

# Analysis of the UK Police Dataset report

## Abstract

**OBJECTIVE:** Contemporary research into geography of crime is mainly based on the opportunity-based crime theories. This report takes UK Police Dataset of London region as research target. It aims to explore trends in crime, search hotspots of crime, as well as predict changing patterns of hotspots.

**METHODS:** The author adopts a set of techniques to achieve these goals. These include time curve line, spatial data visualisation, multiple linear regression and classification of machine learning. In addition to UK Police Dataset, “English indices of deprivation 2015” dataset and “English Lower Layer Super Output Areas 2011” dataset is also used.

**RESULTS:** There are two major finding of this report. First, June and September are the peak months of crime and “anti-social behaviour” is the most frequent type of crime. In addition, crime count of “anti-social behaviour” can be predicted by region’s population and income level. Second, the downtown area of London is identified as a crime hotspot and the change of hotspots is mainly influenced by “anti-social behaviour” crime.

**CONCLUSIONS:** The study contributes to our understanding of opportunity-based crime theories and applies computerized geographic information systems to geography of crime. It is unfortunate that, due to time restrictions, the accuracy of regression and classification model is low and it needs to be improved. Based on these findings, further research could be conducted in order to establish predictive policing and crime forecasting.

Word Count: 3051

## Table of Contents

### *Abstract*

<b>1. Introduction and aims .....</b>	<b>1</b>
<b>2. Methodology.....</b>	<b>2</b>
2.1 Trend of crime.....	2
2.2 Analysis of crime types.....	4
2.3 Visualisation of “anti-social behaviour” on map .....	5
2.4 Prediction of crime.....	7
2.5 Classification of hotspot change.....	8
2.6 Summary .....	11
<b>3. Results and discussion.....</b>	<b>12</b>
3.1 The peak month.....	12
3.2 The most frequent crime type.....	13
3.3 Hotspots of “anti-social behaviour” .....	14
3.4 The multiple linear regression model of crime .....	15
3.5 Classification model of hotspot area change.....	17
<b>4. Conclusion .....</b>	<b>19</b>

### *References*

### *R code*

# 1. Introduction and aims

Research into geography of crime has a long history. In the 19th century, Guerry (1833) analysed and plotted the crime map of France grounded on his criminal study on person and property. During the 1910s, an important view, ecological model of urban geography, was developed and supported by the Chicago School of Sociology (Park, Burgess, McKenzie & Wirth, 1925; Shaw & McKay, 1942). During the end of the 20th century, there was another significant finding: opportunity-based crime theories, greatly contribute to the research of geography of crime (Eck & Weisburd, 1995). In their study, they use these theories to investigate the importance of crime place and offer improvements for prevention policies. Today, Christophe and Wim (2017) hold the view that opportunity-based theories are increasingly basis of contemporary studies. Therefore, the topic of this report is based on the theories in the opportunity theories of crime such as the routine activities theory and so forth. It includes trends of crimes, spatial data visualisation, prediction of crime and classification of crime hotspots.

There are four main aims in this report. The first aim is to explore the trend of crime and find frequent crime type. According to the routine activities theory, time is one of the main factors for a crime to occur (Cohen & Felson, 1979). There are thus peak months manifest in each year. The second aim is to find hotspots of crime. As most crimes are distributed in high-crime intensity places. For example, in their study, Steenbeek and Weisburd (2015) established that 4371 crime events occurred on at least one street segment but on many streets, there was little or no crime. The third aim is to predict crime count based on census data of area. The rational choice perspective argues that offenders' decision-making process converge in physical and social environment (Cornish & Clarke, 2008). Therefore, it is possible to build a model to investigate the relationship between number of crimes with attributes of region. The final aim is to build a classification model of hotspot area change. Because of the success of hotspot crime, scholars have begun to investigate where future hotspots are likely to appear based on present crime data (Bowers, Johnson & Pease, 2004). The overall structure of the report takes the form of four sections to present these aims, including introduction and aims, methodology, results and discussion, and conclusion.

## 2. Methodology

This section describes the methodology to achieve four aims and every part follows three stages: gathering data, structuring data, and exploring or visualizing data. In this report, the author chooses crime dataset of London between 2015.12 to 2018.11 from [\[data.police.uk\]](https://data.police.uk) (including Inner London boroughs and the City of London). All analyses were carried out using R Studio, Version 1.1.463.

### 2.1 Trend of crime

The purpose of this section is to plot a time curve of crime data and investigate the time trends such as high-incidence months every year.

#### 2.1.1 Gathering data

For the UK police dataset, the author selects “Data range: December 2015 to November 2018”; “Forces: Metropolitan Police Service and City of London Police”; “Datasets: Include crime data” and generate and download files. It is, however, an established fact that crime data can be separated as different files by month.

Therefore, the author puts all files in one folder ("~/Downloads/Crime/dataset") and attempts to gather them into a new CSV file in order to read them once. First, using function `list.files()` to obtain all files' names. Second, using function `lapply()` to apply function `read.csv()` to all files and merge them by `do.call(rbind, )`. Finally, writing them into a new file and read it as `crime.data`.

#### 2.1.2 Structuring data

Loading package `dplyr`. In order to display the number of crime in every month, the author groups the data by month using `group_by()` and then counts the occurrences by `summarise(Num = n())`. Finally, saves it as `crime.data.monthly`.

#### 2.1.3 Visualizing data

Loading package `ggplot2` and drawing blank chart:

```
ggplot(crime.data.monthly, aes(Month, Num, group = 1)).
```

The author also uses function `geom_line()` to plot the crime count line and function `geom_point()` to add points every month. In addition, function `geom_smooth()` is used to determine patterns of crime.

Moreover, adding a title and changing the axes labels and text format. Therefore, this line chart can show the time trend of crime number and it is easy to identify the peak months.

## 2.2 Analysis of crime types

Based on the time curve, the aim of this section is to compare different crime types in peak months by bar chart and explore the most frequent crime type.

