University of Twente
2018 - 2019

The Forecasting-Ability of Google Search Frequencies for Predicting Crime in the Netherlands.

Müller, S.-A.

Psychology (M.Sc.): Conflict, Risk & Safety

BMS: Behavioural, Management and Social Sciences

University of Twente, Enschede (NL)

Master's Thesis

1st supervisor: Dr. ir. de Vries, P.

2nd supervisor: Dr. Stel, M.

Abstract

Being able to predict the development of crime includes several advantages. In the past, some interesting approaches have been already made to evaluate other factors that are useful for predicting the development of crime besides crime rates themselves. The aim of this paper is it to explore one of these factors by answering the question whether Google search frequencies (GSFs) of terms that display the intention to commit a crime would be able to enhance the accuracy of a quantitative model in forecasting future crime development. To explore these questions, data collected from the platform Google Trends or the data portal of the Central Bureau for Statistics (CBS) are used to create three different models: 1) a linear regression model, 2) an AR(3) model, and 3) an ARIMA(3,1,3) model. The results show that GSFs, while not necessarily lessening the forecasting ability of a model, do not increase the accuracy of the model compared to a model without GSFs. Although the results in this study are non-significant, further investigations are highly recommended.

Keywords: Crime, Time-series analysis, Google search frequencies (GSFs), Forecasting, Autoregressive, ARIMA

Introduction

In the field of social science, researchers hope that getting more and more access to a great amount of complex data, so-called big data, will offer new ways of gaining knowledge about socially relevant behaviour (Connelly, Playford, Gayle, & Dibben, 2016). In addition, it may also offer new ways to predict said behaviour. For instance, Wang, Kifer, Graif, and Li (2016) as well as Williams, Burnap, and Sloan (2017) state that the use of big data enables new approaches in research when it comes to recognizing crime and predicting its development.

Being able to predict the development of crime would certainly include several advantages. One of them is that it enables the police to foresee the development of crime and detect locations where crimes are most likely to occur (McClendon & Meghanathan, 2015). This, in turn, may allow them to take more preventive actions instead of simply reacting to a crime and to be better prepared for an increase in crime (e.g., early communication with civilians and other institutions, sensible distribution of field staff, etc.). Unsurprisingly, attempts to utilize big data for crime research have already been made. For instance, McClendon and Meghanathan (2015) were able to create a linear regression model on crime data that can be used to forecast future crime development. However, there is evidence that also other factors, like chances of employment (Freedman, Owens, & Bohn, 2018) or even geographical information (Wang, Kifer, Graif, & Li, 2016), can contribute valuable information to the prediction of crime development. More importantly, there are indications that including other factors in an analysis, instead of solely using crime rates, will even enhance the prediction accuracy of a model (Wang et al., 2016).

Although some interesting research approaches, like the ones mentioned above, do exist that use one type of big data for crime research, the option to combine crime rates with big 'social' data seems to be relatively unexplored (Williams, Burnap, & Sloan, 2017). Therefore,

the goal of this study was it to search for a type of big 'social' data that could possibly contribute to the prediction of crime development.

Theoretical framework

Big 'social' data and crime. While studies wherein big 'social' data are used for predicting crime development are relatively limited (Williams et al., 2017), there still can be found some interesting investigations in scientific literature. One example is a study conducted by Williams et al. (2017), where information about disorder and individual offenses, which were posted on the social media platform Twitter by civilians, were used to evaluate crime behaviour patterns and to forecast the development of crime in London. Williams et al. (2017) suggested that the use of social media data would bring with it the advantage of using 'naturally occurring' social data to observe the development of crime that are also faster available than later provided crime rate statistics.

As a limitation, however, Williams et al. (2017) point out that social media data would not be able to reflect the entire population, since some people are less inclined to use or even possess a social media account than other ones (e.g., senior citizens). In addition, Williams et al., 2017 noted that significant results were only found in low-crime areas. They reasoned that individual offenses in high-crime areas would probably not attract enough attention and therefore less likely to be tweeted.

Altogether, the observation of social media data has been evaluated as a promising method to predict crime (Williams et al., 2017). However, certain possible limitations, as discussed above, may be a good reason to search for alternative form of big 'social' data. Recently, considerable attention has been paid to the use of Google search frequencies (GSFs), which are provided by the platform Google Trends, as a predictor for future outcomes. As an example, the patterns of people's information-seeking behaviour on the Internet search engine Google have been already successfully used to predict the development of influenza epidemics

(Ginsberg et al., 2009) or the development of housing prices and sales rates (Wu & Brynjolfsson, 2015). As the use of GSFs would not only offer information from social-media user but from all people using Google, this method might be a promising alternative to Williams et al. (2017) approach. Therefore, the question of this research was whether including GSFs in a quantitative model would be able to enhance its accuracy in forecasting future crime development.

Google search frequencies. While studies regarding crime investigation wherein big 'social' data are used are relatively limited (Williams et al., 2017), there still can be found some successful approaches. One example would be Williams et al. (2017) use of Twitter data to predict the development of crime in London. However, not all people actively use or even possess a Twitter account (Williams et al., 2017); a limitation that could generally apply to any research with social-media data. For illustration, the evaluated number of social-network users in the United Kingdom (UK) in 2017 was around 39.4 million people (Statista, 2018) while the total population of the UK was around 65.7 million people (ONS, 2017). Thus, only 60% of the population in the UK were actually active on social media platforms, while around 90% of all households in the UK had access to the internet in 2017 (ONS, 2018). As it was already mentioned, one promising alternative would be the use of GSFs since these data would not only offer information from social-media user but from all people using Internet search engine Google.

GSFs are numerical information representing peoples' information-seeking behaviour via Google, which are provided by the platform Google Trends. They can range from 0 to 100 and stand for the proportion or the volume of the rates of searches regarding that topic for a certain localisation (e.g., country, region, city) and for a certain time (e.g., year, month, week) (Choi & Varian, 2012; Wu & Brynjolfsson, 2015). GSFs are no fixed numbers, since they can vary for a certain point in time and location depending on the chosen time interval and the day the data are downloaded (Choi & Varian, 2012). While researchers are quite reluctant in making

a predication about GSFs as predictor for events in remote future, its predictability of events in the near future has been mostly accredited (Choi & Varian, 2012; Wu & Brynjolfsson, 2015).

Information-seeking behaviour. Observing the information-seeking behaviour of people through GSFs is not a completely new concept and has already indicated new ways to recognize trends and patterns in data (e.g., Ginsberg et al., 2009; Wu & Brynjolfsson, 2015). Still, there seem to be no indications that its elaboration has been used for something even closely related to crime-prediction research. Most of the theories regarding information-seeking behaviour have in common that the act of seeking information is triggered by the need of information (Savolainen, 2018). The more complex the task, the more complex the information needed (Byström & Järvelin, 1995). To give an example, if people are under the perception that they are in danger (e.g., through an increasing development of crime in their environment), they are very likely to engage in self-protective behaviour (Kievik & Gutteling, 2011). In other words, a high level of perceived risk increases the likeliness that people are motivated enough to show pro-active behaviour, such as seeking information about potential ways to protect themselves (Kievik & Gutteling, 2011). Therefore, a high rate of GSFs that are related to crime might indicate that people feel threatened by an increasing development of crime in their environment. Naturally, the need to seek information could not only be a reaction to crime, but also reflect the intention to commit a crime. Therefore, it was not only considered to use GSFs of search term that may display the reaction to a crime but also ones that may display the intention to commit a crime.

Total and high impact crime rates. To evaluate the development of crime, the focus was laid on both total crime rates and so-called High impact crime (HIC) rates. HIC is a specific term used by the Dutch police to define acts of crime that are supposed to have a high negative influence on the victim of said crime and the general social environment (e.g., neighbourhood, community, media, etc.) (CCV, n.d.; Regioburgemeesters, n.d.). In doing so, HIC distinguishes between four types of crime: 1) burglary (‘diefstal/ inbraak woning’), 2) violence

('geweldsmisdrijven'), 3) assault ('overvallen') and 4) theft ('straatroof'). It was chosen to include HIC rates as sub-categories in the analysis with regard to the possibility that GSFs might only increase the forecasting accuracy of a quantitative model for a specific type of crime.

Current study

In summary, this scientific paper investigated whether the use of GSFs would be able to enhance the accuracy of a quantitative model in forecasting the development of crime. In order to do so, the forecasting-ability of a quantitative model for general and specific types of crime including GSFs of terms that either display the intention to commit a crime or the reaction to it was compared with the forecasting-ability of a quantitative model without including GSFs.

Method

Materials and Data-sets

In order to evaluate a quantitative model to forecast the development of crime, several data-sets were downloaded from either the platform Google Trends or the data portal of the Central Bureau for Statistics ("Centraal Bureau voor Statistiek", CBS). Beforehand, it was established the variables included in the model should be ideally represented by I) GSFs of terms that are positively related to crime and security, II) Crime rates, and III) the Population rates for IV) each Region. Furthermore, all sets would have to meet the condition that they were including data from January 2012 to December 2017 on a monthly level for the Netherland's twelve provinces (Groningen, Friesland, Drenthe, Overijssel, Flevoland, Gelderland, Utrecht, Noord-Holland, Zuid-Holland, Zeeland, Noord-Brabant and Limburg).

