Exploratory Data Analysis, Modelling and Forecasting on NYC crime Time Series data

Inês Silva, Maria Miguel Ribeiro, Renatha Vieira

Faculdade de Ciências da Universidade do Porto

1 How to handle a Time Series

When dealing with time series data, it is crucial to measure the dependencies between observations rather than simply summarize the data, to understand the dynamics and the way data changes over time. This way we can apply mathematical models to approximate and replicate patterns, to further forecast and project future outcomes.

2 Understanding the data

We started by understanding the main components of the data and how to best represent it. We opted for a dataset about NYC Crime Data from NYC Police Department [1] to perform exploratory data analysis. The data set is composed by 25 variables (columns) and 361740 entries (rows).

2.1 Variables

CMPLNT_NUM
CMPLNT_FR_DT
CMPLNT_FR_TM
CMPLNT_TO_DT
CMPLNT_TO_TM
RPT_DT
KY_CD
OFNS_DESC
PD_CD
PD_DESC
CRM_ATPT_CPTD_CD
LAW_CAT_CD
JURIS_DESC
BORO_NM
ADDR_PCT_CD
LOC_OF_OCCUR_DESC
PREM_TYP_DESC
PARKS_NM
HADEVELOPT
X_COORD_CD
Y_COORD_CD
Latitude
Longitude
Lat_Lon

2.2 Choosing a Time Series

Due the large number of attributes in the dataset, we decided to aggregate the data, not distinguishing between the different types of crime. We have selected only two columns that we considered of interest to this analysis: `'NUM_OCURR'` and `'CMPLNT_FR_DT'`. The column `'NUM_OCURR'` represents the sum of total occurrences of crimes in the same day, and the `'CMPLNT_FR_DT'` contains the date of when which crime as occurred.

Next, the data was grouped by month in order to facilitate the exploration and analysis of the time series on a monthly basis. We decided to take into account the period between 2006 and 2014 due to the exponential growth of the number of crimes occurrences after 2016.We decided to discard the data from years before 2006 because prior to that year, we did not have a total of 12 annual observations corresponding to each month. The previous mentioned steps have been performed in Python, and the file is in the form ipynb.

3 Converting data to time series

From this stage onward, all the code was executed in R, with some exceptions. We started by converting the dataset into a time series.

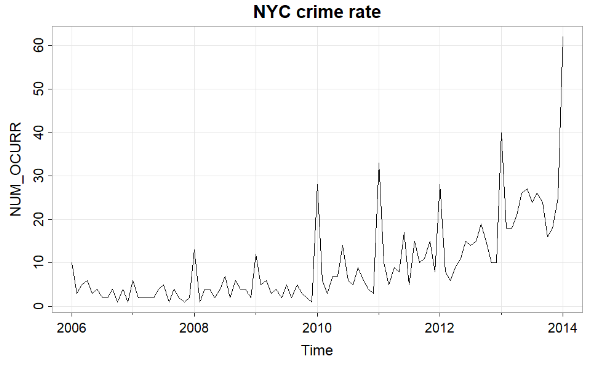


Fig. 1: NYC crime occurrences between 2006 and 2014

This dataset illustrates a noticeable seasonal pattern with a increasing trend over time. As the trend increases, the cycles also increase meaning that there is a positive correlation between trend and seasonality. This suggests that the cycles follow the pattern set the trend and become more pronounced or frequent as the trend grows stronger. It can be identified a seasonal pattern as a steeper peak at the beginning of each year, followed by a gradual but steady increase in the annual average. This happens due to the exponentially growth of the number of crime occurrences after middle 2013 and beginning of 2014.

3.1 Choosing the model

In this time series, the trend is not linear, and the amplitude of the seasonal cycles increases with the trend. We can observe that there is heteroscedasticity in the data, meaning a change in variance, so we need a multiplicative model to represent the relationship between the components. The non-linearity is very important when choosing each method for the data analysis.

$$y_t = T_t \times S_t \times R_t \tag{1}$$

4 Trend

When observing Figure 5, it becomes clear to us the existence of a trend that increases over time. The amplitude of the seasonal cycles generally grows with the trend, with a significant increase from the year 2013. In the case we are analyzing, which is the number of crime occurrences, this means that criminality has a tendency to increase considerably over time, and after the middle of 2013, the growth takes on larger proportions. As explained earlier, due to the non-linear nature of the trend, it is necessary to use non-linear models to estimate and remove it from the time series.

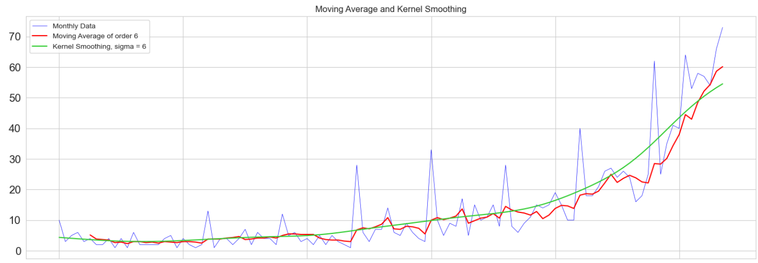


Fig. 2: Estimating the trend using Moving Average and Kernel Smoothing

The Figure 2 shows the comparison between two methods, the moving average of order 6, represented in red, and the kernel smoothing with bandwidth (sigma) equals to 6, in green. The kernel smoothing method appears to be the best candidate to estimate the trend, as it seems to predict the trend more accurately than the moving average one. Various methods were considered to remove the trend, aiming to better understand which one was more effective in capturing the trend of the data.

