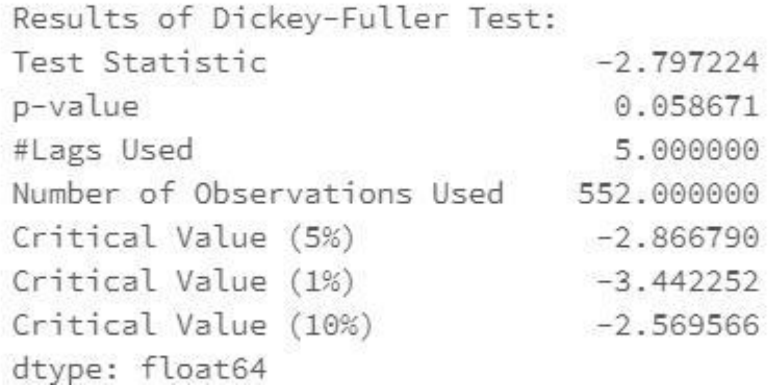


```
ts_log_diff.dropna(inplace=True)
check_stationarity_graph(ts_log_diff)
```





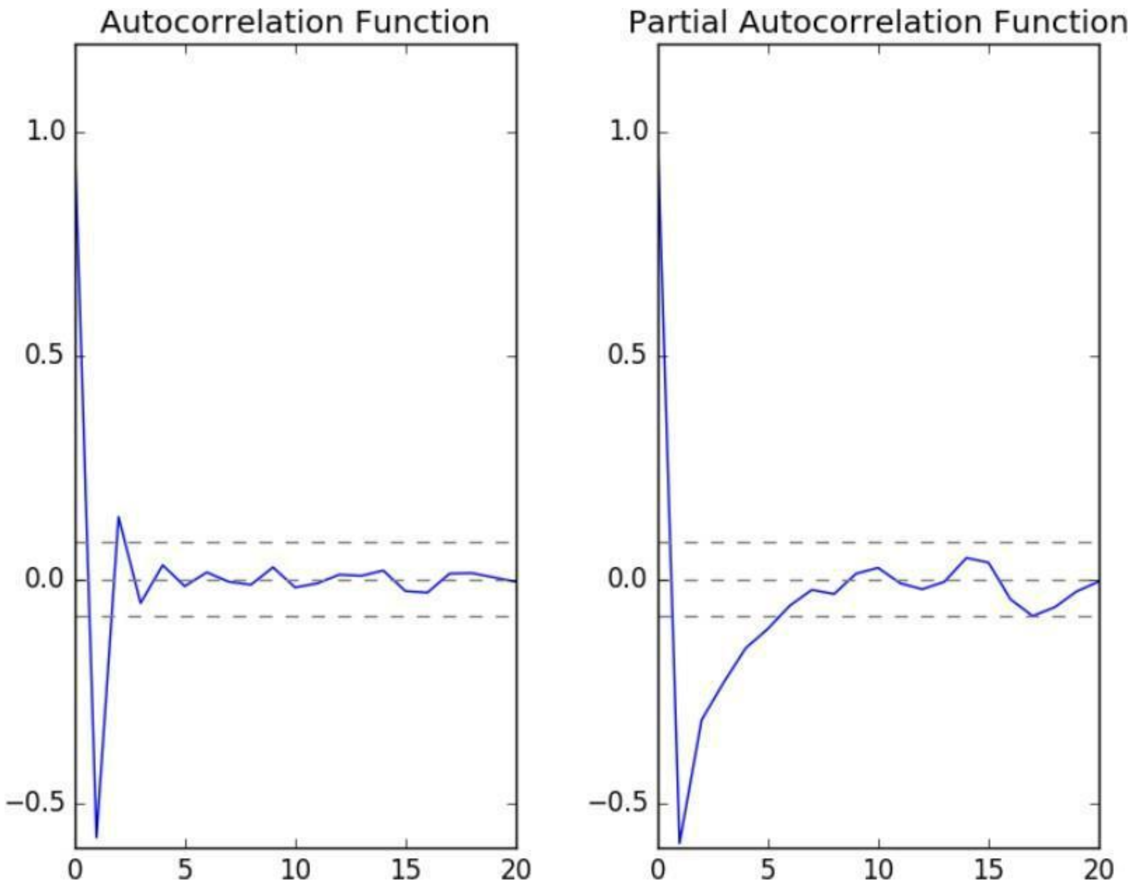
```
check_stationarity_data(ts_log_ewma_diff)
```



