Modeling Frequency of Terrorist Attacks

Gary Cheng and Preetham Gujjula

1 Introduction

Global terrorism is a major issue today, and has been in the American zeitgeist at least since 9/11. Identifying patterns in attacks, and being able to predict future attacks, would be useful to law enforcement and the public. For these reasons, we decided to use this project as an opportunity to study a dataset of terrorist attacks from around the world.

The dataset we chose is the Global Terrorism Database, compiled by the National Consortium for the Study of Terrorism and Responses to Terrorism (START), located at the University of Maryland, College Park. START published the GTD on Kaggle, which is where we procured the dataset from [2]. The GTD aims to be a comprehensive record of all terrorist attacks, starting from 1970. It collects dozens of variables for each attack, including date, country, target, perpetrator, tactic used, weapons used, casualties, and fatalities [1].

We wanted to use the GTD to study the number of terrorist attacks that occur over time. In particular, we planned to derive a time series dataset from the GTD of the number of terrorist attacks that occur each month. We aimed to fit a model to this time series, and use the model to predict the number of terrorist attacks in the next 12 months. We decided on this duration because it was long enough to be valuable while still being within the prediction capabilities of the SARIMA model.

2 Data Analysis

2.1 Cleaning

Our uncleaned dataset contains every terrorist attack from 1970 to 2016. To form a time series out of this dataset, we counted the number of attacks that occurred each month and created a vector of these values. Figure 1 displays a plot of the number of attacks per month since 1970.

It should be noted that the data from 1998 to 2007 was collected retrospectively, as opposed to the other data, which was collected in real time. According to START, retrospective collection may underestimate the true number of terrorist attacks, because some media sources that documented the attacks are no longer available. Indeed, there is a subtle drop in the number of attacks in Figure 1 for those years.

Furthermore, the GTD broadly defines a terrorist attack as "the threatened or actual use of illegal force and violence by a non-state actor to attain a political, economic, religious, or social goal through fear, coercion, or intimidation". This is quite a loose definition, which is understandable,

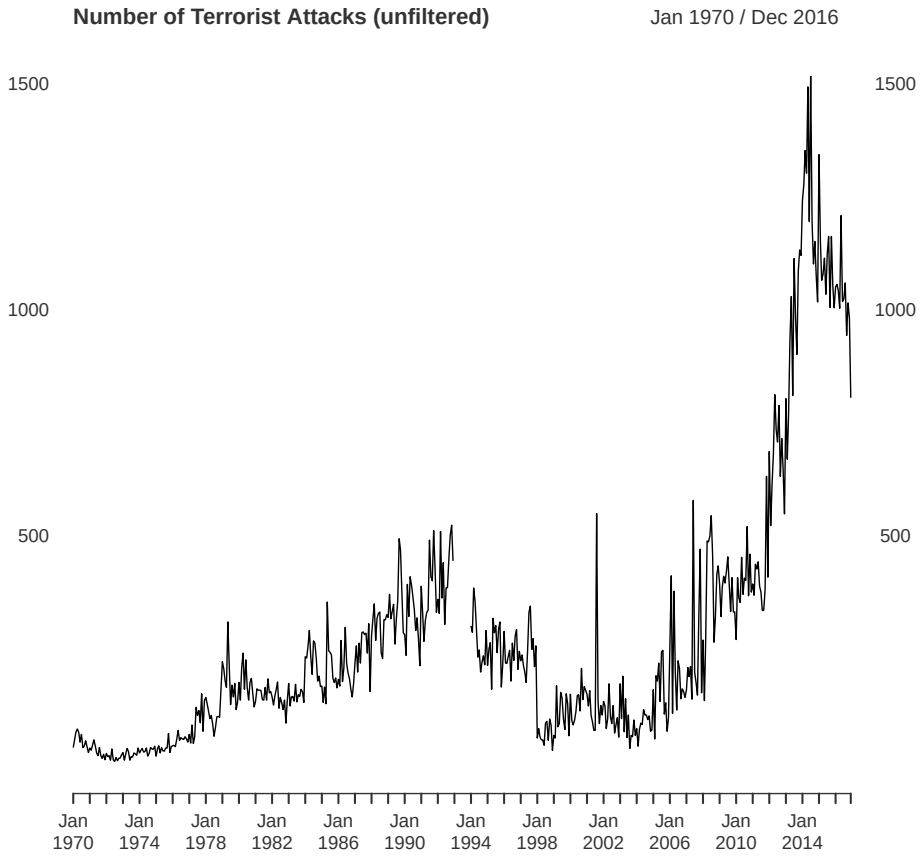

Fig. 1: The original time series of number of attacks per month from 1970 to 2016. Not filtered for Number of Casualities greater than or equal 10