5 Seasonality

There are recurring patterns or cycles in the data that occur at regular intervals. For a clear representation of the seasonal patterns we used the following plots:

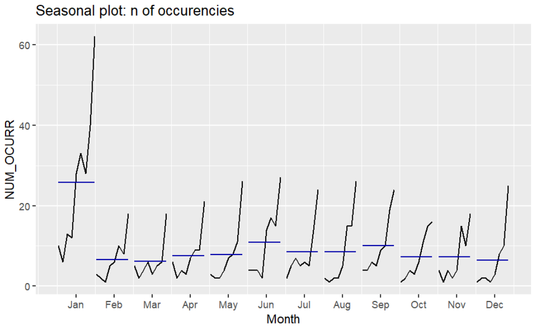


Fig. 3: Seasonal month plot

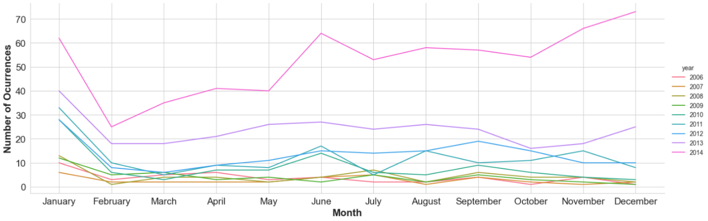


Fig. 4: Seasonal plots from 2006 to 2014

The blue lines in Figure 3 represent the mean of the corresponding months. It can be seen that the mean is not constant and changes over time, and the variables are not underlying a common mean. This happens because since we are on the presence of a trend in the data, the mean is meaningless. In Figure 4, we can identify repetitive patterns, where crime occurrences increase during certain months and decrease during others. For example, we can see that every year, the number of crime occurrences are highest during January, dropping greatly in February, having a peak in June and staying relatively low for the rest of the year, except for the years of 2013 and 2014, since the number of occurrences still grow after decreasing in July.

6 Box-Cox Transformations

To even out the fluctuations within the series, we employ Box-Cox transformations, with one specific approach being to take the logarithm of the data.

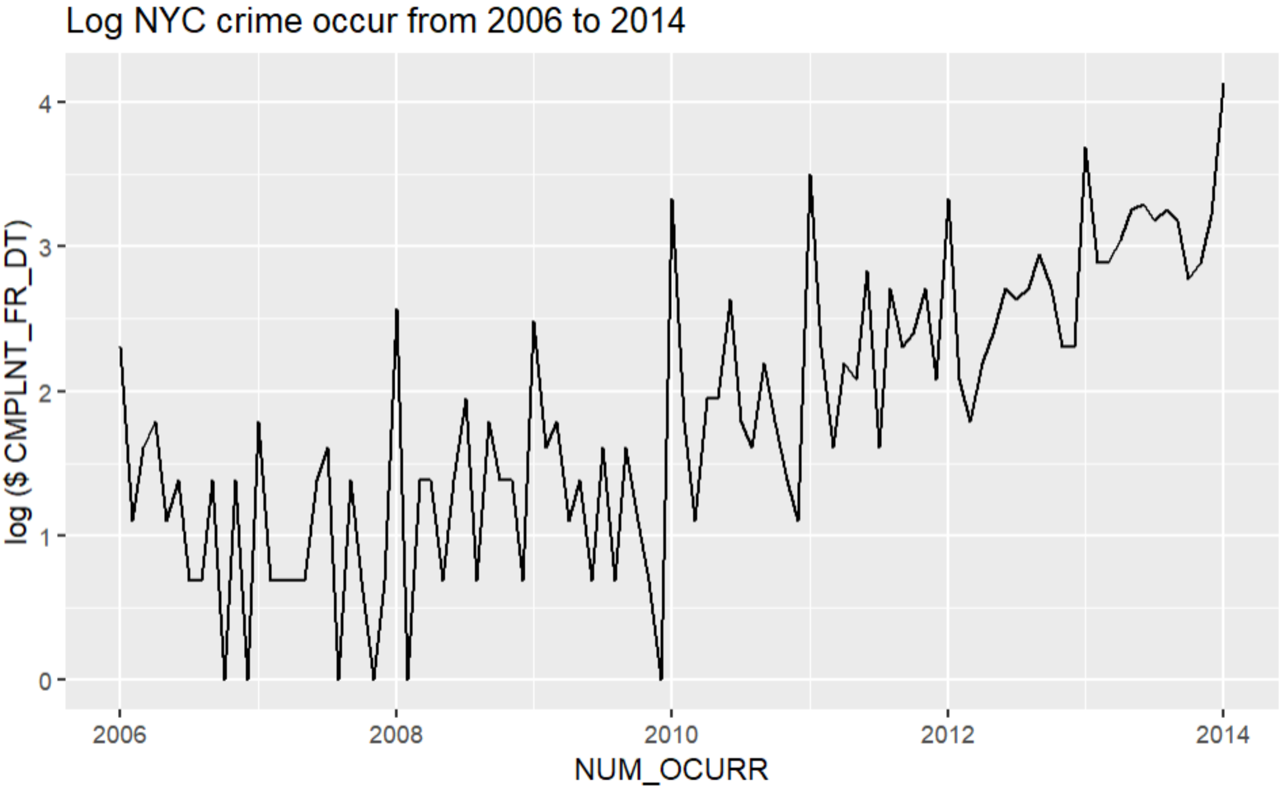


Fig. 5: Log Transformation

This process aimed to achieve a more symmetric and normally distributed dataset, addressing concerns such as heteroscedasticity or non-normality.

7 Time Series Decomposition

7.1 STL decomposition

The Seasonal Decomposition of the Time Series by Loess was implemented in R with the stl() function. This method breaks down a time series into three components: seasonal, trend, and residuals or remainder.