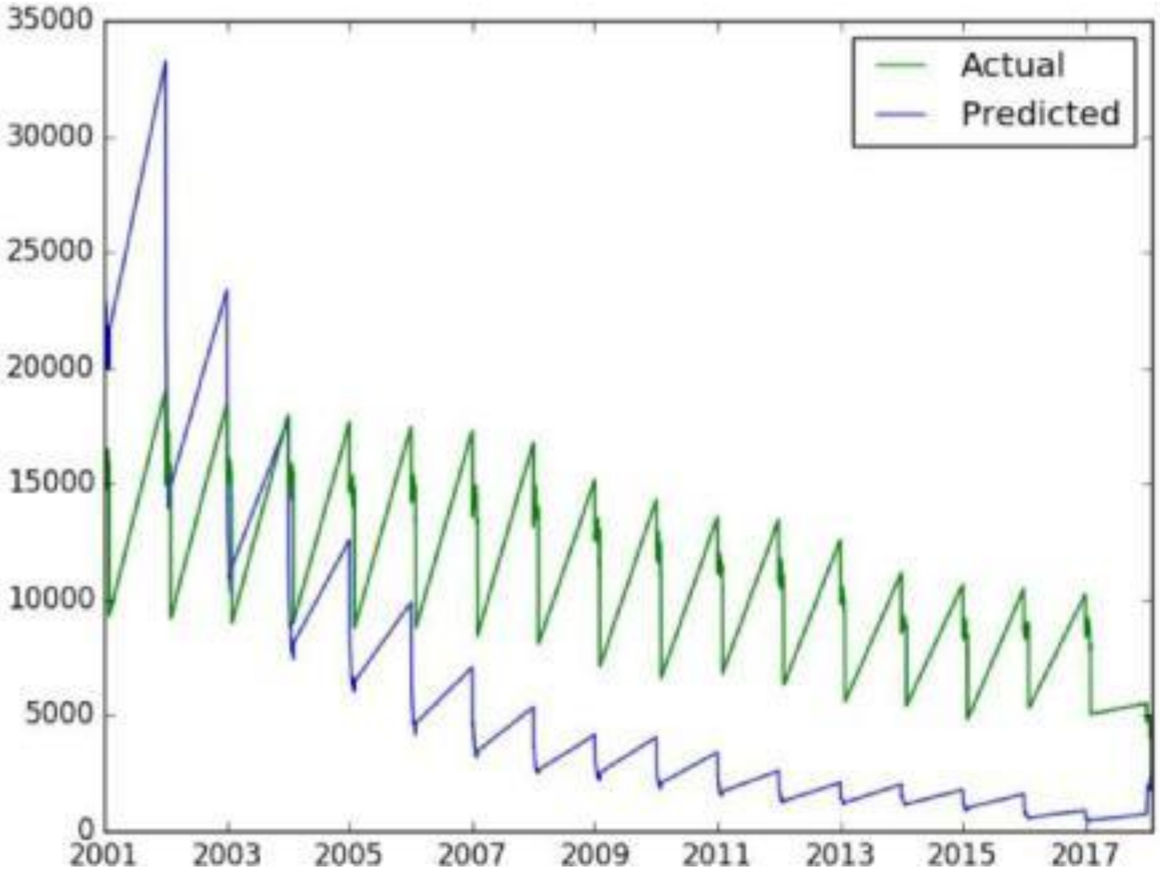
```
Results of Dickey-Fuller Test:
Test Statistic                  -2.797224
p-value                          0.058671
#Lags Used                       5.000000
Number of Observations Used    552.000000
Critical Value (5%)             -2.866790
Critical Value (1%)             -3.442252
Critical Value (10%)            -2.569566
dtype: float64
```