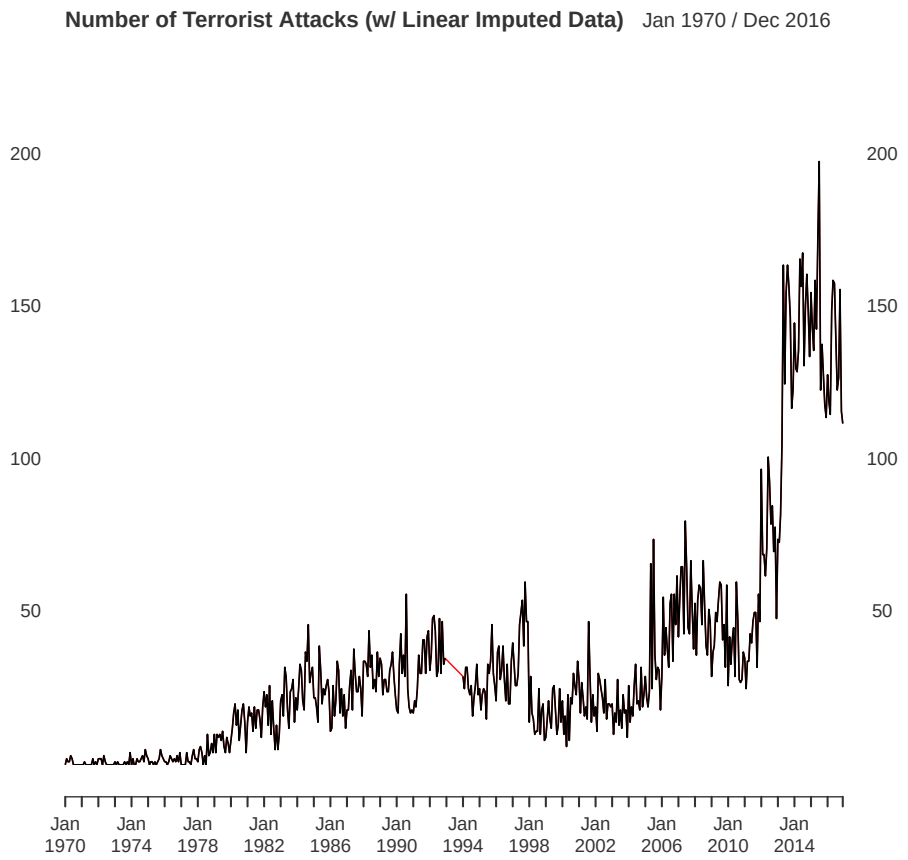

Fig. 2: The original time series with a linear imputation (in red) for the missing data in 1993.

since the GTD aims to be as inclusive as possible, and allow researchers to filter the data according to their own criteria.

For both of the reasons mentioned above, we filtered the original dataset to select attacks that resulted in 10 or more casualties. Our rationale is that the likelihood of under counting terrorist attacks of that magnitude is far less. Furthermore, attacks of this scale are more in line with what a layman would consider a terrorist attack. The plot of the filtered time series is rendered in figure 2 as the black lines. The definition of terrorist attack we will be working with is:

Definition 1: Terrorist Attack

The threatened or actual use of illegal force and violence by a non-state actor to attain a political, economic, religious, or social goal through fear, coercion, or intimidation, where the number of causalities is greater than or equal to 10.

There is a gap of 12 missing data points corresponding to the year 1993 because the data, which was collected by the Pinkerton Global Intelligence Service (PGIS), was lost before being transferred to START. To correct for this gap, we impute the data using the tsimpute library in R. In general, selecting an imputation method is difficult because it requires a model of the data first. Further-