### 2.2.1 Gathering data

The row data is `crime.data` from last section

### 2.2.2 Structuring data

Before analyzing the crime data in peak month, it is necessary to unify month expression. Using function `gsub()` to replace different month number with same month abbreviation, for example:

```
crime.data$Month <- gsub("2016-07", "Jul", crime.data$Month)
crime.data$Month <- gsub("2017-07", "Jul", crime.data$Month)
crime.data$Month <- gsub("2018-07", "Jul", crime.data$Month)
```

After this, peak months data needs to be filtered from raw data by function `filter()` and saved as `crime.data.peak`.

### 2.2.3 Visualizing data

Loading package `ggplot2`. The author uses a bar chart to visualize the distribution of frequency counts by month and breakdown chart based on the crime type. In addition, function `coord_flip()` is used to flip the coordinates and attribute `position="fill"` for display the data as a proportion. Moreover, adding a title, legend and changing the axes labels. Accordingly, this bar chart can clearly compare different crime types and present the most frequent crime type.

## 2.3 Visualisation of “anti-social behaviour” on map

After determining that the most frequent crime type is “anti-social behaviour” crime, the author attempts to visualize it on a map and thus illustrate the hotspots of crime.

### 2.3.1 Gathering data

To download spatial data of London from [\[borders.ukdataservice.ac.uk\]](http://borders.ukdataservice.ac.uk), the author selects “England” and “2011 and later” (then find). In the “Boundaries” box, selecting “English Lower Layer Super Output Areas 2011” and then “List Areas”. In the list of UK cities/counties, the author multiselects areas of Inner London (e.g. Camden, Greenwich, Hackney, Hammersmith and Fulham, Islington, Kensington and Chelsea, Lambeth, Lewisham, Southwark, Tower Hamlets, Wandsworth, Westminster) and the City of London. Finally, “Extract Boundary Data” to download the file (“BoundaryData.zip”).

When reading shapefiles, package `rgdal` is loaded and using function `readOGR()` to read them as `London.shape`. Another step is to obtain “anti-social behaviour” crime data. Using function `filter()` to select “anti-social behaviour” and naming it as `crime.data.antisocial`.

### 2.3.2 Structuring data

When summarizing `crime.data.antisocial`, first loading package `dplyr`. The author then groups the data by LSOA code using function `group_by(LSOA.code)` and counts the occurrences by function `summarise(Num = n())`. Finally, saving the data frame as `crime.antisocial.by.LSOA`. Before plotting map, we need to join data to the shapefiles by the LSOA code using function `left_join()`. In addition, removing missing values by simply assigning it a 0 value.

### 2.3.3 Visualizing data

Loading package `tmap` and using the `tmap_mode()` function set to ‘view’ to make the plot interactive. In addition, the author uses `tm_shape()` to plot map and `tm_fill()` to fill the shape with numbers of crime:

```
tm_shape(london.shape) +  
  tm_fill("Num", alpha = 0.5, style = "kmeans", border.col = "black")
```

Moreover, defining the borders by function `tm_borders()` and adding a scale bar by function `tm_scale_bar()`. Therefore, it is not difficult to find the hotspots of “anti-social behaviour” crime on the map.



## 2.4 Prediction of crime

In order to achieve the aim of prediction, this part builds a multiple linear regression model of crime count with variables of LSOA areas. The variables come from a new dataset: “English indices of deprivation 2015”. It includes the statistics data on every LSOA areas such as income scores, total of population and so forth.

### 2.4.1 Gathering data

First, we need to collect “anti-social behaviour” crime data of hotspot area. Using function `filter()` to screen LSOA area in which number of anti-social crime is more than 95 and saving it as `frequent.antisocial.LSOA.code`. Another stage involves obtaining dataset “English indices of deprivation 2015” from [\[gov.uk\]](http://gov.uk) and then read it as `deprivation2015`.

### 2.4.2 Structuring data

The first step is to select variables columns to work with and save the new data frame as `deprivation2015.variables`. In addition, it is possible to shorten variables names for use, such as `LSOA.deprivation` and `LSOA.employment`. It is then necessary to join anti-social crime data with variables by function `left_join()` and named as `antisocial.with.variables`. Prior to building a model, it is important to remove NA values by function `na.omit()`.

Another task is to find outliers because the number of crimes in hotspot areas are extremely high. In data frame `frequent.antisocial.LSOA.code`, creating a new column of crime type by function `mutate()` and plotting a box plot. Finally, it is necessary to remove outliers by function `filter()`.

### 2.4.3 Exploring data

After structuring data, a model can be constructed using the `lm()` function between numbers of crime with variables including the score of population, income, deprivation, education, employment, health, environment and barriers.

In addition, a function `step()` can be utilised to build another stepwise regression model with choosing backward option. Moreover, it is important to delete several insignificant variables in stepwise regression model to build the final model and see the results of model by function `summary()`.

## 2.5 Classification of hotspot change

Based on the map of “anti-social behaviour”, we can easily find hotspot areas. According to Bowers, Johnson and Pease (2004), however, hotspot areas tend to change through time. Therefore, in this section, the author attempts to build a model to predict the number of crimes in one area increased or not. For example, we aim to predict whether the numbers of crime in “Westminster 018A” area (LSOA.code = E01004734) increase or not in July 2019, based on the crime data in June 2019. The first step is to collect changed crime counts of every LSOA areas in July. The term “True” is used to indicate that amount of crime increased and “FALSE” to indicate that it decreased or unchanged. Second, joining main crime data in June with changing situation of July. Finally, different algorithms can be used to train these data and build a decision tree.