To receive the GSFs, in total 90 search terms were chosen from which we anticipated that one half would display the intention to commit a crime and one half would display the reaction to a crime (see Table A1 in Appendix). Within November the 10th and November the

29th 2018, for each of the 90 search terms was a file downloaded from Google Trends with the proportion of the rates of searches regarding that term, ranged from 0 to 100, for the twelve Dutch provinces from January 2012 to December 2017. It was afterwards chosen to only include the 45 search terms that were expected to display the intention to commit a crime into further processes. The decision was based on the assumption that the intention to commit a crime would be observable before the execution of said crime, while the reaction to that crime would be observable at a later point. Therefore, it was expected that intentional search terms would probably hold a stronger capability to forecast the development of crime than reactional search terms.

The Crime rates were downloaded from the data portal of the CBS. As it is already mentioned above, we originally planned to include the total crime rates and the rates of the HIC sub-categories to investigate whether the type of a crime might influence the accuracy of our model. With regard to the data-sets provided by the CBS, we received five sets of data from January 2012 to December 2017 on a monthly level for all the provinces in the Netherlands with following rates chosen: I) the total crime rates, II) the rates for burglary (‘diefstal’), III) open violence (‘open geweld’), IV) assault (‘overvallen’) and V) theft (‘straatroof’). In addition, a data-set with the Population rates from January 2012 to December 2017 on a monthly level for all the provinces in the Netherlands was downloaded.

Data preparation and exploration

The data preparation and exploration process, as well as the model creation and analysis, were carried out with the open-source environment R (R Core Team, 2018). The data were split in a training and a test set. The training set was created with 75% of the total data (January 2012 until June 2016), while the remaining 25% of the data were put into the test set (July 2016 until December 2017).

The explorative analysis showed that the variable Crime rates for open violence ('open geweld'), assault ('overvallen') and theft ('straatroof') contained respectively 0.15%, 6.79%, and 2.62% missing values in the training set. It was determined that variables which contained 5% or more missing values would be excluded from further analysis processes. Consequently, this lead to the elimination of the variable Crime rates for assault. The other two variables remained.

After that, we chose to reduce the number of GSFs that would be used to create the models by examining the correlations between the GSFs and the Crime rates in the training set. Since we assumed that the intentional search terms have a positive correlation with the development of crime, only GSFs with a correlation threshold that was around zero or higher were included in the model creation. Consequently, a unique set of GSFs was used for each Crime rate and each Region.

Model creation

The question of this study was whether the use of GSFs would be able to enhance the accuracy of a quantitative model in forecasting the development of crime. Evaluating this question was approached by creating models that either did or did not contained GSFs (Alternative model vs. Main model).

The Main model was created as a linear regression model with the predictors Crime rates, the Population rates and the Region, without including GSFs. The dependent variable contained the Crime rates shifted by one point in the future. The Alternative model was created as an autoregressive (AR) model with three time lags. Using three time lags means that not only the last value of a predictor (in this case the predictor Crime rates) is included in a regression model to forecast the next one, but also the second and the third one beforehand (Shumway & Stoffer, 2017). Hereinafter, this model will be therefore referred to as AR(3) model. Other predictors included in the AR(3) model were the GSFs, the Population rates and the Region. In

order to reduce the risk that non-stationarity – change of data and their effectiveness due to their time structure (Shumway & Stoffer, 2017) – may disturb the results, a third model, namely an autoregressive integrative moving average (ARIMA) model, was created. Hereinafter, this model will be referred to as ARIMA(3,1,3) model since the parameters which were used to create the model were $p = 3$, $d = 1$, and $q = 3$. The parameter p represents the number of time lags, as it was already mentioned while describing the AR(3) model. The parameter d represents the so-called degree of differencing (Shumway & Stoffer, 2017). The parameter q represents the order of the moving-average model (Shumway & Stoffer, 2017). The ARIMA(3,1,3) model included the same predictors as the AR(3) model and the same dependent variable.

The three models were compared in their forecasting ability – namely the model fit – with the aim to ultimately chose the best working statistical model. The model fit was evaluated as followed.

Firstly, the Main model and the AR(3) model were compared with each other. Both models were created with data from the training set and for each of them, the average R squared was evaluated. Additionally, the auto-correlation function (ACF) and the partial auto-correlation function (PACF) plots were generated to evaluate whether the values in the plot would improve by decreasing to zero. After that, each model was used to predict future Crime rates for the next three months with data from the test set. In other words, the models containing data from the training set (January 2012 until June 2016) were used to predict the Crime rates for July, August and September 2016. After that, the data from the training set and from July 2016 were used to create a new model that predicted August, September and October 2016. These steps were repeated until December 2017 was reached. These predicted Crime rates for the first, the second and the third month in the future were compared with the actual Crime rates by estimating the three Root Mean Squared Errors (RMSEs) for all four Crime rates.

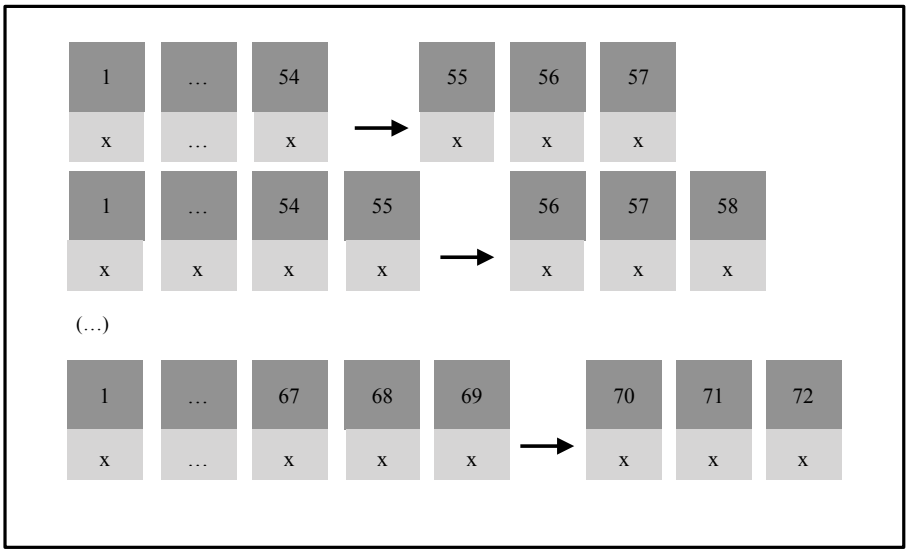

Figure 1: *Visualization of prediction process*

Secondly, the AR(3) model and the ARIMA(3,1,3) model were compared with each other. Therefore, the ARIMA(3,1,3) model was created with data from the training set and for each of them, the ACF and the PACF plots were also generated. After that, the ARIMA(3,1,3) model was used to predict future Crime rates in the exact same way as the AR(3) and the Main model. These predicted Crime rates for the first, the second and the third month in the future were compared with the actual Crime rates by estimating the three Root Mean Squared Errors (RMSEs) for all four Crime rates.

Results

Main model vs AR model

The results show that the average R squared value of the AR(3) model is higher (M = 0.59, Mdn = 0.57) than the average R squared value of the Main model (M = 0.48, Mdn = 0.49), indicating that the effectiveness of the AR(3) model would be most likely higher. Comparing the ACF and PACF plots indicated a slight improvement of the data structure for the AR(3) model.

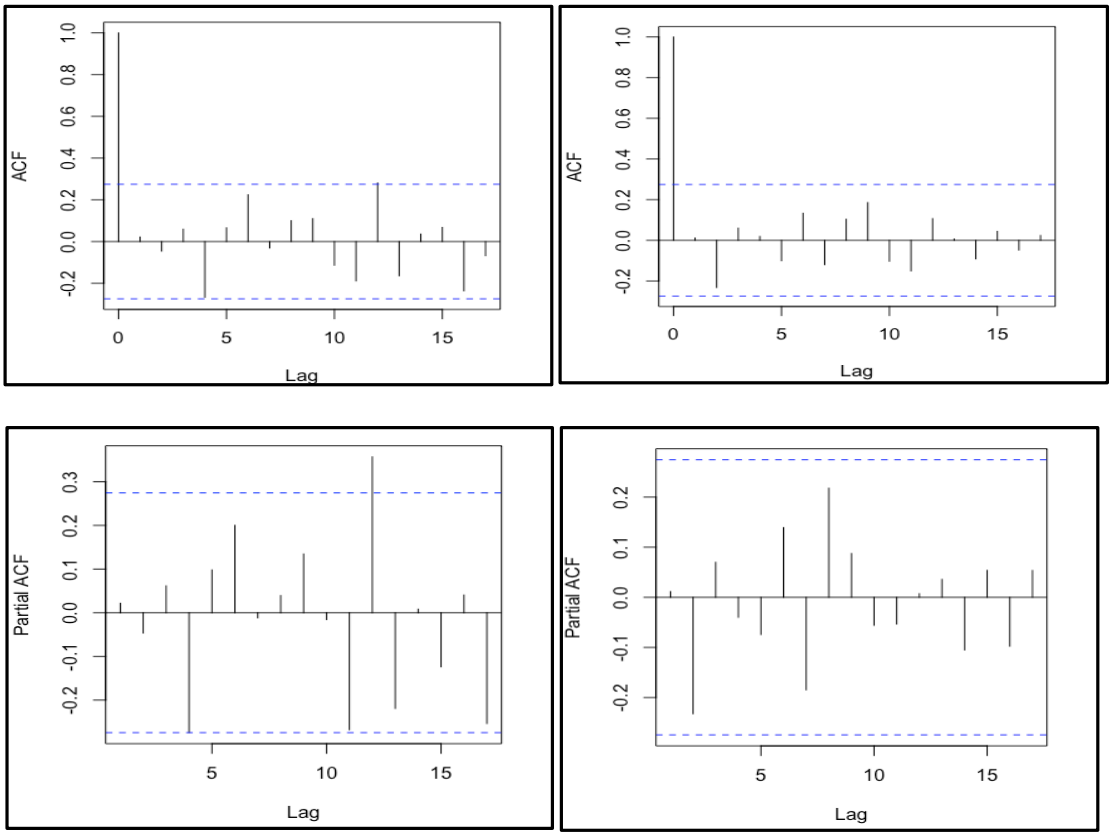

Figure 2: *ACF and PACF examples for the Main model (left) and AR(3) model (right)*

Calculating the mean RMSEs for the Crime rates resulted in the Main model having smaller measurements than the AR(3) model for month 1 (M = 227.97, SD = 149.64 vs M = 282.24, SD = 180.30), month 2 (M = 280.31, SD = 153.87 vs M = 300.09, SD = 159.71) and month 3 (M = 357.84, SD = 253.59 vs M = 410.03, SD = 267.99) (for the other Crime rates see Table A2 and Table A3 in Appendix). This contradict the indications of the average R squared value of the two models.