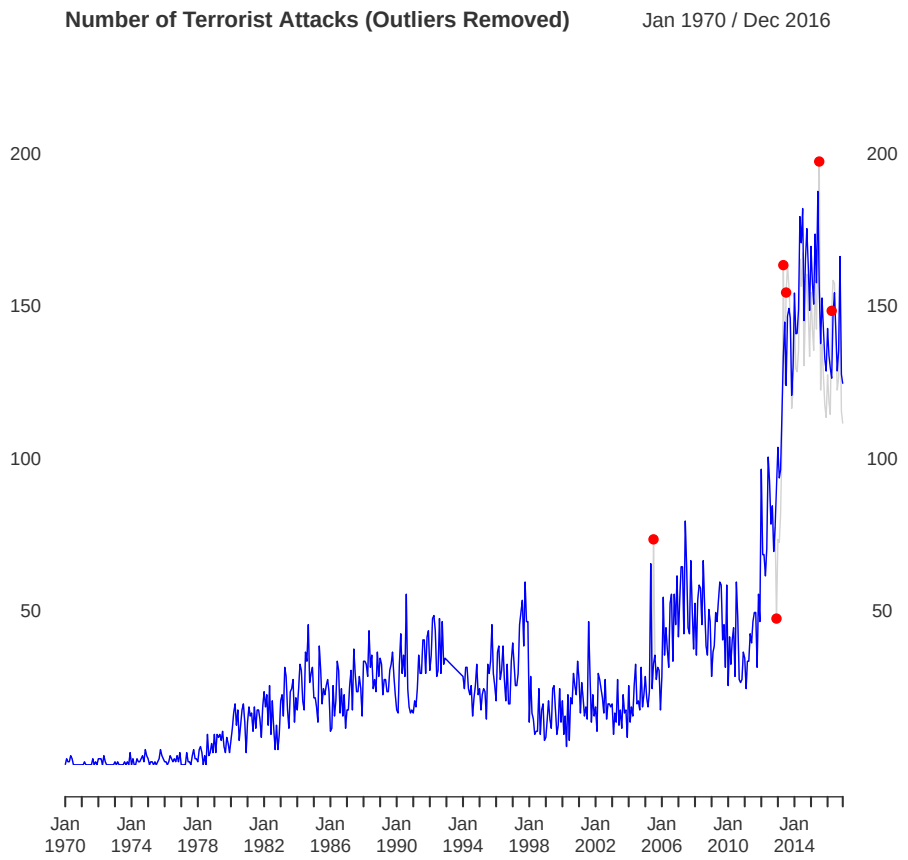

Fig. 3: The imputed time series (in blue) with outliers removed. The original time series shown in gray. The 6 outliers are shown in red.

more, our data loss was not at random locations but rather clustered together. Thus, we opted to use the simplest imputation method, which patches the missing data with a linear fit. Figure 2 displays the time series with the imputed data in red.

Now we filter outliers from the dataset. The library tsoutliers provides a convenient way to identify different kinds of outliers in the dataset. We decided to only filter for additive offset, transient change, and innovational outliers, i.e., outliers manifested as spikes in the underlying dataset. We decided not to filter for level-shift outliers, since we expect to see level-shifts in the timeseries. We also didnt filter for seasonal additive outliers, since we would not be surprised to see seasonal spikes or dips in the time series.

After filtering for 10 casualties, imputing the data, and removing outliers, the time series that we will be working can be found in Figure 3. The outlier effects detected can be viewed in figure 4

For purposes of model selection and cross-validation, we split our dataset into three disjoint components: training, validation, and testing. The training dataset was used to find candidate models. The validation dataset was then used to select the best candidate model. Then finally, the testing dataset was used as a final assessment of how well our selected model performs. It should be noted that these components are made disjoint to ensure that we do not bias the model to overfit on

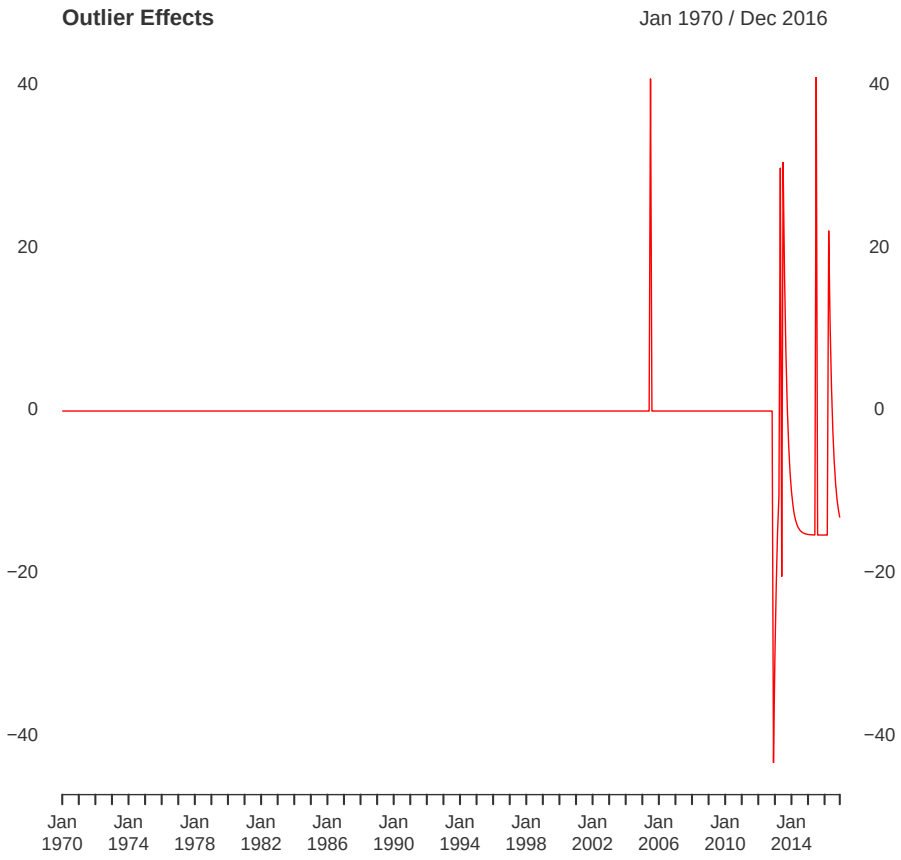

Fig. 4: The outlier effects for each of the identitifed outliers are displayed here.