### 2.5.1 Gathering data

The row data is `crime.data` from the first section. And we also need to select and join crime data of “2016-06” and “2016-07”. Then filtering number of crimes more than 100. We also need to check NA values by function `anyNA()`. Next, it is necessary to create a new column to save changing situation by function `mutate()` and naming data frame as `crime.change.16.by.LSOA.100`. The results are as follows:

```
> crime.change.16.by.LSOA.100
# A tibble: 47 x 4
  LSOA.code num.Jun num.Jul crime.increased
  <chr>      <int>  <int> <lgl>
1 E01000010    156    146 FALSE
2 E01000675     94    107  TRUE
3 E01000858     82    111  TRUE
# ... with 45 more rows
```

### 2.5.2 Structuring data

The first step is to summarize crime data in June for every LSOA areas and join with `crime.change.16.by.LSOA.100`. For example, filtering June and one specific LSOA code by `filter(crime.data$Month == "2016-06" & LSOA.code == c("E01004734"))`. It can then be grouped by crime type and summarizing. We then need to interchange columns and rows and also select

useful variables such as “Anti-social behaviour”, “Violence and sexual offer”. Finally, join these data with `crime.change.16.by.LSOA.100`. The results are as follows:

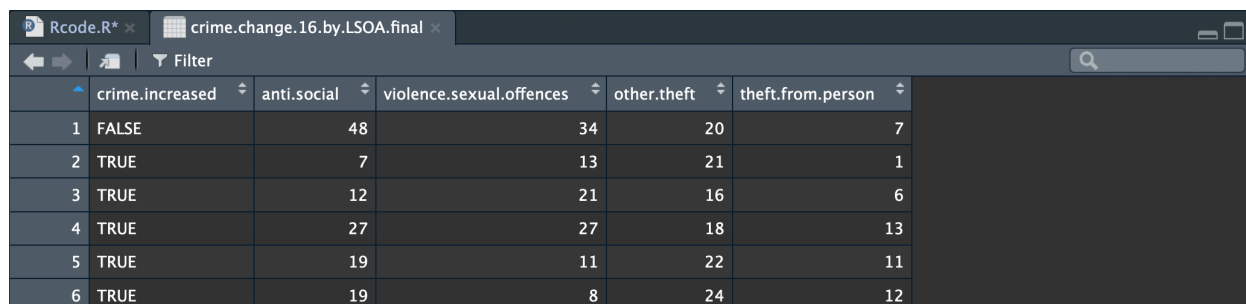


	LSOA.code	num.Jun	num.Jul	crime.increased	Anti-social behaviour	Violence and sexual offences	Other theft	Theft from the person
1	E01004734	447	480	TRUE	117	74	80	57
2	E01000010	156	146	FALSE	NA	NA	NA	NA
3	E01000675	94	107	TRUE	NA	NA	NA	NA
4	E01000858	82	111	TRUE	NA	NA	NA	NA

The Second step is to apply this method to every frequent LSOA area. Before running loop, we need to define LSOA code range. Saving `crime.change.16.by.LSOA.100$LSOA.code` as `LSOA.code.list`. Then, running the loop as followed:

```
for (LSOA.code.each in LSOA.code.list) {
  crime.Jun.16.type.n <- crime.data %>%
    filter(crime.data$Month == "2016-06" & LSOA.code == LSOA.code.each) %>%
    ...
}
```

Therefore, the final data frame is named as `crime.change.16.by.LSOA.final`. We also shorten variables names and it is shown as:



	crime.increased	anti.social	violence.sexual.offences	other.theft	theft.from.person
1	FALSE	48	34	20	7
2	TRUE	7	13	21	1
3	TRUE	12	21	16	6
4	TRUE	27	27	18	13
5	TRUE	19	11	22	11
6	TRUE	19	8	24	12

The third step is to scale the data by package `tidyverse` and function `scale()`. When scaling data, we need to save binary crime condition and add it back at last.

Before we start with applying machine learning, the final step in the process is to create training and test sets. Loading package `caret` and set the seed of the random number generator as 123. It is then possible to use function `createDataPartition()` to split the data into a training set containing 80% of the data

and a test set containing 20% of the data. The training set and test set is named as `crime.train.data` and `crime.test.data`.

### 2.5.3 Exploring data

When training data, the author uses the `caret` package to run different ML algorithms. First, it is necessary to set up a testing environment comprising of using 10-fold (`number=10`) cross validation `method="cv"` and measuring performance using *accuracy*. Models can then be constructed using each of the ML algorithms including *nb*, *lad*, *cart*, *knn*, *svm* and *rf*. The function `resample()` can then be used to compare the performance of the various algorithms and view the results by function `summary()`. In addition, we also use test dataset to test model and obtain the accuracy of classification by function `confusionMatrix()`. Finally, package `rpart` and `rattle` can be used to plot a decision tree.

## 2.6 Summary

The table below summarizes the methodology including used techniques, list of datasets used, used variables in the datasets and used R packages.

Table 1 Methodology

Technique	Dataset	Variable	R Package
Multiple linear regression	English indices of deprivation 2015 <a href="http://gov.uk">[gov.uk]</a>	LSOA.code..2011. LSOA.name..2011. Index.of.Multiple.Deprivation..IMD..Score Income.Score..rate. Employment.Score..rate. Education..Skills.and.Training.Score Health.Deprivation.and.Disability.Score Barriers.to.Housing.and.Services.Score Living.Environment.Score Total.population..mid.2012..excluding.prisoners.	'tidyverse'
Time curve line	UK Police Data [from Nov 2015 to Dec 2018] (Metropolitan Police Service and City of London Police) <a href="http://data.police.uk">[data.police.uk]</a>	Month	'dplyr' 'ggplot2'
Multi-category bar chart		Month Crime.type	'ggplot2'
ML-Classification		LSOA.code Month Crime.type	'dplyr' 'caret' 'tidyverse' 'rpart' 'rattle'
Spatial data visualisation		Crime.type LSOA.code england_lsoa_2011	'rgdal' 'dplyr' 'tmap'
	English Lower Layer Super Output Areas 2011 (Inner London boroughs and the City of London) <a href="http://borders.ukdataservice.ac.uk">[borders.ukdataservice.ac.uk]</a>		

(Excluding word count)

## 3. Results and discussion

### 3.1 The peak month

The first aim of this report is to investigate the crime trends over time. As shown in Figure 1, the number of crimes per month increases with some fluctuation, from about 86000 to 94000, during the last three years. In addition, it is clear that June and September are apparently more than other months. This finding further supports the routine activities theory that time is an important factor for crime (Cohen & Felson, 1979). Therefore, it is argued that in London area, numbers of crimes are rising month by month and the high-incidence months are June and September.