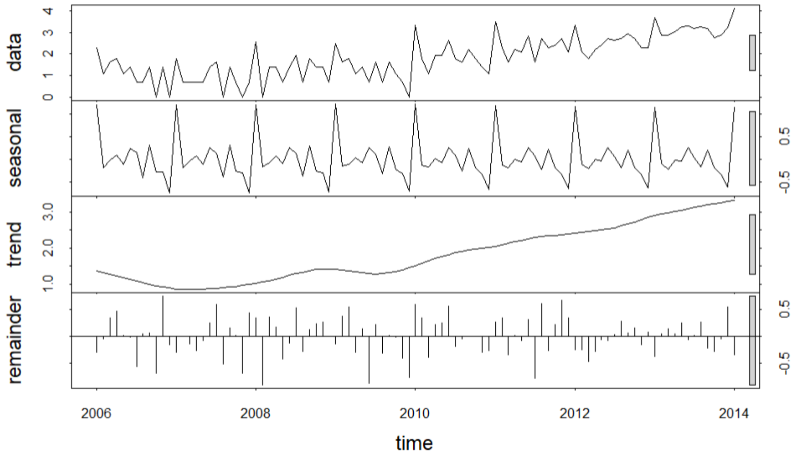


Fig. 6: STL decomposition of the time series

7.2 Removing the Trend

7.2.1 Difference filter

We chose to filter the time series with the simple difference filter to analyze the increments or changes in the number of crime occurrences at consecutive time points. We applied this filter to stabilize the mean of the time series by reducing or removing the trend.

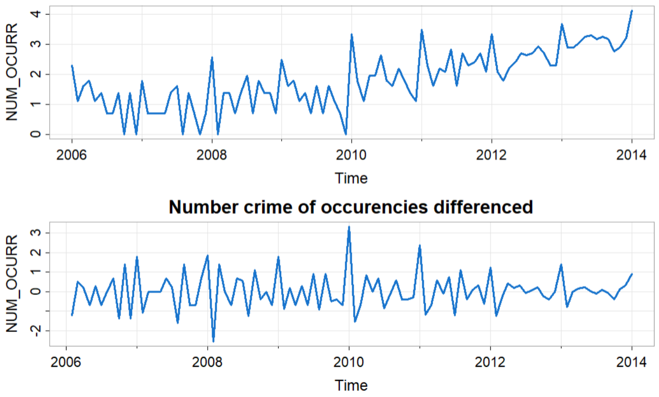


Fig. 7: Detrended Log NYC Crime Data

From this graph we can see that at the beginning of every year there is a peak in the crime rate, which can mean that there was a big increase or decrease of that rate.

7.2.2 Using smoothing techniques An estimation of the trend was performed by using kernel smoothing techniques, as shown in the Figure below. We decided to use a bandwidth (sigma) equal to 6 because because it is a good value to ensure that the smoothing is not too sensitive to data fluctuations.The Figure below shows the original series in red and the series without the influence of the trend in blue.

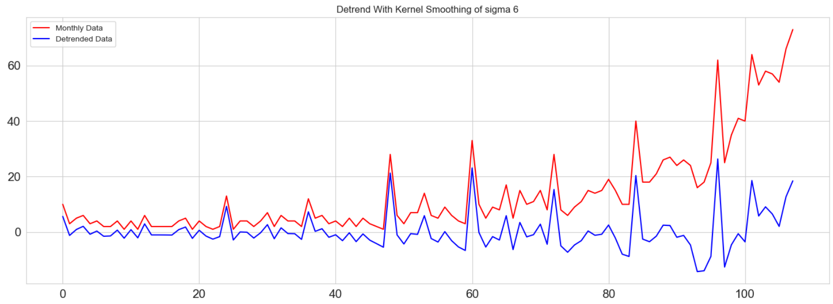


Fig. 8: The time series and the detrended time series using Kernel Smoothing

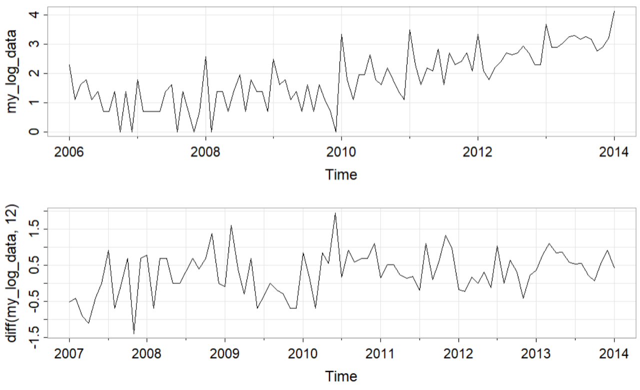
As the method is used with the aim of removing the trend - subtracted the kernel estimation from the time series - the second graph, in blue, displays the residual component, which may contain seasonality and residuals. As we can observe, it remains within a range of numbers and no longer demonstrates a growth in occurrences over time.

Removing the trend can make it easier to identify seasonal patterns and short-term behaviors in the series, and its primary goal is to isolate and highlight the underlying patterns and information in the time series.

7.3 Removing seasonality

We used the seasonal difference operator to extract the underlying components of seasonality from the time series. By employing this technique in our time series, we were able to understand the effects of long-term trends from the seasonal patterns present in the data, providing us better understanding of the individual contributions of trend and seasonality.

We stop filtering the data after applying a first-order differencing, as there was left no evidence of a clear pattern over time.



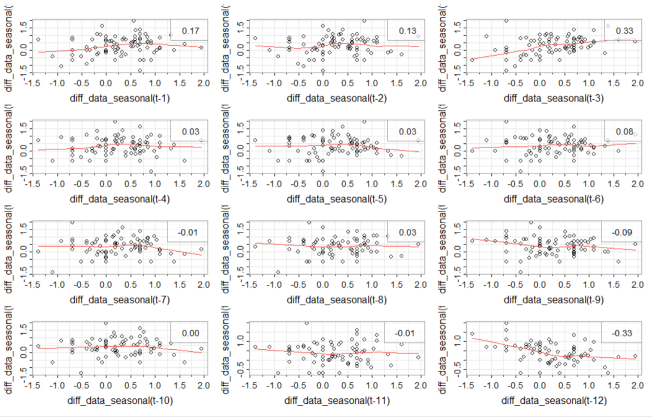


Fig. 9: Seasonal month plot

Fig. 10: Lagplots representing Deseasonalized Data

8 Remainder

What remains after removing the trend and seasonality is the remainder, the residuals of the times series. The goal is to have independent residuals, not predictable and with no clear pattern over time, since this means that the other components were efectively captured.