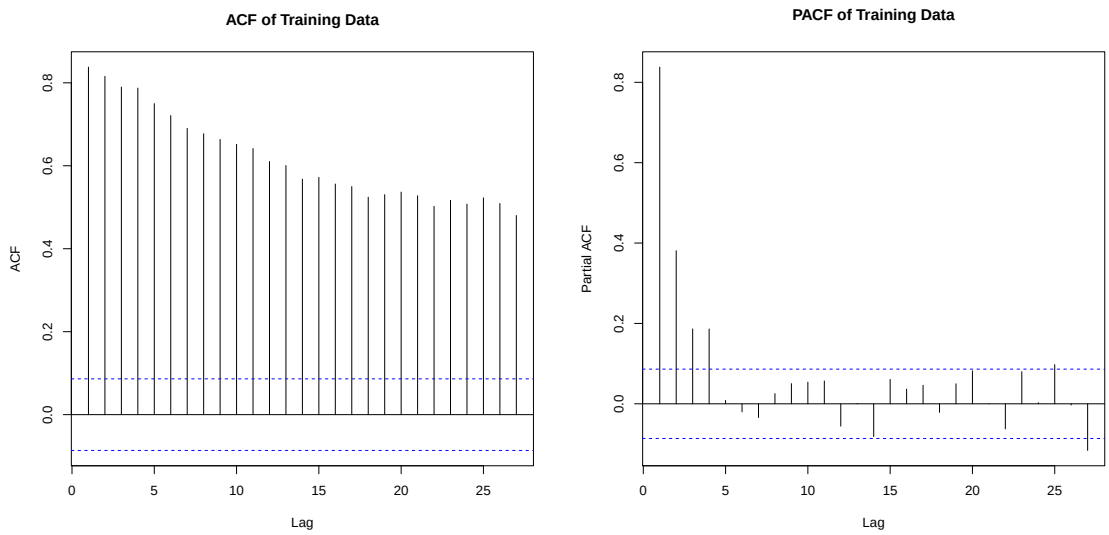

Fig. 5: The ACF and PACF of the time series

testing data. Furthermore, the components are sequential blocks with the training set consisting of data from 1970-2012 inclusive, the validation from 2013-2015 inclusive, and the testing being the last chunk which is the year 2016.

2.2 Chasing Stationarity

Visually, the original timeseries does not seem stationary. In particular, the mean of the series appears to increase as time progresses. In addition, the ACF and the PACF of the timeseries, plotted in figure 5, decays slowly as the lag increases—another indication that the series is not stationary.

One potential method to derive a stationary timeseries is to take the log of the data in the original series. This method would be highly suitable if the original series has an exponentially growing trend. The growth in our original series looks vaguely exponential, which justifies trying to log-transform the series. However, in our case, this method is problematic, since the original series has 0 values (recall that $\log 0$ is undefined). We could replace the 0 values with some small positive ϵ value, as a potential fix. Unfortunately, $\lim_{x\to 0+} \log x = -\infty$, so taking $\log \epsilon$ would result in an extremely negative value. To our knowledge, there is no straightforward way to choose an ϵ, and furthermore $\log \epsilon$ would also be very sensitive to our choice. For this reason, we opted not to log-transform the data.

In lieu of using a log-transform, we tried differencing and twice-differencing the dataset as a method of obtaining a stationary dataset. The ACF and PACF of these the diffed datasets are plotted in figure 6 and figure 7. The sharp drop-off in the ACFs of these datasets suggests that they are both stationary.

For a more quantitative measure of stationarity, we used the augmented Dickey-Fuller test on these two time series. Recall that in this test the null hypothesis is that our time series is non-stationary and the alternate hypothesis that our time series is stationary. Under this test, the original time