Doing a posterior test by measuring the average adjusted R squared value showed that the AR(3) model seemed to have no higher effect (M = 0.45, Mdn = 0.44) than the Main model (M = 0.44, Mdn = 0.45). This indicated that the higher average R squared value of the AR(3) model is most likely caused by the higher amount of predictors.

AR model vs ARIMA model

Comparing the ACF and PACF plots showed no general improvement of the data structure for the ARIMA(3,1,3) model.

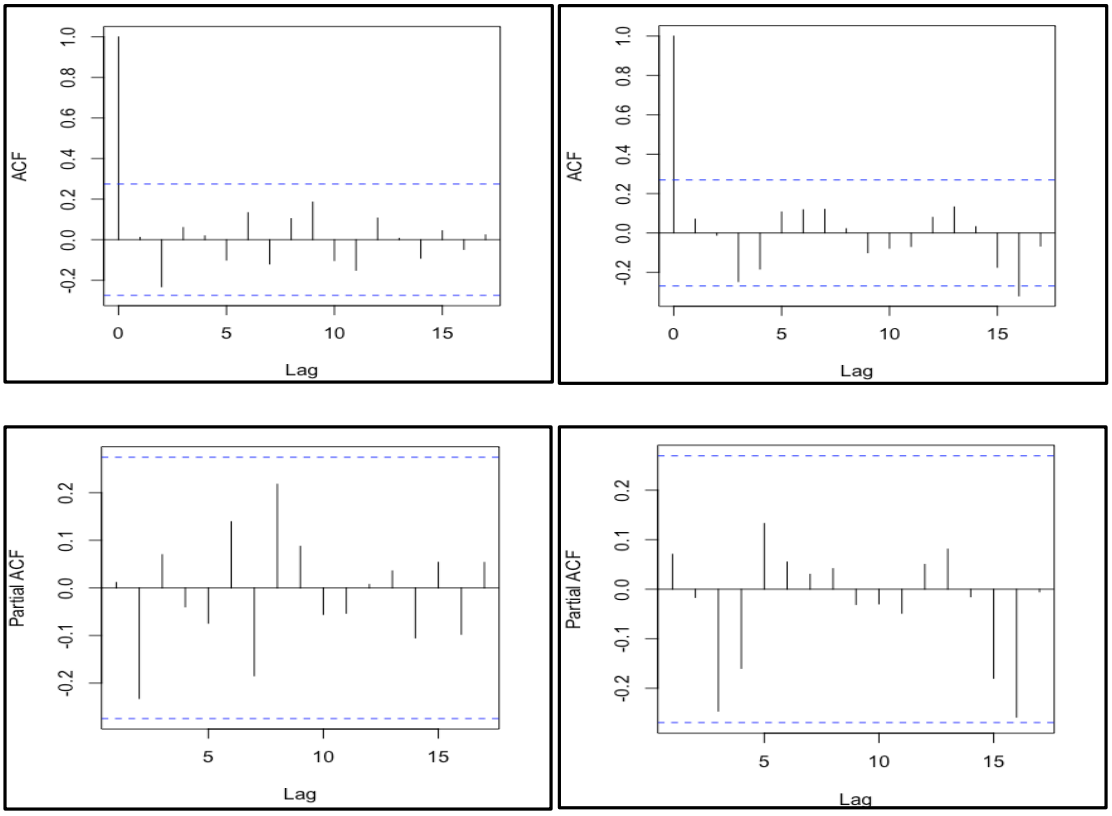

Figure 3: *ACF and PACF examples for the AR(3) model (left) and ARIMA(3,1,3) model (right)*

Calculating the mean RMSEs for the Crime rates resulted in the AR(3) model having smaller measurements than the ARIMA(3,1,3) model for month 1 (M = 282.24, SD = 180.30 vs M = 1830.94, SD = 1915.28.), month 2 (M = 300.09, SD = 159.71 vs MD = 2362.82, SD = 2946.43) and month 3 (M = 410.03, SD = 267.99 vs MD = 2586.03, SD = 2652.23) (for the other Crime rates see Table A2 and Table A4 in Appendix).