The Figure 11 represents the time series without any transformation performed. Starting from the year 2010, there is a notable increase in the number of criminal occurrences, followed by a decline at the beginning of the

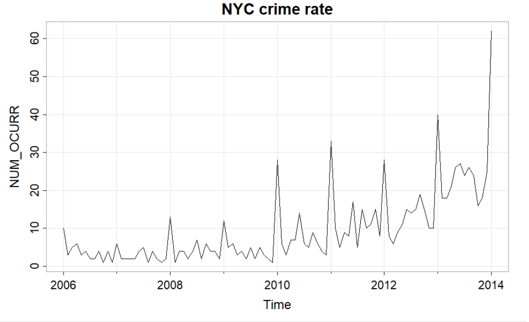


Fig. 11: Time series with trend and seasonality

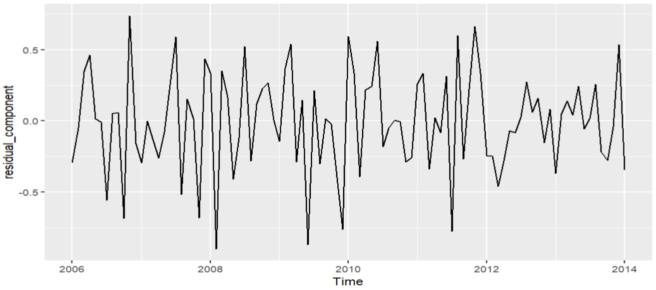


Fig. 12: Remainder of the time series

year 2011 and a subsequent rise at the beginning of the year 2012. This pattern repeats over time and intensifies dramatically at the end of the year 2013, which records the highest number of occurrences in 7 years. As for the Figure 12, it can not be identified any predictive pattern, which means that each observation is independent of the others, resembling white noise.

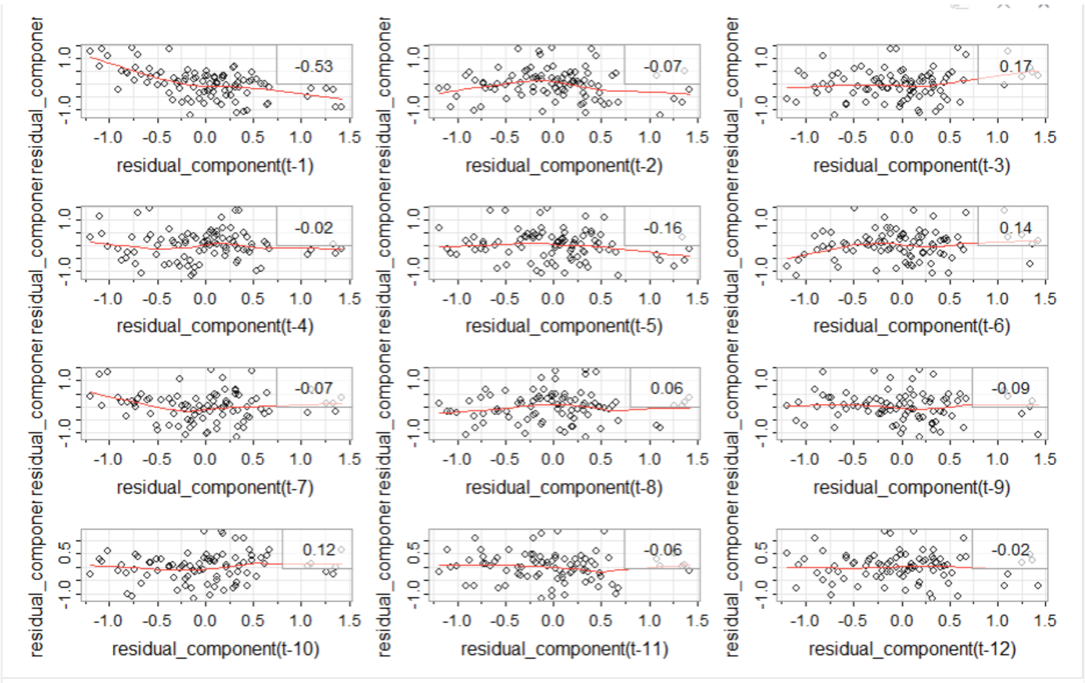


Fig. 13: Lagplots of the remainder

As seen in Figure 13,the observations seem to be independent. The correlation is non significant in any lag, representing a pattern similar to the behaviour of white noise.

We can see this from the ACF and PACF graphs

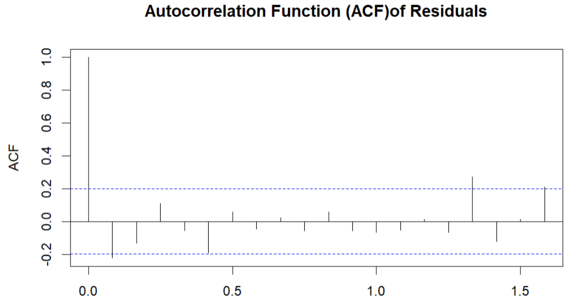


Fig. 14: ACF of the time series

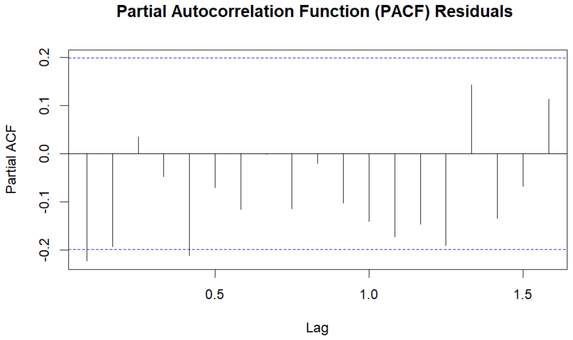


Fig. 15: PACF of the time series

In graph 14 it can be identified correlation only in lag 1. In Figure 15, there is only a little correlation up to middle of lag 1, but still it can be considered non significant. Gathering all of this information, we were able to draw some important conclusions.