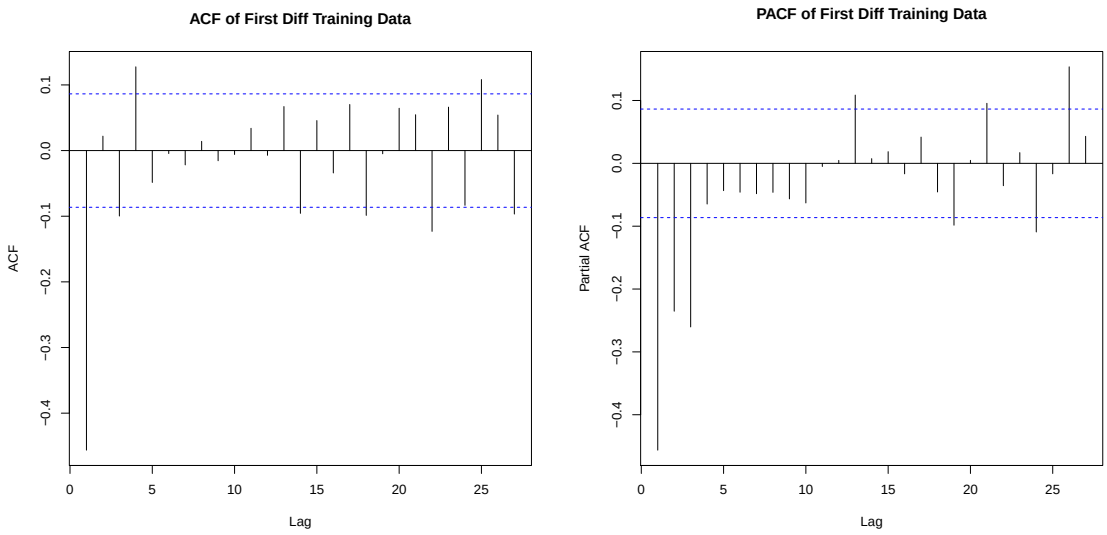

Fig. 6: The ACF and PACF of the first diff time series

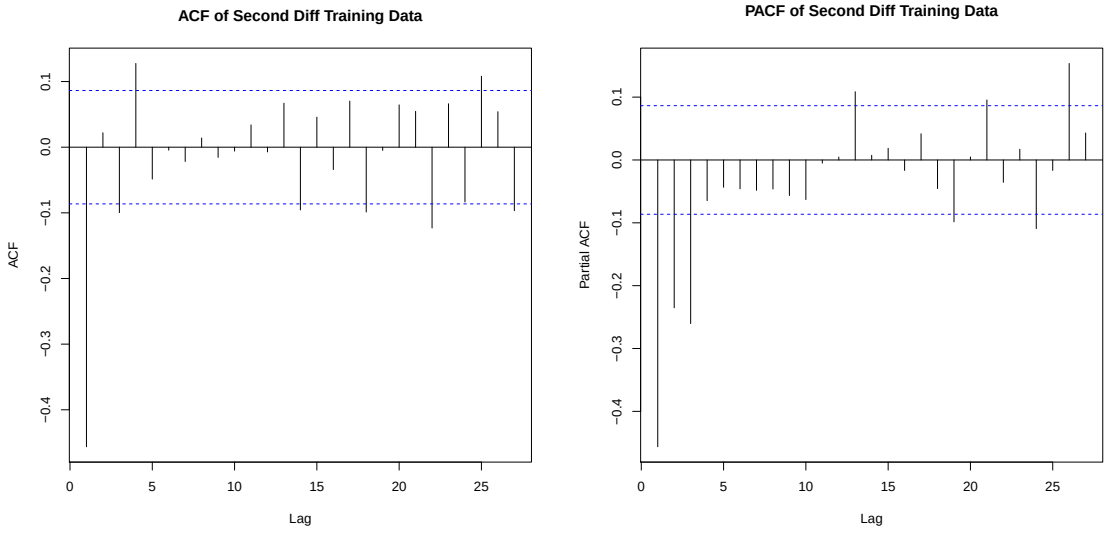

Fig. 7: The ACF and PACF of the second diff time series

AR/MA	0	1	2	3	4	5	6	7
0	x	o	x	x	o	o	o	o
1	x	x	o	x	o	o	o	o
2	x	x	x	x	o	o	o	o
3	x	x	x	x	o	o	o	o
4	x	o	o	o	o	o	o	o
5	x	x	o	o	o	o	o	o
6	x	x	o	o	o	o	o	o
7	x	x	o	o	o	o	o	o

AR/MA	0	1	2	3	4	5	6	7
0	x	x	x	x	o	o	o	o
1	x	x	x	x	o	o	o	o
2	x	x	x	x	o	o	o	o
3	x	o	o	o	o	o	o	o
4	x	x	o	o	o	o	o	o
5	x	x	o	o	o	o	x	o
6	x	x	x	o	o	o	o	o
7	x	x	o	o	o	o	o	o

Tab. 1: The EACF table for once diffed (left) and twice diffed data (right). The "o" points represent values that are candidate models.

series has a p-value of 0.4415 for lag order 8 (this value was selected by the test). The differenced series had a p-value < 0.01 for lag order 8 and the twice-differenced series also had a p-value < 0.01 for lag order 8. This means that at the $\alpha = 1\%$ significance level, we reject the null-hypotheses for the once differenced and twice differenced time series, and accept the alternative hypotheses that these series are stationary. However, we are not able to reject the null for our original dataset for significance level $\alpha = 0.05$.