According to our statistics, Our p-value comes out to be less than 0.01, meaning our time series plot is stationary

ACF (autocorrelation function) and PACF (partial autocorrelation function) plots

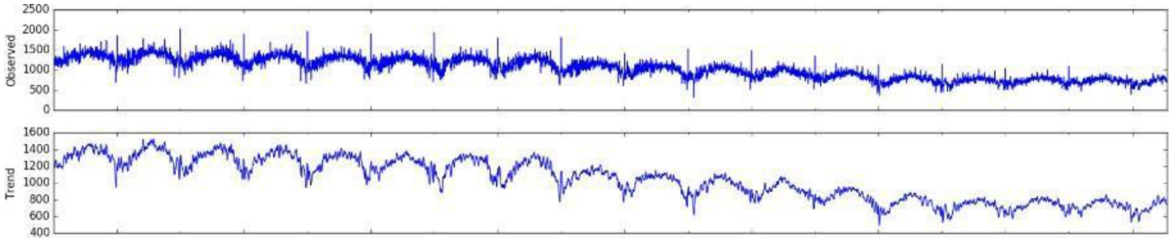


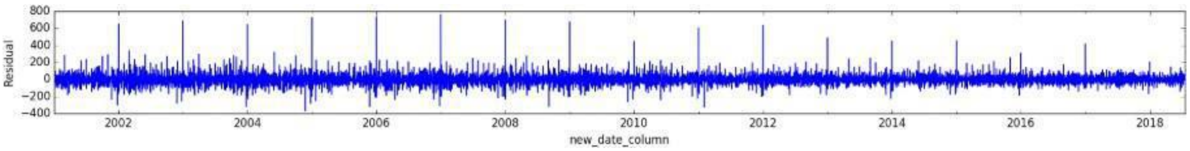
Plotting the graph of actual vs predicted crime count



The results we are getting using moving average time series gives us poor prediction results, so we try to improve the accuracy of the model using Seasonal ARIMA time series modeling

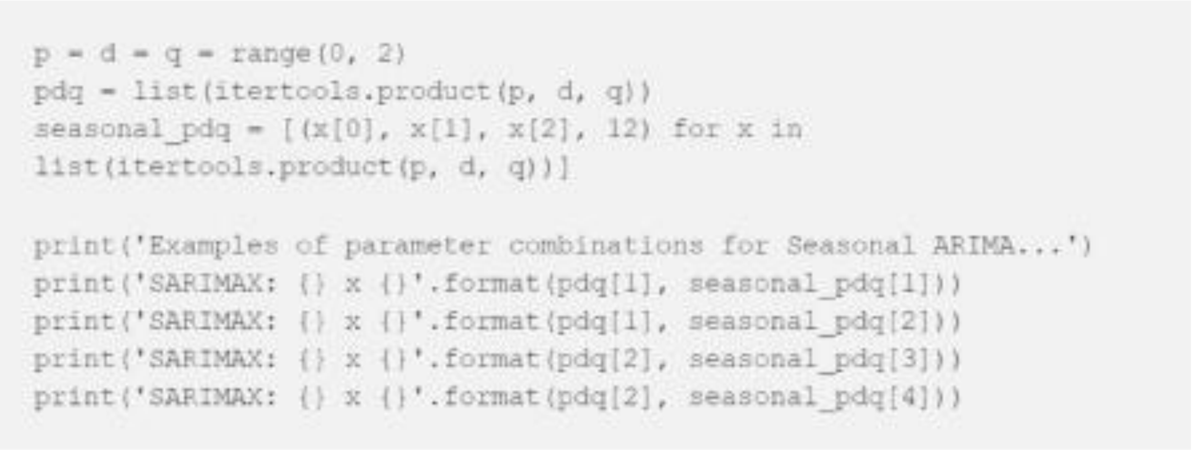
Analysis 4:
Visualize our data using a method called time-series decomposition that allows us to decompose our time series into three distinct components: trend, seasonality, and noise.