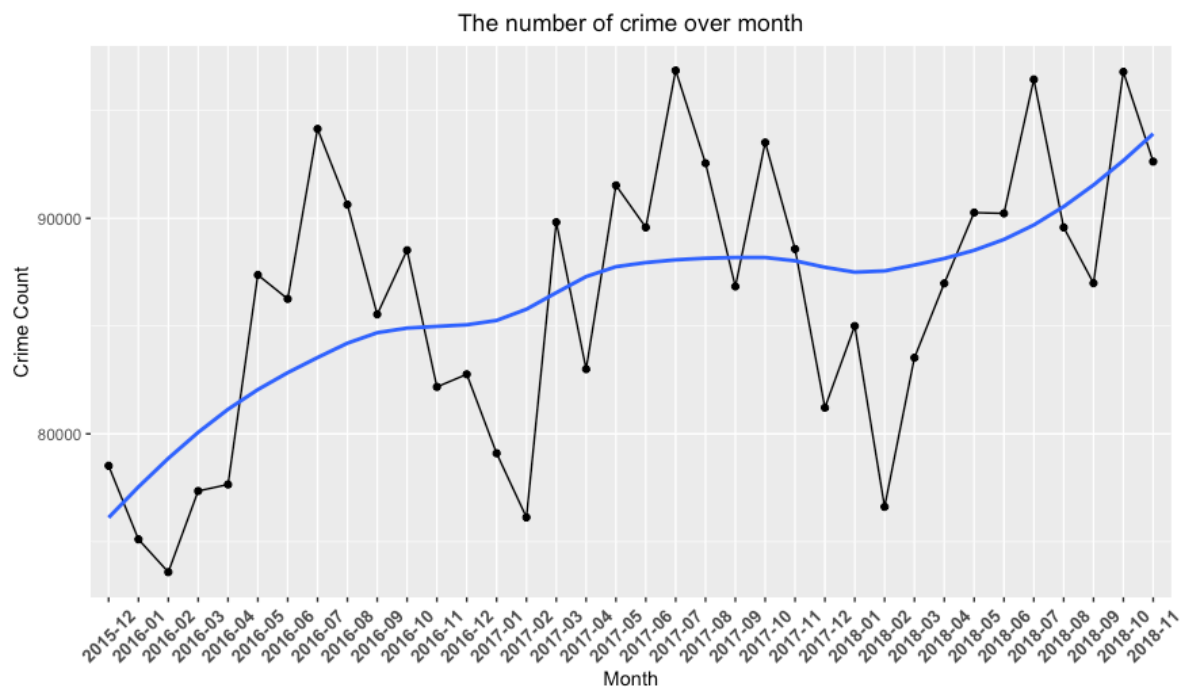


Figure 1

## 3.2 The most frequent crime type

Based on the results above, the author cleans and plots the crime counts of various crime type in peak month (i.e. June and September). Figure 2 compares the different crime types in peak months. It can be seen that “Anti-social behaviour”, “Violence and sexual offences” and “Other theft” are top three crime types. In addition, “Anti-social behaviour” is the most crime type and approximately accounts for 25%. A possible explanation for this might be that “Anti-social behaviour” covers a wide range of unacceptable activity, such as Street drinking and Fireworks misuse (UK Police). It is thus claimed that “Anti-social behaviour” is the most frequent crime type in London.