9 Final Conclusions for the EDA

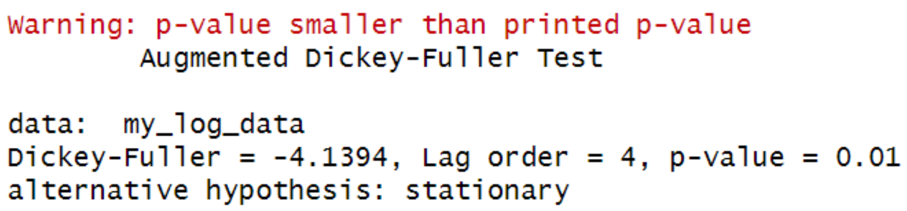
After the exploration and analysis of the number of crime occurrences in NYC, we were able to understand a lot of important insights. The dataset initially shows a notable increase in criminal tendencies over time, with seasonality closely following the overall trend. The dataset exhibits higher crime rates in January and June, while February experiences a drastic drop of the crime rates, suggesting the presence of a clear and strong seasonal pattern. Notably, by analysing the remainder, it can be seen drastic rises or abrupt falls in the number of crime occurrences depending on the lag. The peaks suggest the occurrence of unusual or extraordinary events during these periods, leading to a substantial change in the number of reported crimes and probably related to some events that occurred in NYC in specific moments that we can not predict.

The detrending and deseasonalizing methods applied have generally performed well in capturing the systematic components of the data, as we can evidence from the ACF and PACF plots of the residual component. In the ACF plot, it can be noticed a significant correlation only at lag 0, which suggests a strong correlation between observations at the same time point but not at subsequent lags. This indicates that observations are independent at different lags and do not exhibit a systematic correlation over time, except at the same time point. The alternation between positive and negative values suggests that there is no clear trend of correlation in observations, and the pattern may be more random. Relatively to PACF there is only a slight correlation at lag 0, which is negative, and then there is no more significant correlation. The alternation of correlation from negative to positive at certain points suggests a specific pattern that may indicate the presence of an Autoregressive (AR) component in the model. Due to this, it may be necessary to consider more complex models that take into better account this temporal dependency such as ARIMA models (Autoregressive Integrated Moving Average). Due to this, all the differentiations applied to the time series in these recent sections turned out to be purely exploratory, contributing to a better understanding of the characteristics of this specific series. Therefore, the data used in the following chapters is the one obtained from the data transformed with the natural logarithm only.

10 Modelling

10.1 Augmented Dickey-Fuller Test

We tested for the presence of a unit root in the time series dataset with the Augmented Dickey-Fuller (ADF).



```
Warning: p-value smaller than printed p-value
        Augmented Dickey-Fuller Test

data:  my_log_data
Dickey-Fuller = -4.1394, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary
```

Fig. 16: Augmented Dickie Fuller Test

As we can see in Figure 17, the p-value is inferior to the 0.05 which means that we can reject the null hypothesis, and our time series is proven stationary.

10.2 Training set and Test set

To model the data and evaluate, we need to split the dataset in two parts, the training set and the test set. The training set will represent 80% of the dataset, beginning in January 2006 and ending in June 2012. This set will be used to train the model. The rest of the dataset, from June of 2012 to January of 2014, is saved for the test set, that will be used to see if the trained model has a good performance.

10.3 Building a model

When creating a model it is important to understand which models could best adjust to the data. This is accomplished by looking at the ACF and PACF graphs of the training set, as well as the AIC and BIC criterion for each model.