2.3 Model Selection

To obtain a set of potential ARIMA models that fit our time series, we use the extended sample autocorrelation function (EACF) method. In this method, we generate a table of sample ACFs for ARMA(p, q) models, where p is in between 0 and 7 and q is in between 0 and 13. We use this table to select promising models for further analysis. The EACF tables (generated by the eacf function in R) for the differenced and twice-differenced time series are shown in table 1. Entries with "o" in them correspond to models that are promising. In both tables, large values of p and q are valid model parameters, but we do not want to select too large of a value of p and q because this could mean that we are overfitting the data. For the twice differenced data, the EACF does not have any candidate models where $q < 4$ and $p < 3$. This could be an indication that differencing twice may not be a great choice because we would need large model parameters to fit our data, which may be an indication of overfitting.

None of the models we considered failed the null hypothesis of Ljung-Box test with a significant p-value. In other words, the Ljung-Box test did not identify a statistically significant correlation among the sample residuals, for any of the models that we considered. This is promising, because we expect the sample residuals of a well-fitting model to be independent.

With potential candidate non-seasonal parameters chosen, we now turn to identifying what seasonal lags, we should consider. In the ACF and the PACF plots for both The periodogram for the differenced data in figure 8. For a more useful visual, we also plot a smoothed and tapered periodogram. After experimenting around with kernel and taper choices, we decided to use a Modified Daniell Kernel with $m = 22 \approx \sqrt{n}$ and a split cosine bell with a taper of 0.1. Out of all the periodogram variants that we tried, these values gave us a reasonable looking periodogram that we could find a seasonal lag from.

Upon visually inspecting our smoothed and tapered periodogram, we see a peak around $\omega = 0.27$,

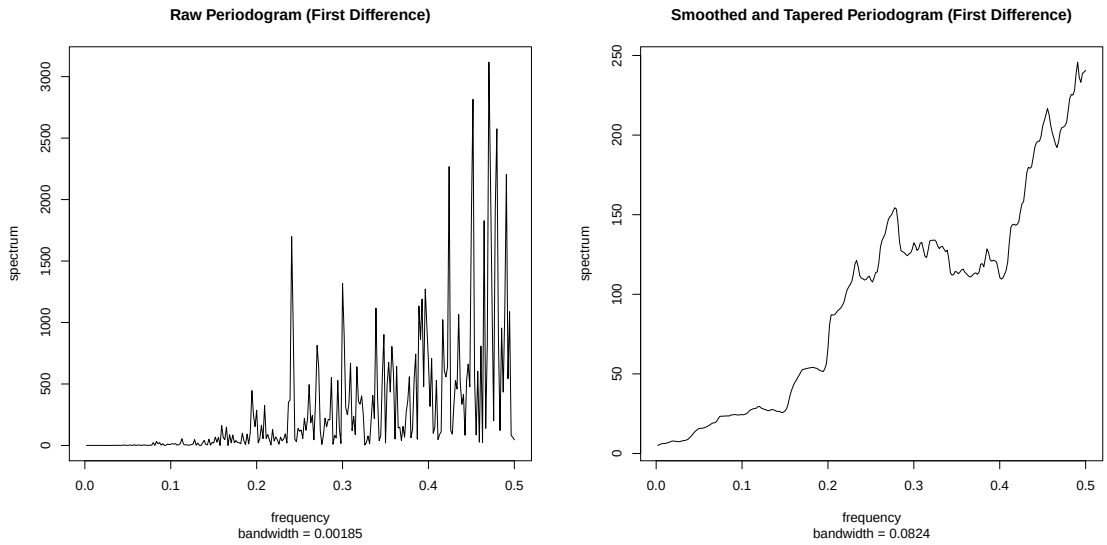

Fig. 8: The Periodogram (left) and a Smoothed and Tapered periodogram (right). Smoothed with a Modified Daniell Kernel with parameter $m = \sqrt{n} \approx 22$ and tapered with a split cosine bell taper with parameter 0.1

suggesting a seasonality in the data of $\frac{1}{0.27} \approx 4$. We later explore some models with this seasonal lag value.

To evaluate how well our models fit the data, we compute the AIC, BIC, and AICc for each of our 8 candidate models, in the table below. We selected these 8 candidate models by playing around with the lower order options returned by the EACF. For each of these non-seasonal candidates we paired it with seasonal models with lag 4, where the seasonal P, Q, and D were not too large. We chose small values of p, d, q, P, D, and Q because we noticed that many of the coefficients in higher order models were insignificant and had AIC, AICc, and BIC that were much larger than the corresponding smaller models. This leads us to conclude that larger order models may be overfitting to our training data set.