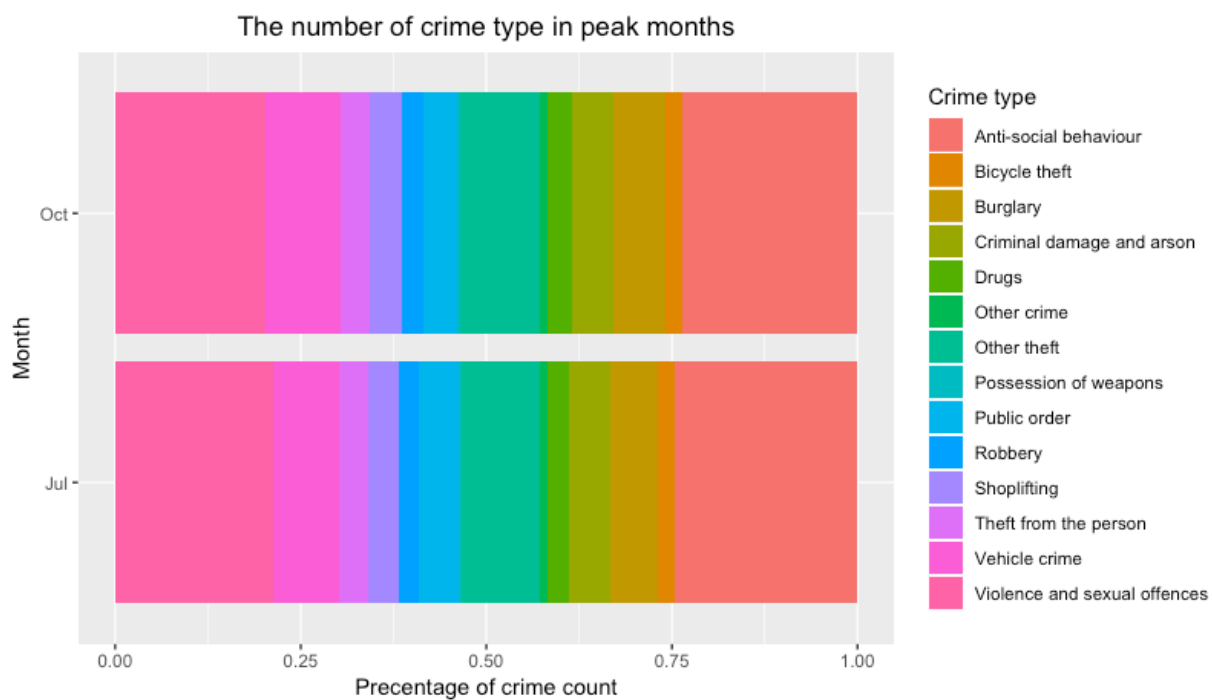


Figure 2

### 3.3 Hotspots of “anti-social behaviour”

As discussed in the “Introduction and aims” section, most crimes are distributed in high-crime intensity places. Accordingly, the aim of this part is to plot the crime map of “anti-social behaviour” and find the hotspot areas. From Figure 3 below, we can see that most “anti-social behaviour” occurs in the city centre and some LSOA areas are extremely higher than others. This result is in line with previous study. There is clear evidence to suggest that downtown of London is hotspot area.

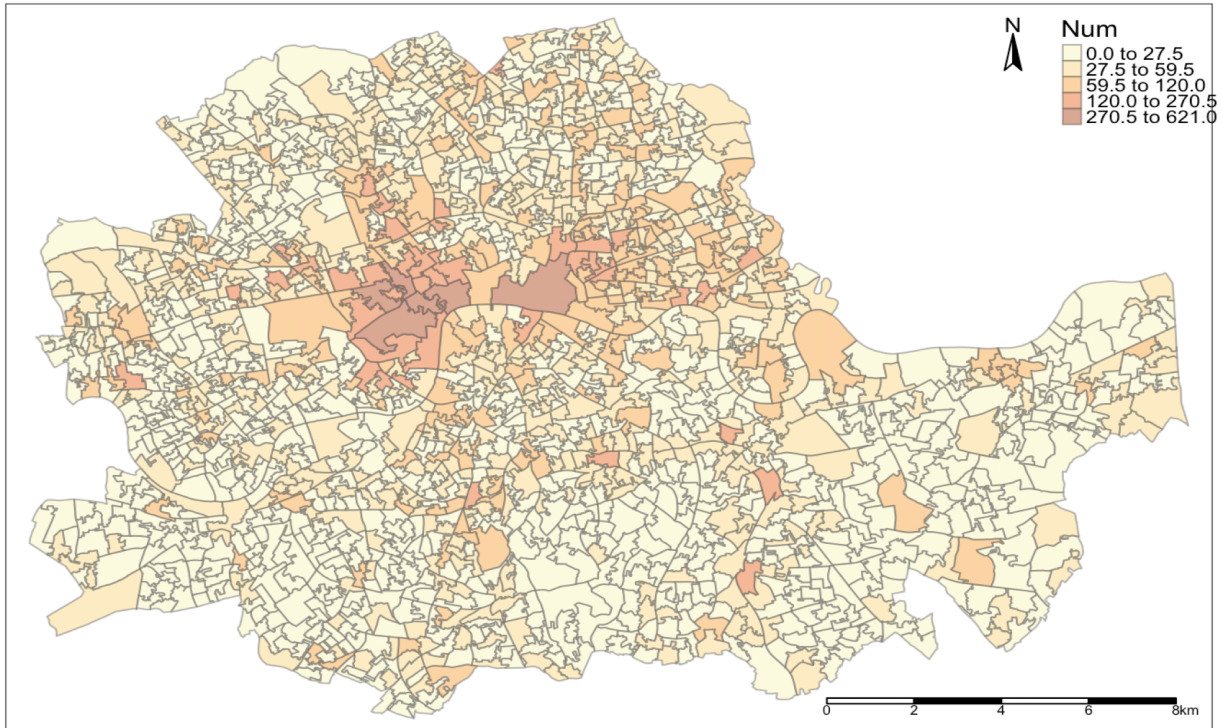


Figure 3



### 3.4 The multiple linear regression model of crime

The aim here is to predict crime based on the attributes of region. When finding outliers, it can be seen from the box plot that the crime count of some region are extremely high. Therefore, we exclude these regions, whose crime counts are more than 300, in order to improve our model.

The results of first regression model is displayed in Table 2.

**Table 2 First regression on crime count with regional attributes, June 2016**

	Intercept	Population	Income	Deprivation	Employment	Education	Health	Barriers	Environment
Estimate	85.51	0.02871	-183	2.813	-132.5	-0.5258	-14.26	-0.995	0.02212
Std. Error	25.15	0.007341	145	1.408	178.2	0.5063	8.077	0.6103	0.3134
t value	3.400	3.911	-1.262	1.998	-0.743	-1.039	-1.766	-1.631	0.071
Pr	0.00089***	0.000148***	0.20908	0.047833*	0.458634	0.300935	0.07979	0.10536	0.943842

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 34.02 on 129 degrees of freedom

Multiple R-squared: 0.1954

Adjusted R-squared: 0.1455

F-statistic: 3.916 on 8 and 129 DF, p-value: 0.0003609

It can be seen that the F-statistic p-value is <0.05 but the R-squared is 0.1455 which is bad. In addition, there are many of the coefficients are  $p > 0.05$  and could potentially be eliminated from the model. Therefore, we attempt to improve our model by stepwise regression with backward option. The result of reduced regression model is displayed in Table 3.