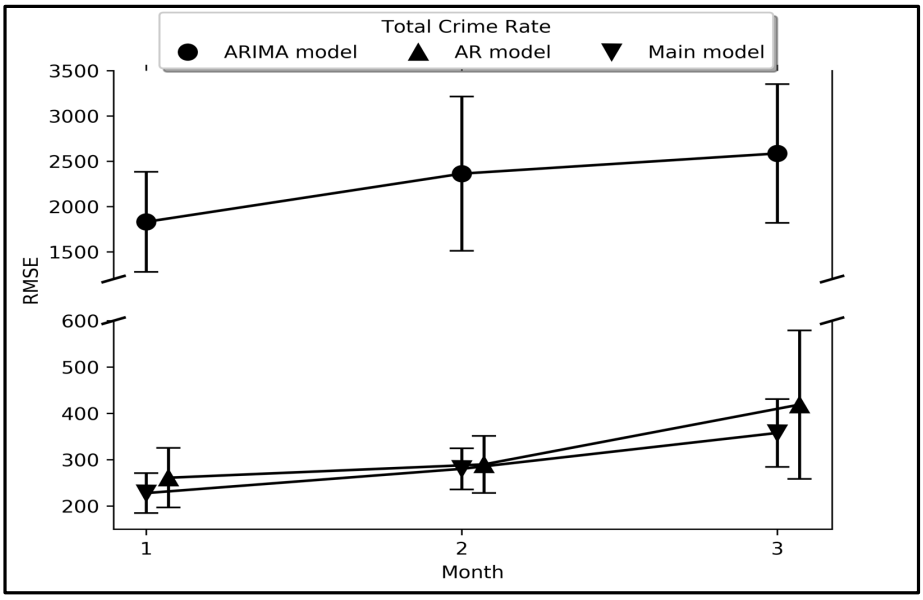

Figure 4: *Models' RMSEs for total crime rates*

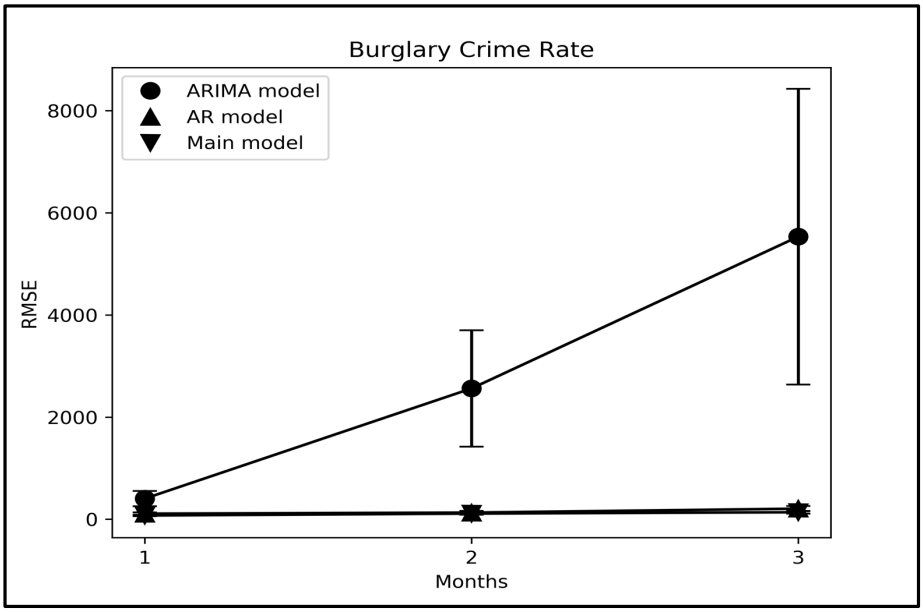

Figure 5: *Models' RMSEs for burglary crime rates*

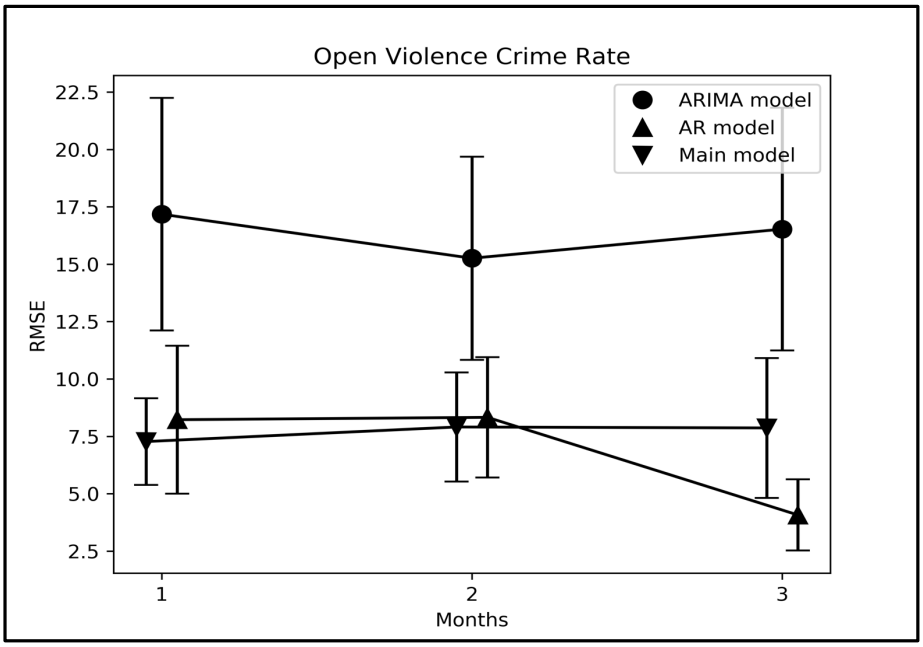

Figure 6: *Models' RMSEs for open violence crime rates*

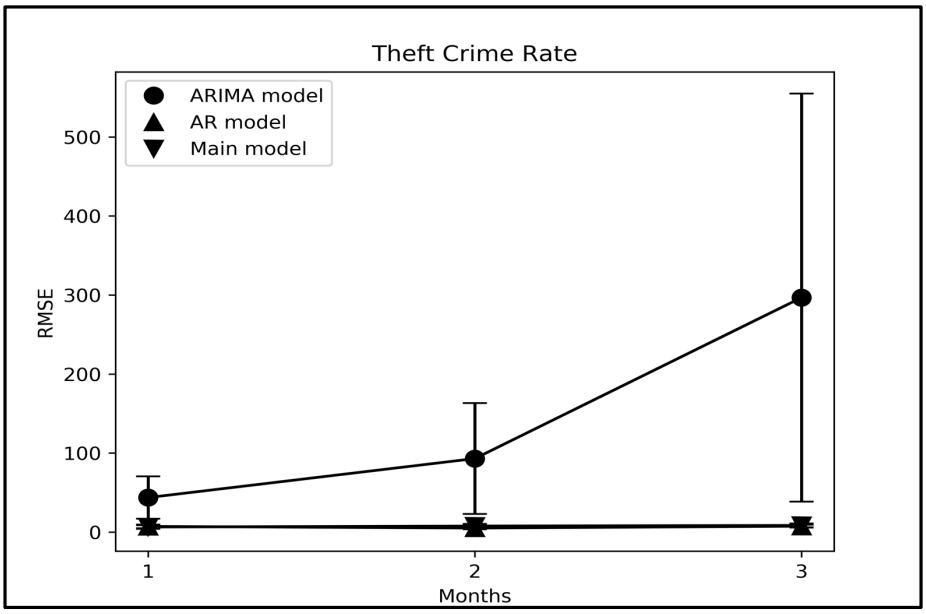

Figure 7: *Models' RMSE for theft crime rates*

Discussion

The focus of this study was to investigate whether the use of GSFs would be able to enhance the accuracy of a quantitative model in forecasting the development of crime. In order

to do so, the forecasting-ability of a quantitative model for general and specific types of crime that included GSFs was compared with the forecasting-ability of a quantitative model that did not include GSFs (Alternative model vs Main model).

The results show that the models that include GSFs (Alternative models) have no higher model fit than the model without GSFs (Main model). There seems to be no difference between the Main model and the AR(3) model in their effectiveness and even a decrease of the effect when it comes to the ARIMA(3,1,3) model which was already indicated by the ACF and PACF plots. While the higher average R squared value of the AR(3) model indicated a higher effectiveness of the model, the RMSEs could not verify that assumption (see Figure 4 to 7).

Doing a posterior test by calculating the average adjusted R square value showed that the AR(3) model seems to have actually no higher effect than the Main model. This indicated that the higher average R squared value of the AR(3) model was most likely caused by including a larger number of predictors. In summary, the results delivered little to no evidence for the hypothesis that GSFs provide additional predictive information for forecasting the development of crime. In addition, including the rates of the HIC sub-categories appeared to be non-effective, since the differences between the models' RMSEs remained constant (see Table A2, Table A3 and Table A4 in Appendix).

However, the non-significance of the results may imply that only the GSFs of these particular terms were not useful to enhance the forecasting-ability of a model. This was an explorative study; GSFs of terms were used from which we assumed that they would reflect someone's intention to commit a crime. The selection of terms was not empirically supported, since there seems to be no scientific literature committed to that kind of issue. Therefore, one cannot be sure whether using GSFs of other terms would or would not have led to more promising results. Consequently, not being able to find significant results with these particular GSFs should not be taken as a suggestion to completely drop explorations between GSFs and the prediction of crime.

Alternatively, trying to answer the research question could have been approached by exploiting a slightly different method. In this study, only a selected number of search terms was used from which we specifically suspected that they would be positively related to the development of crime. Thus, we made use of a so-called theory driven (or top-down) process. We could have also used a more data driven (or bottom-up) process. In other words, instead of using GSFs of only certain terms, one could have done an analysis with a large number of randomly chosen words. After that, words with a (positive) correlation to Crime rates could have been selected and eventually be included in the model. While it cannot be said for sure whether this approach would lead to different result, it may be a promising alternative to investigate in following studies.

Another option would have been to use the GSFs of the terms that were supposed to display the reaction to a crime. In this study, it was decided to use the words indicating the intention to carry out a crime. This was based on the assumption that the intention to commit a crime would be observable before the execution of said crime, while the reaction to that crime would only be observable at a later point. What we cannot control for sure is whether people who intend to commit a crime would actually use the Internet searching engine Google to collect information. Thus, executing an analysis with the GSFs of the reactional terms might be something to consider as a possible follow-up study in the future.

In summary, this study was a first exploratory step in using GSFs with the intention to predict crime. Even if the desired results were not found in this case, there are several ideas for alternative approaches that could be considered for follow-up studies.

Furthermore, the basic use of GSFs, despite several studies that indicated its success (e.g., Ginsberg et al., 2009; Wu & Brynjolfsson, 2015), should always be considered critically. As it was already discussed before, it is essential to keep in mind that GSFs are no total numbers of Google searches. They only reflect the proportion or the volume of the rates of searches regarding that topic, ranged from 0 to 100, for a certain localisation (e.g., country, region, city)

and for a certain time (e.g., year, month, week) (Choi & Varian, 2012; Wu & Brynjolfsson, 2015). On top of that, the GSFs can also vary for a certain point in time and location depending on the chosen time interval and the day the data are downloaded (Choi & Varian, 2012). Therefore, it should be also evaluated critically whether GSFs is a reliable method for the long-term use of a forecasting model. Nevertheless, the use of GSFs has appeared to be a promising method (e.g., Ginsberg et al., 2009; Wu & Brynjolfsson, 2015) and will certainly find more applications in the field of big 'social' data and prediction.

Regarding the analysis, some may voice their concerns that the use of an alternative, more complex statistical method could have led to more promising results. However, that option was deliberately omitted for this study. The most important reason was that McClendon and Meghanathan (2015) were already able to create a linear regression model on crime data that can be used to forecast future crime development. It was also considered as an advantage that a linear regression model appears relatively often in research; at least in the social sciences (Faraway, 2016). The use of a complex model would always require an expert with the necessary skills and the essential equipment (e.g., a computer that would be able to process a great amount of data in a relatively short time). While this may be no problem in academia, it could be an obstacle for an institution like the police. This, in turn, would lead to the question to what extent the costs of creating such a model are justified in relation to the benefit of it. Thus, the simplicity of the model would ensure that a great number of working professionals – at least the ones who were trained in social science – are able to use and understand the model.

In the beginning of the paper, the merits of being able to predict the development of crime were discussed, namely enabling the police to foresee the development of crime and detect locations where crimes are most likely to occur (McClendon & Meghanathan, 2015). This also may allow them to take more preventive actions instead of simply reacting to a crime and to be better prepared for an increase in crime (e.g., early communication with civilians and other institutions, sensible distribution of field staff, etc.).

However, one can argue that using forecasting models to predict crime could also lead to the police developing biases or even prejudices (Hvistendahl, 2016). For example, concerns have been expressed that crime prediction will result in increased arrests in the predicted areas without an actual increased crime (Brantingham, Valasik, & Mohler, 2018). In addition, many organizations fear that this bias, along with ethical prejudice, will only result in increased arresting rates of ethnic minority groups (Hvistendahl, 2016). Although there is literature that seems to prove the opposite (e.g., Brantingham et al., 2018), the debate still persists. Therefore, despite the success of some forecasting models, it should always be considered critically that such predictions may also have some negative consequences (Hvistendahl, 2016).

In conclusion, getting more and more access to big data certainly offers new ways of recognizing and interpreting socially relevant behaviour (Connelly et al., 2016), like in the field of crime investigation (Wang et al., 2016; Williams et al., 2017). Although some interesting research approaches, like the ones mentioned above, do exist that use big data for crime prediction, the option to combine crime rates with big 'social' data seems to be relatively unexplored (Williams et al., 2017). Therefore, the focus of this study was it to investigate whether GSFs could possibly contribute to the prediction of crime development. Even though the results in this study were non-significant, further investigations should be done before the making of a final statement.