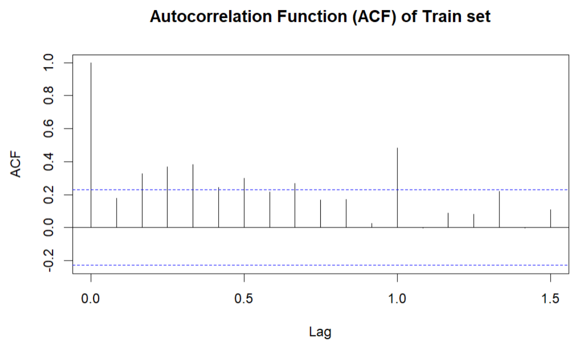


Fig. 17: ACF of the training set

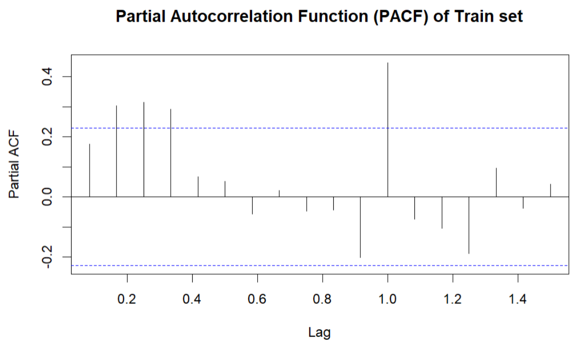


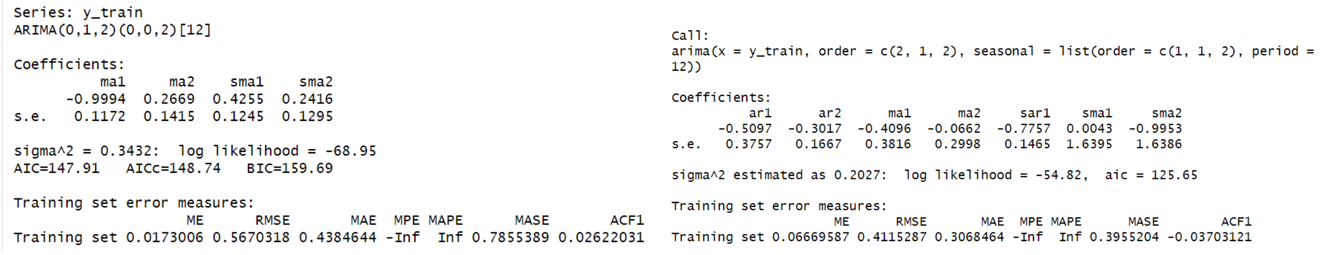
Fig. 18: PACF of the training set

The analysis of the autocorrelation (ACF) and partial autocorrelation (PACF) plots of the training set reveals intriguing patterns in the temporal dependence of the data. In the ACF plot it can be seen correlation of 1 only at lag 0, indicating a strong correlation between an observation and itself. Furthermore, the correlation gradually increases to a maximum of 0.4, after which it starts to decrease again, until lag 1, where we see a peak. After that the values are statistically insignificant.

When examining the PACF, it is notable that correlations are generally slightly higher than 0.2 up to half of lag 1. In lag 1, it can be seen a peak of more than 0.4, indicating a direct influence between the observation in this lag from the actual observation. As stated before, non-significant correlation alternating between positive and negative values at subsequent lags indicate the presence of an oscillating behaviour mostly due to the inherent seasonal pattern in the data. From lag 1 onwards, there is no longer any significant correlation, suggesting that the majority of temporal dependencies have been modeled up to that point. It also suggests that the existing temporal dependencies have been effectively addressed in the modeling process.

To find the best parameters to describe our data, we used the function **autoarima()**. This function indicates automatically the best ARIMA model for the time series, based on the AIC criterion. The result was an

ARIMA(0,1,2)(0,0,2)[12]. Despite the model suggested by this function, it does not always prove to be the best option. Due to this, and since the model did not yield optimal results, a model with both seasonal and no seasonal autoregressive component was found to be more suitable. The corresponding parameters of the chosen model are ARIMA(2,1,2)(1,1,2)[12]. The following graphs illustrate the parameters of these two models, with the decisive criterion for choosing between them being the one that exhibited lower AIC.



```
Series: y_train
ARIMA(0,1,2)(0,0,2)[12]

Coefficients:
          ma1     ma2    sma1    sma2
      -0.9994  0.2669  0.4255  0.2416
s.e.   0.1172  0.1415  0.1245  0.1295

sigma^2 = 0.3432:  log likelihood = -68.95
AIC=147.91   AICc=148.74   BIC=159.69

Training set error measures:
                    ME      RMSE       MAE  MPE MAPE      MASE       ACF1
Training set 0.0173006 0.5670318 0.4384644 -Inf  Inf 0.7855389 0.02622031
```

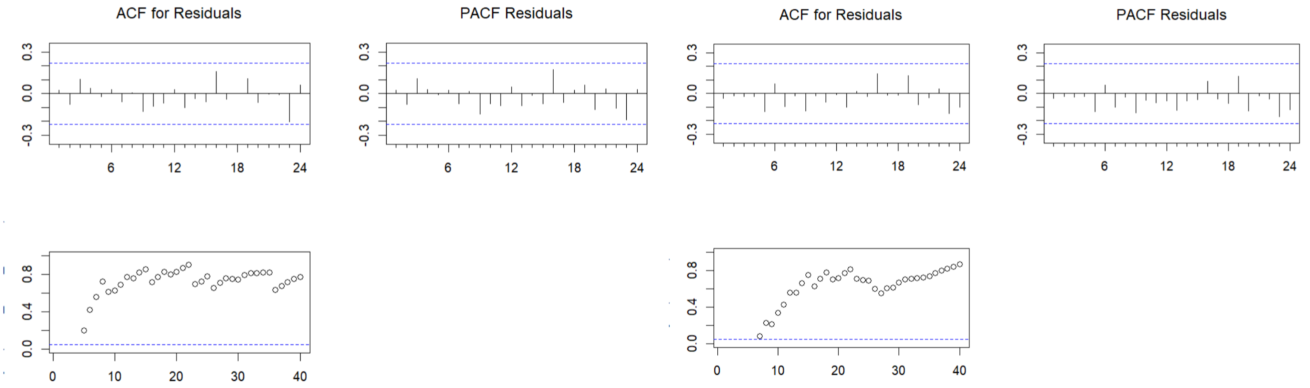
(a) ARIMA(0,1,2)(0,0,2)[12]

```
Call:
arima(x = y_train, order = c(2, 1, 2), seasonal = list(order = c(1, 1, 2), period =
12))

Coefficients:
          ar1      ar2      ma1      ma2     sar1    sma1     sma2
      -0.5097  -0.3017  -0.4096  -0.0662  -0.7757  0.0043  -0.9953
s.e.   0.3757   0.1667   0.3816   0.2998   0.1465  1.6395   1.6386

sigma^2 estimated as 0.2027:  log likelihood = -54.82,  aic = 125.65

Training set error measures:
                     ME      RMSE       MAE  MPE MAPE      MASE        ACF1
Training set 0.06669587 0.4115287 0.3068464 -Inf  Inf 0.3955204 -0.03703121
```

(b) ARIMA(2,1,2)(1,1,2)[12]

Fig. 19: Comparison of parameters from two different models trained with training set

To calculate the goodness-of-fit of the choosen model, we performed the Ljung-Box test. As showed in the figures below, the p-values for both models are above the confidence level. Although one p-value for the ARIMA(2,1,2)(1,1,2)[12] is very close to this confidence level, it is not significant. Therefore, we opted for this model as it already exhibited the lowest information criteria.