**Table 3 Stepwise regression on crime count with regional attributes, June 2016**

	Intercept	Population	Income	Deprivation	Health	Barriers
Estimate	89.61	0.02969	-276.8	2.333	-10.64	-0.8906
Std. Error	17.58	0.007131	97.05	0.9864	7.131	0.5502
t value	5.097	4.164	-2.852	2.366	-1.492	-1.619
Pr	0.00000117***	0.0000561***	0.00505**	0.01946*	0.13819	0.10791

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 33.87 on 132 degrees of freedom

Multiple R-squared: 0.1836, Adjusted R-squared: 0.1527

F-statistic: 5.939 on 5 and 132 DF, p-value: 5.498e-05

It can, however, be seen that there are still some variables which are not significant. It is then necessary to delete them for another regression model and compare models.

**Table 4 Final regression on crime count with regional attributes, June 2016**

	Intercept	Population	Income
Estimate	100.72014	0.02779	-106.91330
Std. Error	14.99663	0.00681	35.48312
t value	6.716	4.080	-3.013
Pr	4.75 x 10 <sup>-10***</sup>	7.67 x 10 <sup>-5***</sup>	0.00309**

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 34.24 on 135 degrees of freedom

Multiple R-squared: 0.1467, Adjusted R-squared: 0.1341

F-statistic: 11.61 on 2 and 135 DF, p-value: 2.23e-05

**Table 5 Analysis of Variance Table**

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	135	158295				
2	132	151445	3	6849.1	1.9899	0.1186

Model 1: Num ~ LSOA.population + LSOA.income

Model 2: Num ~ LSOA.population + LSOA.income + LSOA.deprivation + LSOA.health + LSOA.barriers

As shown in Table 5, because of  $F=2.82$  ( $p\text{-value}=0.0647$ ), it appears that the variables deprivation, health and barriers do not contribute significant information to the crime count once the variables population and income have been taken into consideration.

The final output Table 4 shows that  $F = 11.61$  ( $p=2.23e-05$ ), indicating that we should clearly reject the null hypothesis that the variable population and income collectively have no effect on crime. The results also show that the variable population and income are significant. In addition, the output also shows that  $R^2$  adjusted = 0.1527, which suggests the model can only explain 15.27% of the data. To summarize, it is argued that crime count is significantly influenced by region's population and income with income having a huge negative effect. The regression model is: Crime count =  $100.72014 + 0.02779 \times \text{population} - 106.91330 \times \text{income}$ .

### 3.5 Classification model of hotspot area change

When building a classification model, it is important to compare various algorithms. The accuracy of model is shown in Table 6. It can be seen that based on the mean accuracy score, the k nearest neighbours (knn) model seems to work best on this data. In addition, when testing test dataset, it can be seen from Table 7 that the accuracy of classification is 0.7778.

**Table 6 Accuracy of different models**

	nb	lda	knn	svm	rf
Min	0.50	0.50	0.75	0.50	0.25
1st Qu.	0.6875	0.7500	0.7500	0.7500	0.7500
Median	0.75	0.75	1.00	0.75	1.00
Mean	0.7417	0.7167	0.9000	0.7750	0.8417
3rd Qu.	0.75	0.75	1.00	0.75	1.00
Max	1.00	0.75	1.00	1.00	1.00

**Table 7 Accuracy of classification**

Prediction	Reference	
	False	True
False	0.50	0.50
True	0.6875	0.7500

Accuracy: 0.7778

Another finding concerns the function of a decision tree. It can show the different rules that can be used to predict the change of crime count in hotspot area, and the “anti-social behaviour” variable makes the biggest effect on the change of crime count. It can be seen from Figure 4, in one hotspot area, if the “anti-social behaviour”  $< -0.3342$  and the “violence and sexual offences”  $< 0.1763$  then the crime count will decrease, this area could not be hotspot next month.



Figure 4

## 4. Conclusion

By taking London region as research target, the aim of the present research was to investigate the trend of crime, analyse crime type, plot crime map, predict crime and explore hotspot change. This study has shown that June and September are the peak months of crime and “anti-social behaviour” is the most frequent crime type. In addition, crime count of “anti-social behaviour” can be predicted by region’s population and income level. The second major finding is that downtown area of London is crime hotspots and the change of hotspots is mainly influenced by “anti-social behaviour” crime. The present study provides the comprehensive analysis of crime dataset of London. The main limitation of this study is due to word restrictions. Given more time and space, moreover, there is considerable potential to improve the regression and classification models. Based on crime prediction and hotspot change, further work should be undertaken to explore predictive policing and crime forecasting model.

## References

- Bowers, K. J., Johnson, S. D., & Pease, K. (2004). Prospective Hot-Spotting: The Future of Crime Mapping? *British Journal of Criminology*, 44(5), 641-658. <http://dx.doi.org/10.1093/bjc/azh036>
- Christophe, V., & Wim, B. (2017). The geography of crime and crime control. *Applied Geography*, 86, 220-225. <https://doi.org/10.1016/j.apgeog.2017.08.012>
- Cohen, L. E., & Felson, M. (1979). Social change and crime rate trends: A routine activity approach. *American Sociological Review*, 44(4), 588-608. <http://dx.doi.org/10.2307/2094589>
- Cornish, D. B., & Clarke, R. V. (2008). The rational choice perspective. In R. Wortley, & L. Mazerolle (Eds.), *Environmental criminology and crime analysis* (pp. 21-47). Cullompton: Willan publishing.
- Eck, J. E., & Weisburd, D. L. (1995). Crime places in crime theory. In J. E. Eck, & D. L. Weisburd (Eds.), *Crime and place*. Monsey, New York: Criminal Justice Press.
- Guerrey, A. M. (1833). *Essai sur la Statistique Morale de la France*. Paris: Crochard.
- Park, R., Burgess, E., McKenzie, R., & Wirth, L. (1925). *The city*. Chicago: University of Chicago Press.
- Steenbeek, W., & Weisburd, D. L. (2015). Where the action is in Crime? An examination of variability of crime across different spatial units in the Hague, 2001-2009. *Journal of Quantitative Criminology*, 1-21. <http://dx.doi.org/10.1007/s10940-015-9276-3>