Acknowledgments

I would like to thank my first and second supervisors, dr. ir. P. de Vries and dr. M. Stel, for their support and advice in this project. Also, I would like to thank L. Spieß (M.Sc.) for lending me his professional expertise (and always an open ear). Last but not least, thanks to all the beloved people who stood by my side during the last years. Without you, I would not have made it.

References

Brantingham, P. J., Valasik, M., & Mohler, G. O. (2018). Does predictive policing lead to biased arrests? Results from a randomized controlled trial. *Statistics and Public Policy*, *5*(1), 1 – 6. doi: 10.1080/2330443X.2018.1438940

Byström, K., & Järvelin, K. (1995). Task complexity affects information seeking and use. *Information Processing & Management*, *31*(2), 191 – 213. doi: 10.1016/0306-4573(95)80035-R

Centrum voor Criminaliteitspreventie en Veiligheid (CCV) (n.d.). *Criminele omgeving*. Retrieved from https://hetccv.nl/onderwerpen/veiligheidsbeleving/beinvloedbare-factoren/criminele-omgeving/

Choi, H., & Varian, H. (2012). Predicting the present with Google Trends. *Economic Record*, *88*(s1), 2 – 9. doi: 10.1111/j.1475-4932.2012.00809.x

Connelly, R., Playford, C. J., Gayle, V., & Dibben, C. (2016). The role of administrative data in the big data revolution in social science research. *Social Science Research*, *59*, 1 – 12. doi: 10.1016/j.ssresearch.2016.04.015

Faraway, J. J. (2016). *Linear models with R* (2nd ed.). New York: Taylor & Francis. doi: 10.1201/b17144

Freedman, M., Owens, E., & Bohn, S. (2018). Immigration, employment opportunities, and criminal behavior. *American Economic Journal: Economic Policy*, *10*(2), 117 – 151. doi: 10.1257/pol.20150165

Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S., & Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. *Nature*, *457*(7232), 1012 – 1014. doi: 10.1038/nature07634

Hvistendahl, M. (2016). Crime forecasters. *Science, 353*(6307), 1484 – 1487. doi: 10.1126/science.353.6307.1484

Kievik, M., & Gutteling, J. M. (2011). Yes, we can: motivate Dutch citizens to engage in self-protective behavior with regard to flood risks. *Natural hazards*, *59*(3), 1475 – 1490. doi: 10.1007/s11069-011-9845-1

McClendon, L., & Meghanathan, N. (2015). Using machine learning algorithms to analyze crime data. *Machine Learning and Applications: An International Journal (MLAIJ)*, *2*(1), 1 – 12. doi: 10.5121/mlaij.2015.2101

Office for National Statistics (ONS) (2018). *Internet access – households and individuals* [Data file]. Retrieved from https://www.ons.gov.uk/peoplepopulationandcommunity/householdcharacteristics/homeinternetandsocialmediausage/datasets/internetaccesshouseholdsandindividualsreferencetables

Office for National Statistics (ONS) (2017). *UK Population 2017* [Data file]. Retrieved from https://www.ons.gov.uk/aboutus/transparencyandgovernance/freedomofinformationfoi/ukpopulation2017

R Core Team (2018). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. Retrieved from http://www.R-project.org/

Regioburgemeesters (n.d.). *High Impact Criminaliteit*. Retrieved from http://www.regioburgemeesters.nl/thema/aanpak-onveiligheid/high-impact/

Savolainen, R. (2018). Self-determination and expectancy-value: Comparison of cognitive psychological approaches to motivators for information seeking about job opportunities. *Aslib Journal of Information Management*, *70*(1), 123 – 140. doi: 10.1108/AJIM-10-2017-0242

Shumway, R. H., & Stoffer, D. S. (2017). *Time series analysis and its applications: with R examples* (4th ed.). Cham: Springer.

Statista (2018). *Number of social network users in selected countries in 2017 and 2022 (in millions)*. Retrieved from https://www.statista.com/statistics/278341/number-of-social-network-users-in-selected-countries/

Wang, H., Kifer, D., Graif, C., & Li, Z. (2016). Crime rate inference with big data. In *KDD 2016 - Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 635-644). (Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; Vol. 13-17-August-2016). Association for Computing Machinery. doi: 10.1145/2939672.2939736

Williams, M. L., Burnap, P., & Sloan, L. (2017). Crime sensing with big data: The affordances and limitations of using open-source communications to estimate crime patterns. *The British Journal of Criminology*, *57*(2), 320 – 340. doi:10.1093/bjc/azw031

Wu, L., & Brynjolfsson, E. (2015). The future of prediction: How Google searches foreshadow housing prices and sales. In A. Goldfarb, S. M. Greenstein, & C. E. Tucker (Eds.), *Economic analysis of the digital economy* (pp. 89 – 118). doi: 10.7208/chicago/9780226206981.003.0003

Appendix A

Table 1:

List of 90 Dutch terms that are supposed to be related to crime intention and reaction.

Terms for crime intention	Terms for crime reaction
1. Bom	Inbraak
2. Loper	Woninginbraak
3. Slothaak	Diefstal
4. Gif	Geweldsmisdrijf
5. Venijn	Straatroof
6. Doden	Overval
7. Pickpocket	Autodiefstal
8. Betrappen	Buurtpreventie
9. Mes	Zelfverdediging
10. Pistool	Verkrachting
11. Taser	Winkeldiefstal
12. Opbreken	Gebroken raam
13. Knaldemper	Safe
14. Openbaar Publiek	Kluis
15. Politiële optreden	Zakkenrollen
16. Vluchtweg	Deurgrendel
17. Nummerplaatherkenning	Deurslot
18. Geldtransport	Aantal diefstal
19. vaak gebruikt wachtwoord	Waakhond
20. Strafvervolging	Angst voor verkrachting
21. Strafrechtelijk	Seksuele aanranding
22. Vervalsen	Gestolen
23. Witwassen geld	Moord
24. Identificatie	Doodslag
25. Vingerafdruk	Gewapend
26. Vrijuit gaan	Pepperspray
27. Drukbezocht	Noodweer
28. Knock-Out drug	Huisvredebreuk
29. Bouwplan	Zaakbeschadiging
30. Springlading	Verzekering
31. PIN vaak gebruikt	Bedreiging
32. Explosieven	Bescherming
33. Uncle Fester	Hinderlaag
34. Croftybom	Delict
35. Natronloog	Kliklijn
36. Aluminium	Deurspion
37. Acetonperoxide	Veiligheidsslot
38. Waterstofgas	Alarminrichting
39. Zwavelzuur	Dader
40. Kaliumchloride	Onveilig wijk
41. Patroon	Percentage criminaliteit
42. Perfecte misdaad	Inbraakwerend
43. Verdovingsmiddel	Fraude
44. Knock-outdruppels	Veilig thuis

45. Vlindermes	Diefstalverzekering

Table 2:

AR(3) model RMSEs (Mean and SD)

Crime rates	**Month 1**	**Month 2**	**Month 3**
total	*M* = 282.24 *SD* = 180.30	*M* = 300.09 *SD* = 159.71	*M* = 410.03 *SD* = 267.99
burglary (‘diefstal’)	*M* = 93.18 *SD* = 78.88	*M* = 124.91 *SD* = 103.78	*M* = 148.46 *SD* = 104.69
open violence (‘open geweld’)	*M* = 8.15 *SD* = 7.56	*M* = 9.35 *SD* = 7.45	*M* = 9.30 *SD* = 10.94
theft (‘straatroof’)	*M* = 6.63 *SD* = 6.89	*M* = 8.20 *SD* = 8.08	*M* = 10.71 *SD* = 12.48

Table 3:

Main model RMSEs (Mean and SD)

Crime rates	**Month 1**	**Month 2**	**Month 3**
total	*M* = 227.97 *SD* = 149.64	*M* = 280.31 *SD* = 153.87	*M* = 357.84 *SD* = 253.59
burglary (‘diefstal’)	*M* = 73.76 *SD* = 53.27	*M* = 111.22 *SD* = 83.66	*M* = 137.75 *SD* = 96.58
open violence (‘open geweld’)	*M* = 7.27 *SD* = 6.53	*M* = 7.91 *SD* = 8.23	*M* = 7.87 *SD* = 10.55
theft (‘straatroof’)	*M* = 6.35 *SD* = 7.95	*M* = 7.64 *SD* = 9.05	*M* = 8.42 *SD* = 8.53

Table 4:

ARIMA(3,1,3) model RMSEs (Mean and SD)

Crime rates	**Month 1**	**Month 2**	**Month 3**
total	*M* = 1830.94 *SD* = 1915.28	*M* = 2362.82 *SD* = 2946.43	*M* = 2586.03 *SD* = 2652.23
burglary (‘diefstal’)	*M* = 404.93 *SD* = 519.76	*M* = 2561.17 *SD* = 3949.47	*M* = 5533.00 *SD* = 10022.82
open violence (‘open geweld’)	*M* = 17.18 *SD* = 17.55	*M* = 15.26 *SD* = 15.34	*M* = 16.53 *SD* = 18.32
theft (‘straatroof’)	*M* = 43.65 *SD* = 93.13	*M* = 92.98 *SD* = 242.99	*M* = 296.68 *SD* = 894.25