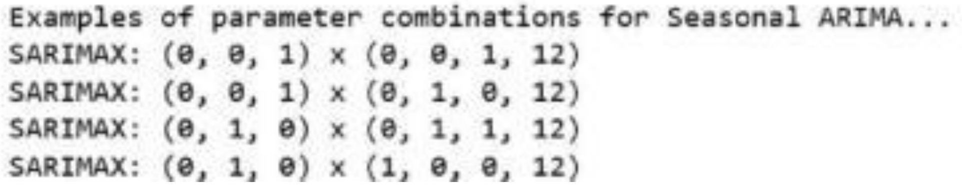


Analysis 5:
Time Series forecasting of crime counts using SARIMA (Seasonal ARIMA modeling)

ARIMA models are denoted with the notation ARIMA (p, d, q). These three parameters account for seasonality, trend, and noise in data

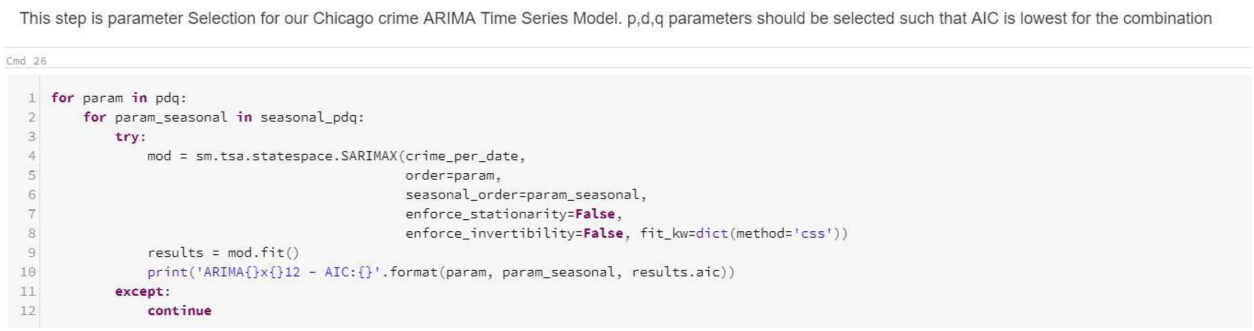


```
p = d = q = range(0, 2)
pdq = list(itertools.product(p, d, q))
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in
list(itertools.product(p, d, q))]

print('Examples of parameter combinations for Seasonal ARIMA...')
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))
```



```
Examples of parameter combinations for Seasonal ARIMA...
SARIMAX: (0, 0, 1) x (0, 0, 1, 12)
SARIMAX: (0, 0, 1) x (0, 1, 0, 12)
SARIMAX: (0, 1, 0) x (0, 1, 1, 12)
SARIMAX: (0, 1, 0) x (1, 0, 0, 12)
```

This step is parameter Selection for our Chicago crime count SARIMA Time Series Model. Our goal here is to use a “grid search” to find the optimal set of parameters that yields the best performance for our model.