(a) ARIMA(0,1,2)(0,0,2)[12]

(b) ARIMA(2,1,2)(1,1,2)[12]

Fig. 20: Ljun-Box test for the different fitting models

The p-values associated with the tests are, in general, very high for the two models, indicating that we cannot reject the null hypotheses. Therefore, as stated before, we can conclude that the observations exhibit no significant correlation between them. This means that the observations are independent, aligning with our intended demonstration.

11 Forecasting

Now that the model was chosen and evaluated for the training set, we forecast the values for the test set, and verify if the predicted values are in line with the observed values.

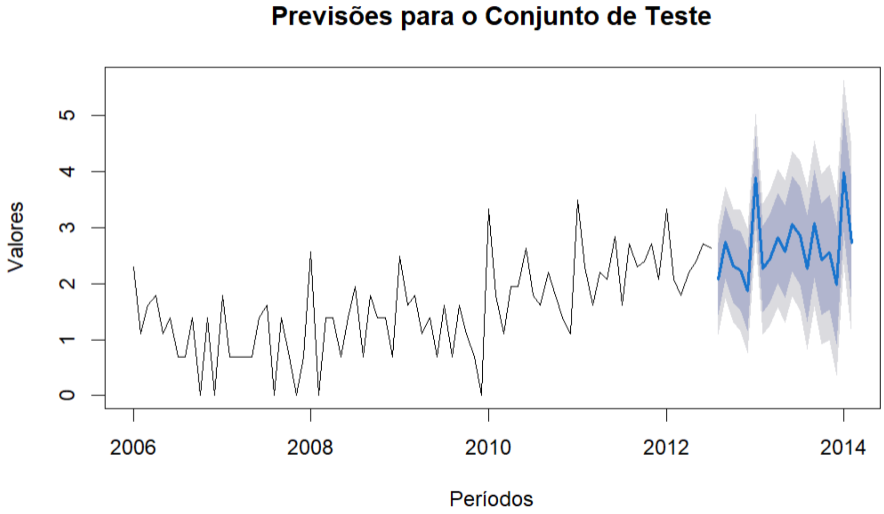


Fig. 21: Forecast with ARIMA(2,1,2)(1,1,2)[12] for the Test Set

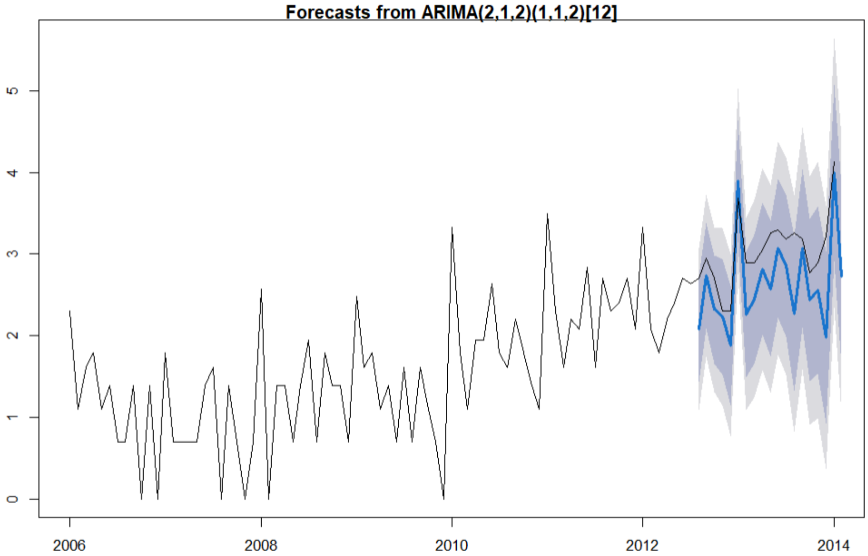
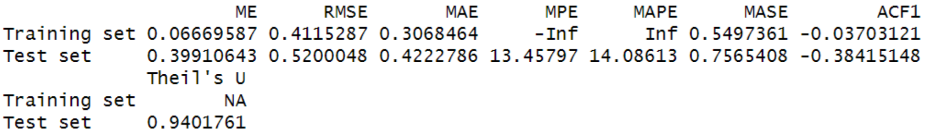


Fig. 22: Predicted and Observed Values with ARIMA(2,1,2)(1,1,2)[12] for the Test Set

Figure 22 shows the observed data from 2006 to 2014 in black, and the predicted values for each month, from June 2012 to January 2014, in blue. As we can see, there is some discrepancy between the observed and predicted values, but the results are promising.

11.1 Final Evaluation

To choose which of the models is more indicated we have to look at various metrics. The most popular are the mean error (ME), root mean squared error (RMSE), mean absolute error (MAE) and the Theil's U.



```
                     ME      RMSE       MAE      MPE     MAPE      MASE        ACF1
Training set 0.06669587 0.4115287 0.3068464     -Inf      Inf 0.5497361 -0.03703121
Test set     0.39910643 0.5200048 0.4222786 13.45797 14.08613 0.7565408 -0.38415148
             Theil's U
Training set        NA
Test set     0.9401761
```

Fig. 23: Comparison of the error metrics obtained with ARIMA(2,1,2)(1,1,2)[12] for the Train and Test sets

In the training set, the mean error (ME) is effectively centered around zero, indicating a satisfactory alignment between predicted and actual values. Key metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) further affirm a commendable accuracy, particularly when contextualized with the dataset's magnitude. The Mean Absolute Scaled Error (MASE), which gauges the model's accuracy against a naive forecast, holds a value of 0.5497361. This suggests a reasonable precision in predictions, with lower MASE values indicating enhanced accuracy.