Appendix B

R Script

```
## Data directories
Trends_data = "~/Desktop/Intentie_Data/Data/GTrendsData/"
Pop_data = "~/Desktop/Intentie_Data/Data/PopulationData/"
CR_data = "~/Desktop/Intentie_Data/Data/CRData/"

predict_1 <- function(model, design_matrix) {
	counter = 0
	for (coeff in 1 : length(model$coefficients)){
		coefficient = model$coefficients[coeff]

		if (counter == 0){
			y = coefficient
		}else{
			y = y + (coefficient*design_matrix[counter])
		}
		counter = counter + 1
		#print(y)
	}
	return(y)
}

RMSE = function(m, o){
	sqrt(mean((m - o)^2))
}

##
make_model = function(dependent_var, design_matrix){
	model = lm(dependent_var ~ design_matrix)
	return(model)
}

##

mk_designMatrix = function(){

}


# READ-IN ALL THE DATA
################################################################################
################################################

# Some important variables
DATAMAT <- array(rep(NaN, 72*51*12), c(72, 51, 12)) # in here, we store all the data

#### READ-IN ALL THE DATA
# GTrends Data
```

```
all.the.files <- list.files(path = Trends_data, pattern = ".csv", full.names = TRUE) # also tells us the order of regions in DATAMAT
all.the.data <- list()
c = 1
for (i in all.the.files)
{
tmp = read.csv(i,skip = 0, header = TRUE, sep = "\t")
DATAMAT[1:72,1:45,c] = data.matrix(tmp[,2:46])
c = c + 1
}

# CR data
all_files = list.files(path = CR_data, pattern = ".csv", full.names = TRUE)
sort_order = c(4,2,1,9,8,3,12,7,11,10,6,5) # in here is the order of the indices such that we can assign the CR data properly to the regions specified in DATAMAT
cr_var = 1
for (file in all_files)
{
	start = 1
	stop = 72
	for (region in 1:length(sort_order))
	{
		tmp = read.csv(file,skip = 0, header = TRUE, sep = ";")
		tmp = tmp[73:nrow(tmp),]
		DATAMAT[1:72,45+cr_var,sort_order[region]] = data.matrix(tmp[start:stop,4])
		start = start + 72
		stop = stop + 72
	}
	cr_var = cr_var + 1
}

# Popo data
all_files = list.files(path = Pop_data, pattern = ".csv", full.names = TRUE)
sort_order = c(4,2,1,9,8,3,12,7,11,10,6,5) # in here is the order of the indices such that we can assign the CR data properly to the regions specified in DATAMAT
pop = read.csv(all_files,skip = 1, header = FALSE, sep = ";", as.is = FALSE)


for (region in 1:12)
	{counter = 4
	for (element in 1:length(pop[region,4:75]))
	    {
		DATAMAT[element,51,sort_order[region]] = strtoi(gsub(' ','',
data.matrix(pop[region,counter])))
		counter = counter + 1
	}
}


# DATA PREPARATION
```

```
################################################################################
##############################

### Split data in training and test-set
Training_data = DATAMAT[1:54,,]
Test_data = DATAMAT[55:72,,]


### Check missing data distribution
missing_vals = array(rep(NaN, 12*6), c(12,6))
for (region in 1:12){
	counter = 1
	for (variable in 46:51){
	    missing_vals[region, counter] = sum(is.na(Training_data[,variable,region]))
	    counter = counter + 1
	}
}

# ==> Overvall has too many missing data (variable index = 49)


### Calculate correlation (separately for each crime rate (except overvall))
correlation_matrix = array(rep(NaN, 12*45*4), c(12,45,4)) # in here, we store all the relevant
correlations
var_list = c(46,47,48,50) # only use these crime rate variables 'cos overvall we don't wanna
have

# Populate the correlation matrix
for (crimeRate in 1:4){
	for (region in 1:12){
		for (variable in 1:45){
			correlation_matrix[region, variable,crimeRate] =
cor(Training_data[,c(variable,var_list[crimeRate]),region])[1,2]
		}
	}
}
# ==> If a word has no frequencies (all values = 0), then cor will return Na

### Shortlist variables by filtering correlations per region and word
correl_thresholds = 0 # filter correlations by only taking those that are larger than the threshold

candidate_vars = correlation_matrix > correl_thresholds # returns matrix with same
dimensionality as correlation_matrix
candidate_vars[is.na(candidate_vars)] = FALSE
### ===> We use candidate_vars as an index matrix to select per region the shortlisted word
variables

### Calculate number of variables (i.e., words) per region
var_stats = array(rep(NaN, 12*4), c(12,4))
for (crimeRate in 1:4){
	for (region in 1:12){
```

```
            var_stats[region, crimeRate] = sum(candidate_vars[region,,crimeRate])
      }
}


# Model building
################################################################################
################################

####
#### AR(3) regression
####
NULL_lin_models_tmp = list()
ALT_lin_models_tmp  = list()
counter = 1
var_list = c(46,47,48,50)
tmp = Training_data[3:53,1:45,] # take care that we have to create lagged versions
for (region in 1:12){
      for (crimeRate in 1 : length(var_list)){
            if (var_stats[region,crimeRate] > 0){
                  ### Create design matrix
                  # Second last column = lagged crime rate; last column = population data
                  dependent_var = Training_data[4:54,var_list[crimeRate],region]
                  design_matrix     =     cbind(tmp[,c(candidate_vars[region,,crimeRate]),region],
Training_data[1:51,var_list[crimeRate],region],Training_data[2:52,var_list[crimeRate],region
], Training_data[3:53,var_list[crimeRate],region],Training_data[3:53,51,region])
                  model = lm(dependent_var ~ design_matrix)
                  ALT_lin_models_tmp[[counter]] = model

                  design_matrix                                                    =
cbind(Training_data[1:51,var_list[crimeRate],region],
Training_data[2:52,var_list[crimeRate],region],
Training_data[3:53,var_list[crimeRate],region],Training_data[3:53,51,region])
                  model = lm(dependent_var ~ design_matrix)
                  NULL_lin_models_tmp[[counter]] = model
            } else{
                  # put NaN in case there are no variables
                  ALT_lin_models_tmp[[counter]] = NaN
                  NULL_lin_models_tmp[[counter]] = NaN
            }
            counter = counter +1
      }
}


# Check R^2
################################################################################
################################
r_alt = array(rep(NaN,48))
r_null = array(rep(NaN,48))
```

```
for (model in 1:48){
      if (is.na(ALT_lin_models_tmp[[model]]) != TRUE){
      r_alt[model] = summary(ALT_lin_models_tmp[[model]])$r.squared
      r_null[model] = summary(NULL_lin_models_tmp[[model]])$r.squared

}
}
print(summary(r_alt))
print(summary(r_null))


#acf(NULL_lin_models_tmp[[]]$residuals)
#acf(ALT_lin_models_tmp[[]]$residuals)

#pacf(NULL_lin_models_tmp[[]]$residuals)
#pacf(ALT_lin_models_tmp[[]]$residuals)

# Model predictions
##############################################################################
####################################
# Iteratively


var_list = c(46,47,48,50)

tmp = DATAMAT[1:72,1:45,]
month_predictions = array(rep(NaN,3))

replica_pred_error_matrix = array(rep(NaN, 6*3), c(6,3))
region_pred_error_matrix = array(rep(NaN, 12*3), c(12,3))
crime_pred_error_matrix =  array(rep(NaN, 4*3), c(4,3))
crime_pred_error_matrix_SD = array(rep(NaN, 4*3), c(4,3))

counter_b = 1
counter_c = 2
counter_ade = 3

counter.B = 52
counter.C = 53
counter.ADE = 54

dep_1 = 4
dep_2 = 5
dep_3 = 6

true_dep = 55
for (crimeRate in 1:length(var_list)){

      for (region in 1:12){

            true_dep = 55
```

```
counter_b = 1
counter_c = 2
counter_ade = 3

counter.B = 52
counter.C = 53
counter.ADE = 54

dep_1 = 4
dep_2 = 5
dep_3 = 6

if (sum(candidate_vars[region,,crimeRate]) > 0){
      for (Replica in 1 : 6){

            for (iMonth in 1 : 3){

                  ### Create Design Matrix ###
                  if (iMonth == 1){
                        a                                                   =
tmp[counter_ade:(counter_ade+50),c(candidate_vars[region,,crimeRate]),region]
                        b                                                   =
DATAMAT[counter_b:(counter_b+50),var_list[crimeRate],region]
                        c                                                   =
DATAMAT[counter_c:(counter_c+50),var_list[crimeRate],region]
                        d                                                   =
DATAMAT[counter_ade:(counter_ade+50),var_list[crimeRate],region]
                        e = DATAMAT[counter_ade:(counter_ade+50),51,region]
                        dependent_var                                       =
c(DATAMAT[dep_1:(dep_1+50),var_list[crimeRate],region])
                        design_matrix = cbind(a,b,c,d,e)
                        ### make model
                        m1 = make_model(dependent_var, design_matrix)
                        ### make design marix for new predictions
                        a                                                   =
tmp[counter.ADE,c(candidate_vars[region,,crimeRate]),region]
                        b = DATAMAT[counter.B,var_list[crimeRate],region]
                        c = DATAMAT[counter.C,var_list[crimeRate],region]
                        d = DATAMAT[counter.ADE,var_list[crimeRate],region]
                        e = DATAMAT[counter.ADE,51,region]
                        design_matrix = c(a,b,c,d,e)
                        ### Make prediction
                        y = predict_1(m1, design_matrix)
                        month_predictions[1] = y
                        replica_pred_error_matrix[Replica,iMonth]           =
RMSE(y,DATAMAT[true_dep,var_list[crimeRate],region])

                  }else if (iMonth == 2){
                        a                                                   =
tmp[counter_ade:(counter_ade+50),c(candidate_vars[region,,crimeRate]),region]
```

```
                    b                                                   =
DATAMAT[counter_b:(counter_b+50),var_list[crimeRate],region]
                    c                                                   =
DATAMAT[counter_c:(counter_c+50),var_list[crimeRate],region]
                    d                                                   =
DATAMAT[counter_ade:(counter_ade+50),var_list[crimeRate],region]
                    e = DATAMAT[counter_ade:(counter_ade+50),51,region]
                    dependent_var                                       =