To aid model selection, we also calculate mean-squared error for each of our candidate models. We calculate MSE in the following way:

Algorithm 1: MSE Calculation

1. for year˙of˙data in validation:
   (a) prediction = forecast(1 year ahead, training˙data)
   (b) squared˙error += $\|\text{prediction} - \text{year˙of˙data}\|_2^2$
   (c) training˙data = concatenate(training˙data, year˙of˙data)
2. MSE = squared˙error / 3

Recall that our validation set consists of 3 years of data. So to calculate our MSE we predict one year ahead and take the sum of square difference between our prediction and what actually

$(p,d,q) \times (P,D,Q)_s$	AIC	AICc	BIC	MSE
$(0,1,1)$	5.248	5.252	4.265	13212.18
$(0,1,1) \times (1,0,1)_4$	5.237	5.241	4.270	13451.05
$(0,1,1) \times (1,1,1)_4$	5.249	5.254	4.274	13325.71
$(0,1,1) \times (1,1,2)_4$	5.242	5.246	4.275	13728.1
$(1,1,2)$	5.256	5.260	4.288	13237.34
$(1,1,2) \times (1,0,1)_4$	5.243	5.247	4.292	13785.23
$(1,1,2) \times (1,1,1)_4$	5.252	5.257	4.294	14144.5
$(1,1,2) \times (1,1,2)_4$	5.250	5.254	4.299	13826.03
$(3,2,1)$	5.259	5.263	4.291	13216.42
$(3,2,1) \times (1,0,1)_4$	5.254	5.258	4.303	13664.76
$(0,1,1) \times (1,0,1)_3$	5.246	5.250	4.279	14670.6
$(3,2,1) \times (1,0,1)_3$	5.254	5.259	4.303	13456.07

Tab. 2: The metrics associated with different SARIMA models denoted by the left hand column.

happened. We do this three total times for each of the 3 years in the validation and take the average. The averaging is useful to ensure that our model is generalizable to different forecasts and not to just a single one. We also could have performed a sum of square calculation 25 times instead of just 3 by performing a sum of square calculation on 12 month ahead forecasts starting from every month in the validation set that has 12 datapoints following it. We chose not to do this because we felt that the sum of square would be be too correlated with one another. A model that may only fit poorly for a portion of the overall validation set may get penalized repeatedly for it in this scheme.

The AIC, AICc, BIC, and MSE for each of our candidate models is shown in table 2.

Note that BIC tends to work better when the number of observations is large relative to the model order, and AICc does better when it is relatively smaller in comparison to the model order [3]. In our project, the number of observations is quite large relative to the model order; for this reason, we decided to focus primarily on MSE and BIC values in determining our best model.

Based on our validation methods, we chose as an ARIMA(0, 1, 1) as our final model. Even though its AIC and AICc values are middle of the pack, it had the best BIC and MSE values. The full properties of this model is shown in figure 9.

2.4 Forecasting

Using our final model, we generated a forecast of ¨ months, displayed in figure ——. The prediction fits the behavior of the data fairly well. What actually occurred, shown in gray fits between the 95% confidence interval of the forecast.

3 Conclusion

Through our analysis, we demonstrated that major terrorist attacks over a short period of time follow a predictable pattern. In particular, the ARIMA model we selected produced an reasonably accurate forecast of the monthly terrorist attacks in 2016.