However, in the test set, all metrics increased a little bit but stood still close to zero. Some noteworthy values for Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are observed with 0.4115287 and 0.3068464 in training, respectively, and 0.6170978 and 0.5271292 in testing. Additionally, error percentage metrics (MPE and MAPE) caution is advised when interpreting Mean Percentage Error (MPE) and Mean Absolute Percentage Error (MAPE) due to the presence of Inf and -Inf values due in instances involving zero values in the series. The ACF1 values are -0.03703121 in training and 0.38415148 in testing. Since this values are close to zero it suggests that the residuals at lag 1 are not significantly correlated or have little residual autocorrelation, indicating that the model has successfully captured the temporal patterns in the data. Finally, Theil's U, a measure of predictive accuracy, has a value of 0.9401761 in the test set which implies that the model is providing more accurate predictions than a basic, naive approach. It indicates that the model is capturing patterns or relationships in the data that the simple benchmark is not accounting for.

12 Prediction intervals

Confidence intervals (CIs) are statistical tools used to estimate the range within which the true value of a parameter is likely to fall. The interpretation of these intervals relies on the chosen confidence level of 95%.

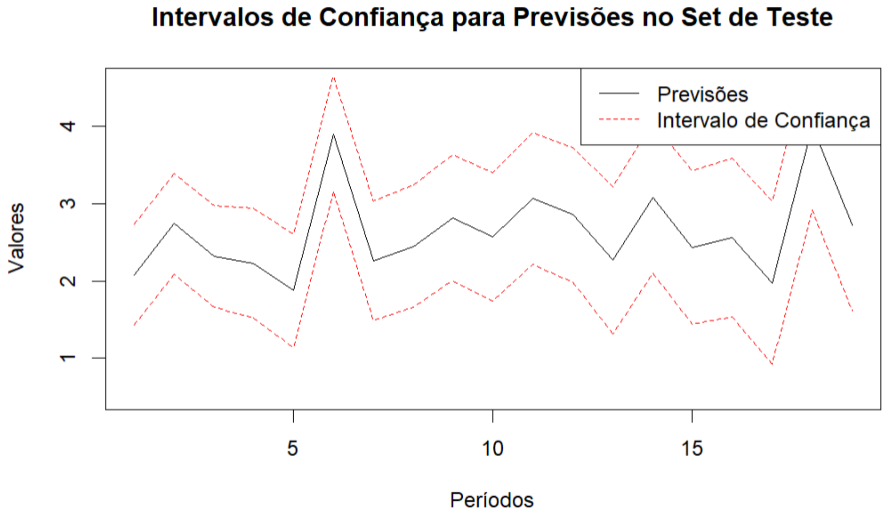


Fig. 24: Confidence Intervals (CIs) for the Test Set

They were calculated for the model parameters and are presented as upper and lower bounds. With an average amplitude and standard deviation of approximately 2.16 and 0.59, respectively, the confidence interval sizes can be classified as reasonably good. The results suggest a relatively low average interval width, indicating a tendency for more precise confidence intervals. Furthermore, the relatively low standard deviation indicates a small variability in ranges, suggesting reasonable consistency in the model estimates. These results provide evidence that the model is generating consistent and reliable confidence intervals for the parameters.

13 Discussion of the results

The choice of model and its parameters is very important for the forecasting of future observations. After evaluating the time series using the criterion of least information and separating the dataset into training and test, several different parameters were explored to obtain different ARIMA models. It was found that although autoarima function can give good results, this was not the case. ARIMA(2,1,2)(1,1,2)[12] showed the best results and was used for forecasting. It showed promising results despite some discrepancy between the observed and predicted values. Overall we obtained a consistent model with reliable confidence intervals.

13.1 Benefits of the model

These models excel in capturing underlying trends and seasonal patterns, providing a comprehensive understanding of temporal behaviours within the data. A notable feature of ARIMA models is their incorporation of automatic differencing, a valuable mechanism for addressing non-stationarity by applying differencing seamlessly to the time series. This automatic adjustment streamlines the modeling process, enhancing the model's ability to handle varied data structures. Furthermore, ARIMA models offer ease of interpretation with clear insights into model parameters - p (autoregressive lags), d (differencing), and q (moving average lags). This transparency enhances accessibility and facilitates a deeper comprehension of the model's configuration.

13.2 Limitations

However, ARIMA models may have limitations in efficiently capturing long-term dependencies. One crucial assumption is the assumption of linearity, relying on the belief in a linear relationship between past observations and forecasted values. It is important to note that this assumption may not always be aligned with the true nature of the time series, potentially limiting the model's accuracy. This models can be sensitive to extreme values, and the presence of outliers may introduce noise and affect too the accuracy of predictions. The stationarity requirement is a key consideration since this models works best with stationary data, and achieving stationarity may involve data transformations.

14 Conclusion

The analysis of crime occurrences in NYC revealed significant insights. Initially, we can observed an increase in criminal tendencies over time, with clear seasonal patterns and particularly higher crime rates in January and June. February stood out with a drastic drop, suggesting a strong seasonal pattern. Detrending and deseasonalizing methods effectively captured systematic components, as evidenced by ACF and PACF plots. The peaks in the data can imply unusual events that have a direct impact in crime rates and which are not included in this analysis. Although simple differentiations aided in understanding the series, ARIMA models may better account for temporal dependencies. For forecasting, ARIMA(2,1,2)(1,1,2)[12] yielded the best results, despite slight discrepancies between observed and predicted values. This model provided consistent forecasts with reliable confidence intervals, emphasizing the importance of selecting appropriate model parameters for accurate predictions.

It is important to consider and emphasize that the data under analysis pertains to a subject that is perhaps somewhat unpredictable or susceptible to random variations. Consequently, predicting crime rates becomes difficult since the factors associated with an increase in criminal activities can be challenging to forecast. This unpredictability is contingent on a possible variety of external factors, such as changes in economic conditions or shifts in social dynamics. The data inherent uncertainty in forecasting arises from its dependence on numerous interconnected factors, which are not explored or considered in-depth within the scope of this report.

References

1. Data Society: Nyc crime data (2023), https://data.world/data-society/nyc-crime-data