## R code

```
# *****#
# Trend of crime
# *****#

# 1.Gathering data
setwd("~/Downloads/Crime/dataset")
fileNames <- list.files(pattern = "*.csv") # get the files names
readFiles <- lapply(fileNames, # apply read.csv
                    function(x)
                      read.csv(x, stringsAsFactors = F, header = T))
fileBind <- do.call(rbind, readFiles) # rbind them
write.csv(fileBind, "crimedata_2015-12_2018-11.csv")

crime.data <- read.csv("crimedata_2015-12_2018-11.csv",
                      header = TRUE,
                      stringsAsFactors = FALSE)
crime.data <- crime.data[, -1] # delete sequence number
View(crime.data)

# 2.Structuring data

library(dplyr)
crime.data.monthly <- crime.data %>% # count by month
  group_by(Month) %>%
  summarise(Num = n())
View(crime.data.monthly)

# 3.Visualizing data
library(ggplot2)
ggplot(crime.data.monthly, aes(Month, Num, group = 1)) +
  geom_line() +
  geom_point() +
```

```
geom_smooth(se = FALSE) +  
  labs(x = "Month", y = "Crime Count", title = "The number of crime over  
month") +  
  theme(axis.text.x = element_text( # change text format  
    size = 10, face = "bold", vjust = 0.5, hjust = 0.5, angle = 45)) +  
  theme(plot.title = element_text(hjust = 0.5)) # center the title
```



```

# *****#
# Analysis of crime types
# *****#

# 1.Structuring data
crime.data$Month <- gsub("2016-07", "Jul", crime.data$Month)
crime.data$Month <- gsub("2017-07", "Jul", crime.data$Month)
crime.data$Month <- gsub("2018-07", "Jul", crime.data$Month)
crime.data$Month <- gsub("2016-10", "Oct", crime.data$Month)
crime.data$Month <- gsub("2017-10", "Oct", crime.data$Month)
crime.data$Month <- gsub("2018-10", "Oct", crime.data$Month)
crime.data.peak <- filter(crime.data, (Month %in% c("Jul", "Oct")))
View(crime.data.peak)

# 2.Visualising data
ggplot(data=crime.data.peak) +
  geom_bar(mapping=aes(x = Month, fill=Crime.type), position="fill") +
  coord_flip() +
  labs(fill = "Crime type") +
  ylab("Precentage of crime count") +
  ggtitle("The number of crime type in peak months") +
  theme(plot.title = element_text(hjust = 0.5)) # center the title

# *****#
# Visualisation of “anti-social behaviour” on map
# *****#

# 1.Gathering data
library(rgdal)
london.shape <- readOGR(dsn = "~/Downloads/Crime/BoundaryData",
                        layer = "england_lsoa_2011")

```

```
crime.data.antisocial <- filter(crime.data.peak,  
                                Crime.type == "Anti-social behaviour")  
View(crime.data.antisocial)
```

# 2.Structuring data

```
library(dplyr)  
crime.antisocial.by.LSOA <- crime.data.antisocial %>%  
  group_by(LSOA.code) %>% # count by LSOA code  
  summarise(Num = n())  
View(crime.antisocial.by.LSOA)
```

```
london.shape@data <- left_join(london.shape@data,  
                                crime.antisocial.by.LSOA,  
                                by = c('code' = 'LSOA.code'))  
london.shape[is.na(london.shape@data$Num)] <- 0 # remove missing data  
View(london.shape@data)
```

# 3.Visualising data

```
library(tmap)  
tmap_mode("plot")  
tm_shape(london.shape) +  
  tm_fill("Num", alpha = 0.5, style = "kmeans", border.col = "black") +  
  tm_borders(alpha = 0.5) +  
  tm_compass(position=c("right", "top")) +  
  tm_scale_bar() # add a scale bar
```

```

# *****#
# Prediction of crime
# *****#

# 1.Gathering data
frequent.antisocial.LSOA.code <- filter(crime.antisocial.by.LSOA, Num >= 95)
View(frequent.antisocial.LSOA.code)

setwd("~/Downloads/Crime")
deprivation2015 <- read.csv("Deprivation2015.csv")
View(deprivation2015)

# 2.Structuring data
deprivation2015.variables <- deprivation2015 %>%
  select(LSOA.code..2011., LSOA.name..2011.,
         Index.of.Multiple.Deprivation..IMD..Score,
         Income.Score..rate., Employment.Score..rate.,
         Education..Skills.and.Training.Score,
         Health.Deprivation.and.Disability.Score,
         Barriers.to.Housing.and.Services.Score,
         Living.Environment.Score,
         Total.population..mid.2012..excluding.prisoners.)

names(deprivation2015.variables)[names(deprivation2015.variables)=="LSOA.code
..2011."]<-"LSOA.code"
names(deprivation2015.variables)[names(deprivation2015.variables)=="LSOA.name
..2011."]<-"LSOA.name"
names(deprivation2015.variables)[names(deprivation2015.variables)=="Index.of.
Multiple.Deprivation..IMD..Score"]<-"LSOA.deprivation"
names(deprivation2015.variables)[names(deprivation2015.variables)=="Income.Sc
ore..rate."]<-"LSOA.income"

```

```

names(deprivation2015.variables)[names(deprivation2015.variables)=="Employment.Score..rate."]<-"LSOA.employment"
names(deprivation2015.variables)[names(deprivation2015.variables)=="Education..Skills.and.Training.Score"]<-"LSOA.edu"
names(deprivation2015.variables)[names(deprivation2015.variables)=="Health.Deprivation.and.Disability.Score"]<-"LSOA.health"
names(deprivation2015.variables)[names(deprivation2015.variables)=="Barriers.to.Housing.and.Services.Score"]<-"LSOA.barriers"
names(deprivation2015.variables)[names(deprivation2015.variables)=="Living.Environment.Score"]<-"LSOA.enviroment"
names(deprivation2015.variables)[names(deprivation2015.variables)=="Total.population..mid.2012..excluding.prisoners."]<-"LSOA.population"
View(deprivation2015.variables)