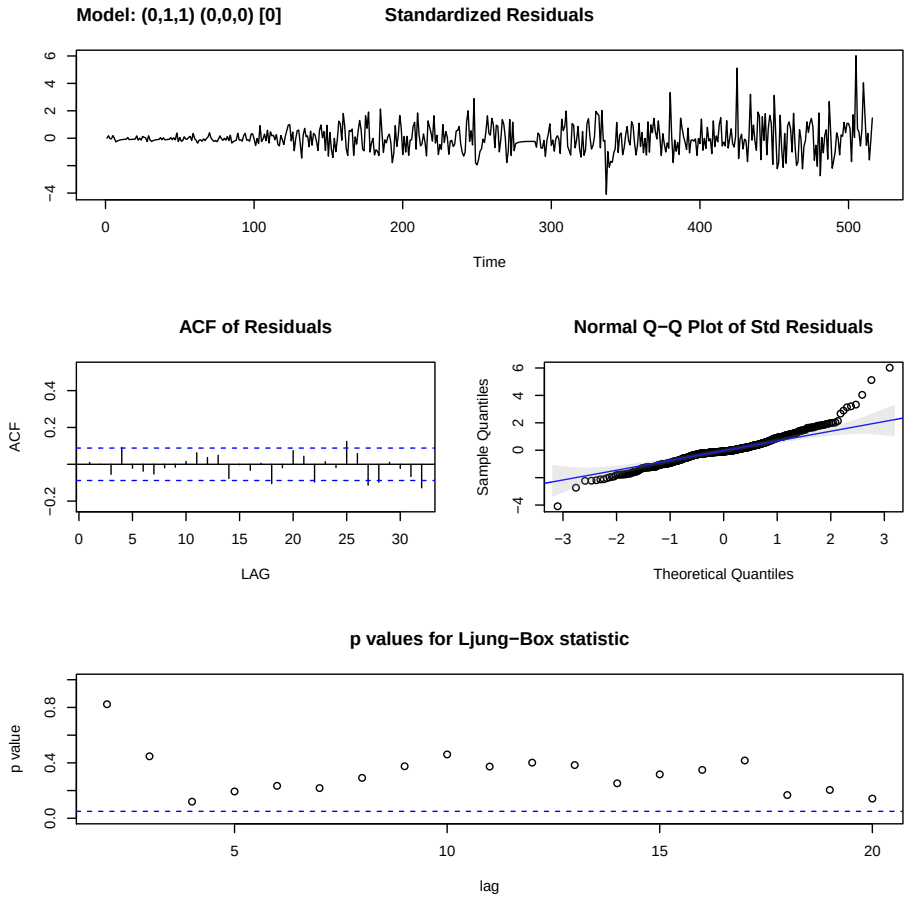

Fig. 9: This is the information from our best model.

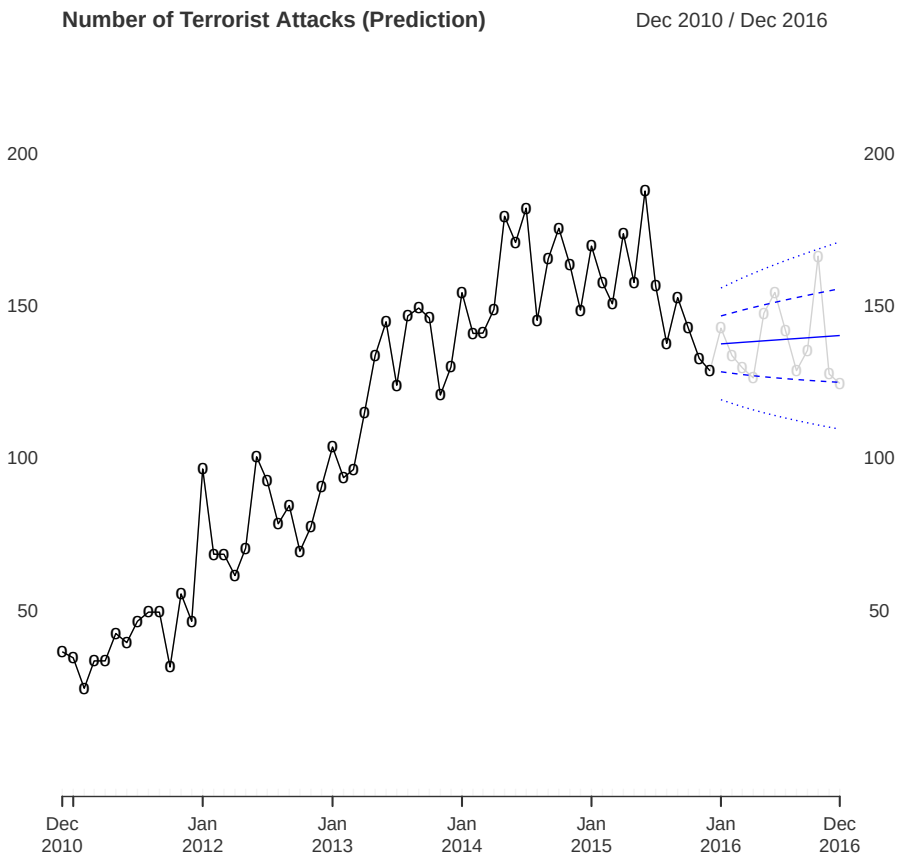

Fig. 10: We use the model ———- to generate a 12 month prediction denoted by the blue line. 1 sd and 2 sd error bars are the dotted lines. What actually occured is shown by the light gray.

Notably, our non-seasonal models fit the time series better than our seasonal models, which was an unexpected result. In fact, our final model was ARIMA(0, 1, 1), which is a fairly simple model. We suspect that considering data from all areas of the globe may have obscured seasonal and more complex, potentially non-linear patterns. In the future, filtering the data for a specific country or region may produce a more seasonal dataset. For example, incidents of terrorism in a given country may spike around election time, or during a specific season of the year.

Our next objective in analyzing the GTD would be to study its correlation to other datasets. Generally, terrorism attacks are motivated by an external cause, such as a collapse of a government or an ethnic conflict within a state. Therefore, studying the correlation between terrorist attacks over time and another time series would probably yield fruitful results. For example, we would expect that the GDP of a country and the number of terrorist attacks that occur in a country to be correlated.

This project was a good foray into the patterns in the GTD, but future research into filtered subsets of the GTD, as well as into exploring the relationship between the GTD and other datasets, is likely to produce novel results.

References

[1] *Data Collection Methodology*. URL: http://www.start.umd.edu/gtd/using-gtd/. (accessed: 04.16.2018).

[2] *Global Terrorism Database*. URL: https://www.kaggle.com/START-UMD/gtd. (accessed: 04.16.2018).

[3] David S. Stoffer Robert H. Shumway. *Time Series Analysis and Its Applications*. 2017.