c(DATAMAT[dep_2:(dep_2+49),var_list[crimeRate],region],y)
                    design_matrix = cbind(a,b,c,d,e)
                    ### make model
                    m2 = make_model(dependent_var, design_matrix)
                    ### make design marix for new predictions
                    a                                                   =
tmp[counter.ADE,c(candidate_vars[region,,crimeRate]),region]
                    b = DATAMAT[counter.B,var_list[crimeRate],region]
                    c = DATAMAT[counter.C,var_list[crimeRate],region]
                    d = DATAMAT[counter.ADE,var_list[crimeRate],region]
                    e = DATAMAT[counter.ADE,51,region]
                    design_matrix = c(a,b,c,d,e)
                    ### make prediction
                    y = predict_1(m2, design_matrix)
                    month_predictions[2] = y
                    replica_pred_error_matrix[Replica,iMonth]           =
RMSE(y,DATAMAT[(true_dep+1),var_list[crimeRate],region])
                }else{
                    a                                                   =
tmp[counter_ade:(counter_ade+50),c(candidate_vars[region,,crimeRate]),region]
                    b                                                   =
DATAMAT[counter_b:(counter_b+50),var_list[crimeRate],region]
                    c                                                   =
DATAMAT[counter_c:(counter_c+50),var_list[crimeRate],region]
                    d                                                   =
DATAMAT[counter_ade:(counter_ade+50),var_list[crimeRate],region]
                    e = DATAMAT[counter_ade:(counter_ade+50),51,region]
                    dependent_var                                       =
c(DATAMAT[dep_3:(dep_3+48),var_list[crimeRate],region],month_predictions[1],month_pr
edictions[2])
                    design_matrix = cbind(a,b,c,d,e)
                    ### make model
                    m3 = make_model(dependent_var, design_matrix)
                    ### make design marix for new predictions
                    a                                                   =
tmp[counter.ADE,c(candidate_vars[region,,crimeRate]),region]
                    b = DATAMAT[counter.B,var_list[crimeRate],region]
                    c = DATAMAT[counter.C,var_list[crimeRate],region]
                    d = DATAMAT[counter.ADE,var_list[crimeRate],region]
                    e = DATAMAT[counter.ADE,51,region]
                    design_matrix = c(a,b,c,d,e)
                    ### make prediction
                    y = predict_1(m3, design_matrix)
```

```
                                month_predictions[3] = y
                                replica_pred_error_matrix[Replica,iMonth]                    =
RMSE(y,DATAMAT[(true_dep+2),var_list[crimeRate],region])
                            }

                            region_pred_error_matrix[region,1:3]                             =
colMeans(replica_pred_error_matrix, na.rm = TRUE)

                    }
                        crime_pred_error_matrix[crimeRate,1:3]                               =
colMeans(region_pred_error_matrix, na.rm = TRUE)

                        crime_pred_error_matrix_SD[crimeRate,1]                             <-
sd(region_pred_error_matrix[,1], na.rm = TRUE)
                        crime_pred_error_matrix_SD[crimeRate,2]                             <-
sd(region_pred_error_matrix[,2], na.rm = TRUE)
                        crime_pred_error_matrix_SD[crimeRate,3]                             <-
sd(region_pred_error_matrix[,3], na.rm = TRUE)



                         ### Update counters


                        counter_b = counter_b + 1
                        counter_c = counter_c + 1
                        counter_ade = counter_ade + 1

                        counter.B = counter.B + 1
                        counter.C = counter.C + 1
                        counter.ADE = counter.ADE + 1

                        dep_1 = dep_1 + 1
                        dep_2 = dep_2 + 1
                        dep_3 = dep_3 + 1

                        true_dep = true_dep + 1
                    }
                }else{
                    #print(sum(candidate_vars[region,,crimeRate]))
                }

        } # end region

} # end crime rate



# Model predictions NULL
#########################################################################
#####################################
```

```
# Iteratively


var_list = c(46,47,48,50)

#tmp = DATAMAT[1:72,1:45,]
month_predictions = array(rep(NaN,3))

replica_pred_error_matrixNULL = array(rep(NaN, 6*3), c(6,3))
region_pred_error_matrixNULL = array(rep(NaN, 12*3), c(12,3))
crime_pred_error_matrixNULL =  array(rep(NaN, 4*3), c(4,3))
crime_pred_error_matrix_SDNULL = array(rep(NaN, 4*3), c(4,3))

counter_b = 1
counter_c = 2
counter_ade = 3

counter.B = 52
counter.C = 53
counter.ADE = 54

dep_1 = 4
dep_2 = 5
dep_3 = 6

true_dep = 55
for (crimeRate in 1:length(var_list)){

      for (region in 1:12){

            true_dep = 55

            counter_b = 1
            counter_c = 2
            counter_ade = 3

            counter.B = 52
            counter.C = 53
            counter.ADE = 54

            dep_1 = 4
            dep_2 = 5
            dep_3 = 6

            if (sum(candidate_vars[region,,crimeRate]) > 0){

                  for (Replica in 1 : 6){

                        for (iMonth in 1 : 3){

                              ### Create Design Matrix ###
```

```
                    if (iMonth == 1){
                        #a                                                  =
tmp[counter_ade:(counter_ade+50),c(candidate_vars[region,,crimeRate]),region]
                        b                                                   =
DATAMAT[counter_b:(counter_b+50),var_list[crimeRate],region]
                        c                                                   =
DATAMAT[counter_c:(counter_c+50),var_list[crimeRate],region]
                        d                                                   =
DATAMAT[counter_ade:(counter_ade+50),var_list[crimeRate],region]
                        e = DATAMAT[counter_ade:(counter_ade+50),51,region]
                        dependent_var                                       =
c(DATAMAT[dep_1:(dep_1+50),var_list[crimeRate],region])
                        design_matrix = cbind(b,c,d,e)
                        ### make model
                        m1 = make_model(dependent_var, design_matrix)
                        ### make design marix for new predictions
                        #a                                                  =
tmp[counter.ADE,c(candidate_vars[region,,crimeRate]),region]
                        b = DATAMAT[counter.B,var_list[crimeRate],region]
                        c = DATAMAT[counter.C,var_list[crimeRate],region]
                        d = DATAMAT[counter.ADE,var_list[crimeRate],region]
                        e = DATAMAT[counter.ADE,51,region]
                        design_matrix = c(b,c,d,e)
                        ### Make prediction
                        y = predict_1(m1, design_matrix)
                        month_predictions[1] = y
                        replica_pred_error_matrixNULL[Replica,iMonth]       =
RMSE(y,DATAMAT[true_dep,var_list[crimeRate],region])

                    }else if (iMonth == 2){
                        #a                                                  =
tmp[counter_ade:(counter_ade+50),c(candidate_vars[region,,crimeRate]),region]
                        b                                                   =
DATAMAT[counter_b:(counter_b+50),var_list[crimeRate],region]
                        c                                                   =
DATAMAT[counter_c:(counter_c+50),var_list[crimeRate],region]
                        d                                                   =
DATAMAT[counter_ade:(counter_ade+50),var_list[crimeRate],region]
                        e = DATAMAT[counter_ade:(counter_ade+50),51,region]
                        dependent_var                                       =
c(DATAMAT[dep_2:(dep_2+49),var_list[crimeRate],region],y)
                        design_matrix = cbind(b,c,d,e)
                        ### make model
                        m2 = make_model(dependent_var, design_matrix)
                        ### make design marix for new predictions
                        #a                                                  =
tmp[counter.ADE,c(candidate_vars[region,,crimeRate]),region]
                        b = DATAMAT[counter.B,var_list[crimeRate],region]
                        c = DATAMAT[counter.C,var_list[crimeRate],region]
                        d = DATAMAT[counter.ADE,var_list[crimeRate],region]
                        e = DATAMAT[counter.ADE,51,region]
```

```
                    design_matrix = c(b,c,d,e)
                    ### make prediction
                    y = predict_1(m2, design_matrix)
                    month_predictions[2] = y
                    replica_pred_error_matrixNULL[Replica,iMonth]          =
RMSE(y,DATAMAT[(true_dep+1),var_list[crimeRate],region])
                }else{
                    #a                                                     =
tmp[counter_ade:(counter_ade+50),c(candidate_vars[region,,crimeRate]),region]
                    b                                                     =
DATAMAT[counter_b:(counter_b+50),var_list[crimeRate],region]
                    c                                                     =
DATAMAT[counter_c:(counter_c+50),var_list[crimeRate],region]
                    d                                                     =
DATAMAT[counter_ade:(counter_ade+50),var_list[crimeRate],region]
                    e = DATAMAT[counter_ade:(counter_ade+50),51,region]
                    dependent_var                                         =
c(DATAMAT[dep_3:(dep_3+48),var_list[crimeRate],region],month_predictions[1],month_pr
edictions[2])
                    design_matrix = cbind(b,c,d,e)
                    ### make model
                    m3 = make_model(dependent_var, design_matrix)
                    ### make design marix for new predictions
                    #a                                                     =