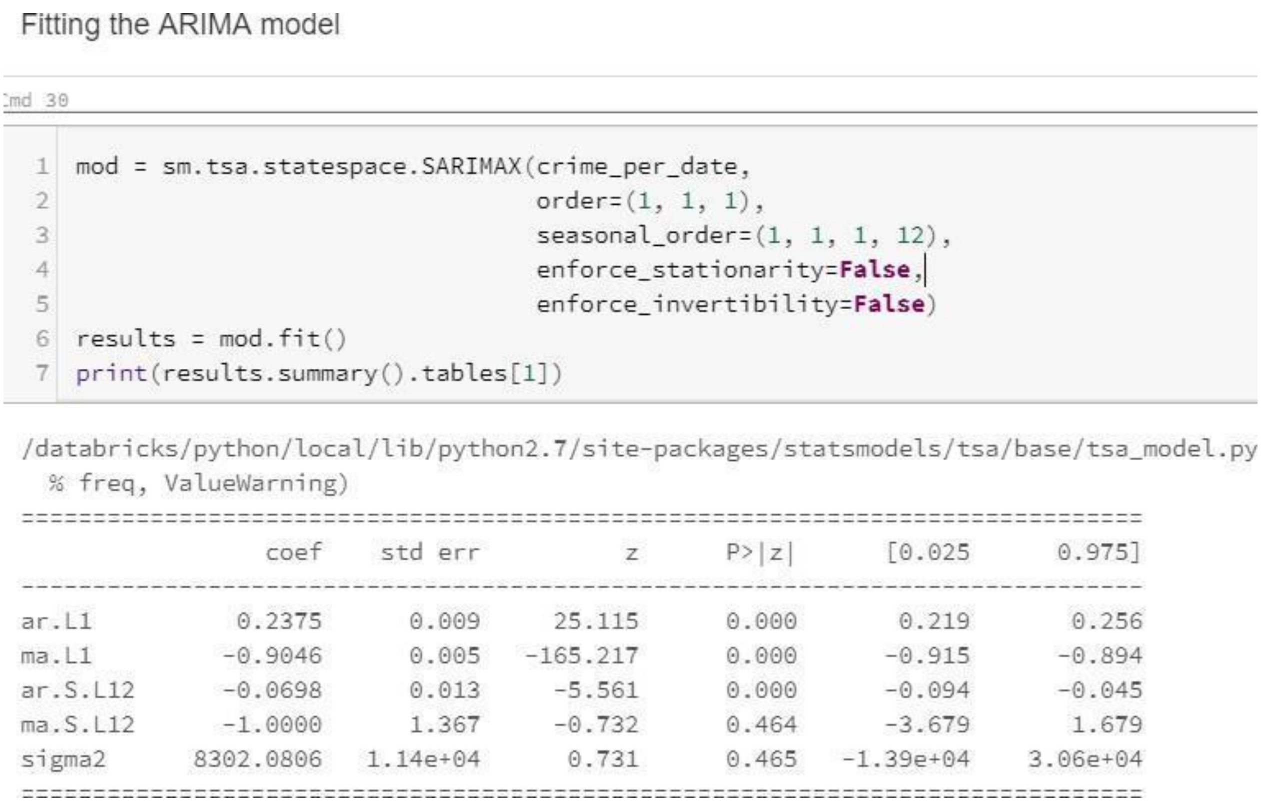
This step is parameter Selection for our Chicago crime ARIMA Time Series Model. p,d,q parameters should be selected such that AIC is lowest for the combination

Cmd 26

```
for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(crime_per_date,
                                            order=param,
                                            seasonal_order=param_seasonal,
                                            enforce_stationarity=False,
                                            enforce_invertibility=False, fit_kw=dict(method='css'))
            results = mod.fit()
            print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic))
        except:
            continue
```

SARIMAX (1, 1, 1)x(1, 1, 1, 12) yields the lowest AIC value of 757783.3. Therefore we should consider this to be optimal option.

Fitting the SARIMA model



Fitting the ARIMA model

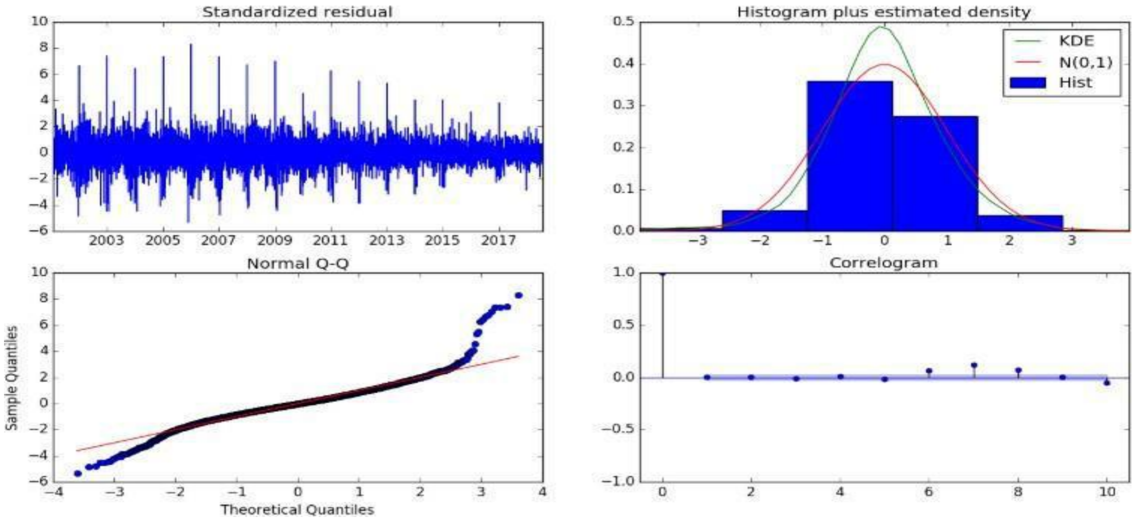
Cmd 30

```
mod = sm.tsa.statespace.SARIMAX(crime_per_date,
                                order=(1, 1, 1),
                                seasonal_order=(1, 1, 1, 12),
                                enforce_stationarity=False,
                                enforce_invertibility=False)
results = mod.fit()
print(results.summary().tables[1])
```

```
/databricks/python/local/lib/python2.7/site-packages/statsmodels/tsa/base/tsa_model.py
  % freq, ValueWarning)
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.2375      0.009     25.115      0.000       0.219       0.256
ma.L1         -0.9046      0.005   -165.217      0.000      -0.915      -0.894
ar.S.L12      -0.0698      0.013     -5.561      0.000      -0.094      -0.045
ma.S.L12      -1.0000      1.367     -0.732      0.464      -3.679       1.679
sigma2      8302.0806   1.14e+04      0.731      0.465   -1.39e+04    3.06e+04
==============================================================================
```

Plot statistics: p-value:0.46

Analysis 6: Plotting Model Diagnostics. Our histogram plot suggests that the model residuals are normally distributed.

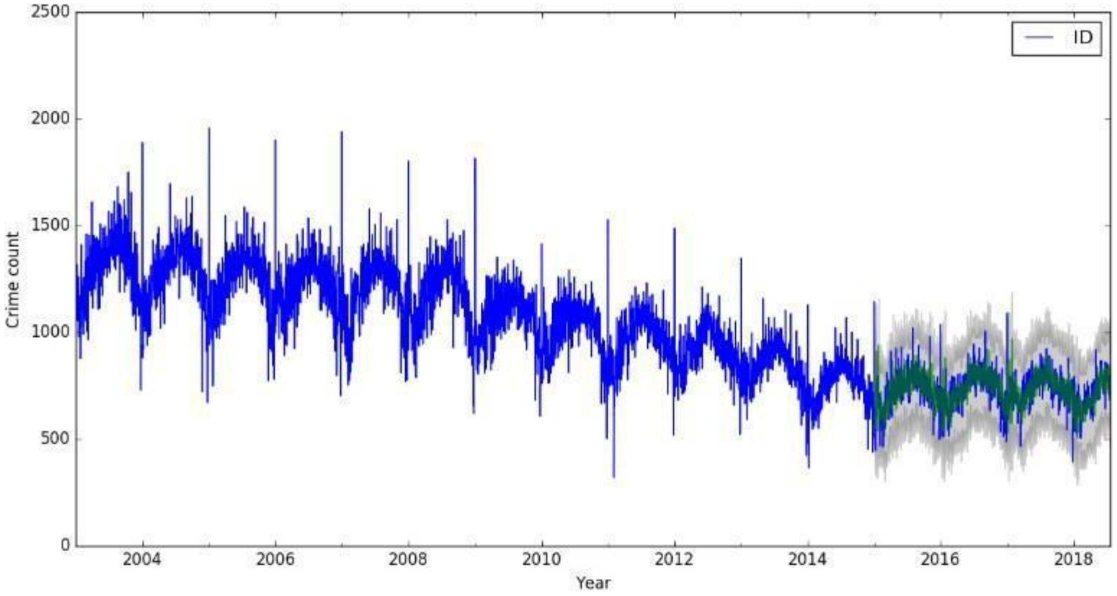


Key points:

- Q-Q plot denotes normal distribution of data
- Estimated the probability density function (PDF) of a continuous random variable and correlation statistics by plotting correlogram showing serial correlation that changes overtime
- The correlations are very low and doesn't seem to have a pattern
- On the y axis is the autocorrelation. The x axis tells you the lag. So, if x=1 we are looking at the correlation of 2018 with 2017, 2017 with 2016, etc. If x=2, we have a lag of 2 and we are looking at the correlation of 2018 with 2016, 2017 with 2015, etc.

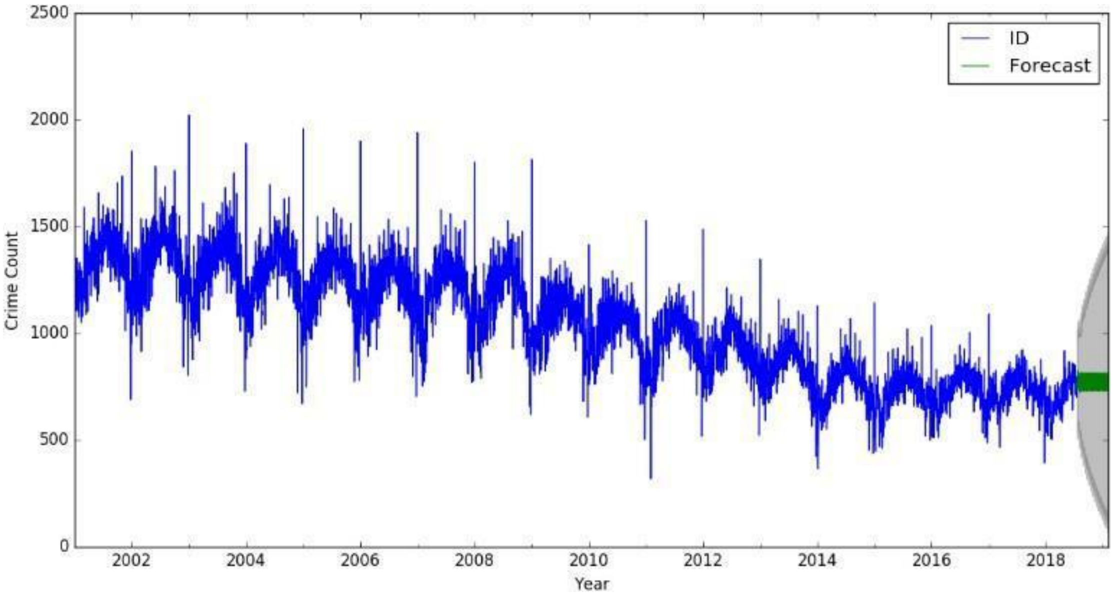
Analysis 5: Validating Forecasts

To help us understand the accuracy of our forecasts, we compare predicted crimes to actual crimes of the time series, and we set forecasts to start at 2015–01–01 to the end of the data.



Graph of actual crimes vs predicted crimes

Visualizing future forecasts for crimes in Chicago. Generating confidence intervals which grow larger as we move further out into the future. No of steps considered:500



Conclusion:

1. Most occurring types of crimes by location, hour and week
2. Increasing criminal activities consisting of thefts, robberies etc over the years from 2001-present and predicted the occurrence of crimes
3. Calculated crime and arrest rates using Logistic Regression with 69% accuracy
4. Time series Forecasting using ARIMA and improving model accuracy using Seasonal time Series SARIMA model

References:

https://towardsdatascience.com/an-end-to-end-project-on-time-series-analysis-and-forecasting-with-python-4835e6bf050b

https://datascienceplus.com/spark-dataframes-exploring-chicago-crimes/

https://datascienceplus.com/spark-dataframes-exploring-chicago-crimes/

https://github.com/ajitkoduri/Chicago-Crime-Analysis/blob/master/Chicago%20Sex%20Crimes%20Analysis.ipynb