tmp[counter.ADE,c(candidate_vars[region,,crimeRate]),region]
                    b = DATAMAT[counter.B,var_list[crimeRate],region]
                    c = DATAMAT[counter.C,var_list[crimeRate],region]
                    d = DATAMAT[counter.ADE,var_list[crimeRate],region]
                    e = DATAMAT[counter.ADE,51,region]
                    design_matrix = c(b,c,d,e)
                    ### make prediction
                    y = predict_1(m3, design_matrix)
                    month_predictions[3] = y
                    replica_pred_error_matrixNULL[Replica,iMonth]          =
RMSE(y,DATAMAT[(true_dep+2),var_list[crimeRate],region])
                }

                region_pred_error_matrixNULL[region,1:3]                  =
colMeans(replica_pred_error_matrixNULL, na.rm = TRUE)


            }
            crime_pred_error_matrixNULL[crimeRate,1:3]                    =
colMeans(region_pred_error_matrixNULL, na.rm = TRUE)

            crime_pred_error_matrix_SDNULL[crimeRate,1]                  <-
sd(region_pred_error_matrixNULL[,1], na.rm = TRUE)
            crime_pred_error_matrix_SDNULL[crimeRate,2]                  <-
sd(region_pred_error_matrixNULL[,2], na.rm = TRUE)
            crime_pred_error_matrix_SDNULL[crimeRate,3]                  <-
sd(region_pred_error_matrixNULL[,3], na.rm = TRUE)
```

```
                    ### Update counters
                    counter_b = counter_b + 1
                    counter_c = counter_c + 1
                    counter_ade = counter_ade + 1

                    counter.B = counter.B + 1
                    counter.C = counter.C + 1
                    counter.ADE = counter.ADE + 1

                    dep_1 = dep_1 + 1
                    dep_2 = dep_2 + 1
                    dep_3 = dep_3 + 1

                    true_dep = true_dep + 1
                }
            }else{
                #print(sum(candidate_vars[region,,crimeRate]))
            }

        } # end region

} # end crime rate


################################################
################################################
################################################
############################################
#########################


# Check  adjusted R^2
#############################################################################
###############################

r.adj_alt = array(rep(NaN,48))
r.adj_null = array(rep(NaN,48))
for (model in 1:48){
    if (is.na(ALT_lin_models_tmp[[model]]) != TRUE){
        r.adj_alt[model] = summary(ALT_lin_models_tmp[[model]])$adj.r.squared
        r.adj_null[model] = summary(NULL_lin_models_tmp[[model]])$adj.r.squared
    }
}
print(summary(r.adj_alt))
print(summary(r.adj_null))
```

```
########################
################### comparing ALT with ARIMA approach

library(forecast)

#######
ARIMA_models_tmp = list()
counter = 1
var_list = c(46,47,48,50)
tmp = Training_data[1:53,1:45,] # take care that we have to create lagged versions
for (region in 1:12){
      for (crimeRate in 1 : length(var_list)){
            if (var_stats[region,crimeRate] > 0){
                  ### Create design matrix
                  # Second last column = lagged crime rate; last column = population data
                  dependent_var = Training_data[1:53,var_list[crimeRate],region]
                  design_matrix                                                    =
cbind(tmp[,c(candidate_vars[region,,crimeRate]),region],Training_data[1:53,51,region])
                  model = Arima(dependent_var, order = c(3, 1, 3), xreg = design_matrix, method
= "ML")
                  ARIMA_models_tmp[[counter]] = model

            } else{
                  # put NaN in case there are no variables
                  ARIMA_models_tmp[[counter]] = NaN

            }
            counter = counter +1
      }
}

tmp = Training_data[2:54,1:45,]
New_design_matrix                                                              =
cbind(tmp[,c(candidate_vars[region,,crimeRate]),region],Training_data[2:54,51,region])
y = predict(ARIMA_models_tmp[[1]], newxreg = New_design_matrix)


#acf(ARIMA_models_tmp[[]]$residuals)
#acf(ALT_lin_models_tmp[[]]$residuals)

#pacf(ARIMA_models_tmp[[]]$residuals)
#pacf(ALT_lin_models_tmp[[]]$residuals)

########### make predictions

var_list = c(46,47,48,50)

tmp = DATAMAT[1:72,1:45,]
month_predictions = array(rep(NaN,3))

ARIMAreplica_pred_error_matrix = array(rep(NaN, 6*3), c(6,3))
```

```
ARIMAregion_pred_error_matrix = array(rep(NaN, 12*3), c(12,3))
ARIMAcrime_pred_error_matrix =  array(rep(NaN, 4*3), c(4,3))
ARIMAcrime_pred_error_matrix_SD = array(rep(NaN, 4*3), c(4,3))

counter.i = 1
counter.I = 2

x_1 = 1
x_2 = 2
x_3 = 3

pred_1 = 2
pred_2 = 3
pred_3 = 4

true_dep = 55
for (crimeRate in 1:length(var_list)){
      for (region in 1:12){
            true_dep = 55

            counter.i = 1
            counter.I = 2

            x_1 = 1
            x_2 = 2
            x_3 = 3

            pred_1 = 2
            pred_2 = 3
            pred_3 = 4

            if (sum(candidate_vars[region,,crimeRate]) > 0){

                  for (Replica in 1 : 6){

                        for (iMonth in 1 : 3){

                              ### Create Design Matrix ###
                              if (iMonth == 1){
                                    a                                                   =
tmp[counter.i:(counter.i+52),c(candidate_vars[region,,crimeRate]),region]
                                    b = DATAMAT[counter.i:(counter.i+52),51,region]
                                    x = c(DATAMAT[x_1:(x_1+52),var_list[crimeRate],region])
                                    design_matrix = cbind(a,b)
                                    ### make model
                                    m1 = Arima(x, order = c(3, 1, 3), xreg = design_matrix, method
= "ML")
                                    ### make design marix for new predictions
                                    a                                                   =
tmp[counter.I:(counter.I+52),c(candidate_vars[region,,crimeRate]),region]
                                    b = DATAMAT[counter.I:(counter.I+52),51,region]
```

```
                    new_design_matrix = cbind(a,b)
                    ### Make prediction
                    y = predict(m1, newxreg = new_design_matrix)
                    month_predictions[1] = y$pred[pred_1]
                    ARIMAreplica_pred_error_matrix[Replica,iMonth]          =
RMSE(y$pred[pred_1],DATAMAT[true_dep,var_list[crimeRate],region])

                }else if (iMonth == 2){
                    a                                                       =
tmp[counter.i:(counter.i+52),c(candidate_vars[region,,crimeRate]),region]
                    b = DATAMAT[counter.i:(counter.i+52),51,region]
                    x   =   c(DATAMAT[x_2:(x_2+51),var_list[crimeRate],region],
month_predictions[1])
                    design_matrix = cbind(a,b)
                    ### make model
                    m2 = Arima(x, order = c(3, 1, 3), xreg = design_matrix, method
= "ML")
                    ### make design marix for new predictions
                    a                                                       =
tmp[counter.I:(counter.I+52),c(candidate_vars[region,,crimeRate]),region]
                    b = DATAMAT[counter.I:(counter.I+52),51,region]
                    new_design_matrix = cbind(a,b)
                    ### make prediction
                    y = predict(m2, newxreg = new_design_matrix)
                    month_predictions[2] = y$pred[pred_2]
                    ARIMAreplica_pred_error_matrix[Replica,iMonth]          =
RMSE(y$pred[pred_2],DATAMAT[(true_dep+1),var_list[crimeRate],region])

                }else{
                    a                                                       =
tmp[counter.i:(counter.i+52),c(candidate_vars[region,,crimeRate]),region]
                    b = DATAMAT[counter.i:(counter.i+52),51,region]
                    x   =   c(DATAMAT[x_3:(x_3+50),var_list[crimeRate],region],
month_predictions[1], month_predictions[2])
                    design_matrix = cbind(a,b)
                    ### make model
                    m3 = Arima(x, order = c(3, 1, 3), xreg = design_matrix, method
= "ML")
                    ### make design marix for new predictions
                    a                                                       =
tmp[counter.I:(counter.I+52),c(candidate_vars[region,,crimeRate]),region]
                    b = DATAMAT[counter.I:(counter.I+52),51,region]
                    new_design_matrix = cbind(a,b)
                    ### make prediction
                    y = predict(m3, newxreg = new_design_matrix)
                    month_predictions[3] = y$pred[pred_3]
                    ARIMAreplica_pred_error_matrix[Replica,iMonth]          =
RMSE(y$pred[pred_3],DATAMAT[(true_dep+2),var_list[crimeRate],region])

                }
```

```
                        ARIMAregion_pred_error_matrix[region,1:3]                    =
colMeans(ARIMAreplica_pred_error_matrix, na.rm = TRUE)


                    }


                    ARIMAcrime_pred_error_matrix[crimeRate,1:3]                    =
colMeans(ARIMAregion_pred_error_matrix, na.rm = TRUE)

                    ARIMAcrime_pred_error_matrix_SD[crimeRate,1]                   <-
sd(ARIMAregion_pred_error_matrix[,1], na.rm = TRUE)
                    ARIMAcrime_pred_error_matrix_SD[crimeRate,2]                   <-
sd(ARIMAregion_pred_error_matrix[,2], na.rm = TRUE)
                    ARIMAcrime_pred_error_matrix_SD[crimeRate,3]                   <-
sd(ARIMAregion_pred_error_matrix[,3], na.rm = TRUE)

                    ### Update counters
                    true_dep = true_dep + 1

                    counter.i = counter.i + 1
                    counter.I = counter.I + 1

                    x_1 = x_1 + 1
                    x_2 = x_2 + 1
                    x_3 = x_3 + 1

                    pred_1 = pred_1 + 1
                    pred_2 = pred_2 + 1
                    pred_3 = pred_3 + 1

                }
            }else{
                #print(sum(candidate_vars[region,,crimeRate]))
            }

        } # end region

} # end crime rate

########################################
########################################
```