```

```

antisocial.with.variables <- left_join(deprivation2015.variables,
                                       frequent.antisocial.LSOA.code,
                                       by = "LSOA.code")
anyNA(antisocial.with.variables) # if data contain NA values
is.na(antisocial.with.variables) # find out NA values
antisocial.with.variables <- na.omit(antisocial.with.variables) # remove NA values
View(antisocial.with.variables)

```

```

antisocial.LSOA.outlier <- frequent.antisocial.LSOA.code %>%
  mutate(Crime.type = "Antisocial")
View(antisocial.LSOA.outlier)
ggplot(antisocial.LSOA.outlier, aes(Crime.type, Num)) + # find outliers by boxplot
  geom_boxplot() +
  labs(x = "Crime type", y = "Crime Count", title = "The boxplot of crime count every region") +
  theme(plot.title = element_text(hjust = 0.5)) # center the title

```

```
antisocial.with.variables <- antisocial.with.variables %>% # remove outlier
  filter(antisocial.with.variables$Num <= 300)
View(antisocial.with.variables)
```

```
# 3.Exploring data
```

```
antisocial.model <- lm(
  formula = Num ~ LSOA.population + LSOA.income + LSOA.deprivation +
  LSOA.employment + LSOA.edu + LSOA.health + LSOA.barriers + LSOA.enviroment,
  data = antisocial.with.variables)
summary(antisocial.model)
```

```
par(mfrow=c(2,2))
plot(antisocial.model)
```

```
antisocial.model.reduced <- step( # stepwise regression
  antisocial.model, direction = "backward")
summary(antisocial.model.reduced)
```

```
antisocial.model.final <- lm(
  formula = Num ~ LSOA.population + LSOA.income,
  data = antisocial.with.variables)
summary(antisocial.model.final)
```

```
anova(antisocial.model.final,antisocial.model.reduced)
```

```

# *****#
# Classification of hotspot change
# *****#

# 1.Gathering data

library(dplyr)
crime.Jun.16.by.LSOA <- crime.data %>%
  filter(crime.data$Month == "2016-06") %>%
  group_by(LSOA.code) %>%
  summarise(num.Jun = n())
View(crime.Jun.16.by.LSOA)

crime.Jul.16.by.LSOA <- crime.data %>%
  filter(crime.data$Month == "2016-07") %>%
  group_by(LSOA.code) %>%
  summarise(num.Jul = n())
View(crime.Jul.16.by.LSOA)

crime.change.16.by.LSOA <- crime.Jun.16.by.LSOA %>%
  left_join(crime.Jul.16.by.LSOA, by = "LSOA.code")
crime.change.16.by.LSOA.100 <- filter(crime.change.16.by.LSOA,
                                     crime.change.16.by.LSOA$num.Jul >= 100)
View(crime.change.16.by.LSOA.100)

anyNA(crime.change.16.by.LSOA.100) # if data contain NA values
crime.change.16.by.LSOA.100 <- crime.change.16.by.LSOA.100[-1, ] # remove
NULL value
crime.change.16.by.LSOA.100 <- mutate(
  crime.change.16.by.LSOA.100,
  crime.increased = num.Jul > num.Jun)
View(crime.change.16.by.LSOA.100)

```

```

# 2.Structuring data
# test for one area
crime.Jun.16.type.n.F <- crime.data %>%
  filter(crime.data$Month == "2016-06" & LSOA.code == c("E01004734")) %>%
  group_by(Crime.type) %>%
  summarise(n.type.Jun.16 = n())
View(crime.Jun.16.type.n.F)

LSOA.code <- c("E01004734") # interchange columns and rows
data.frame(crime.Jun.16.type.n.F, row.names=1)
t1 <- t(data.frame(crime.Jun.16.type.n.F,row.names=1))
t2 <- as.data.frame(t1,row.names=F)
t3.F <- as.data.frame(cbind(LSOA.code,t2))
View(t3.F)
t3.F <- select(t3.F, LSOA.code, `Anti-social behaviour`, `Violence and sexual
offences`, `Other theft`, `Theft from the person`)

crime.change.16.by.LSOA.test.F <- crime.change.16.by.LSOA.100 %>%
  left_join(t3.F, by = "LSOA.code")
View(crime.change.16.by.LSOA.test.F)

# apply to every LSOA areas
LSOA.code.list <- crime.change.16.by.LSOA.100$LSOA.code
View(LSOA.code.list)

for (LSOA.code.each in LSOA.code.list) {
  crime.Jun.16.type.n <- crime.data %>%
    filter(crime.data$Month == "2016-06" & LSOA.code == LSOA.code.each) %>%
    group_by(Crime.type) %>%
    summarise(n.type.Jun.16 = n())
  LSOA.code <- LSOA.code.each
  data.frame(crime.Jun.16.type.n, row.names=1)
  t1 <- t(data.frame(crime.Jun.16.type.n,row.names=1))

```

```

t2 <- as.data.frame(t1,row.names=F)
t3 <- as.data.frame(cbind(LSOA.code,t2))
t3 <- select(t3, LSOA.code, `Anti-social behaviour`, `Violence and sexual
offences`, `Other theft`, `Theft from the person`)
t3.F <- rbind(t3, t3.F)
View(t3.F)
}

crime.change.16.by.LSOA.final <- crime.change.16.by.LSOA.100 %>%
  left_join(t3.F, by = "LSOA.code")
crime.change.16.by.LSOA.final <- crime.change.16.by.LSOA.final[, c(-1, -2, -
3)]

names(crime.change.16.by.LSOA.final)[names(crime.change.16.by.LSOA.final)=="A
nti-social behaviour"]<-"anti.social"
names(crime.change.16.by.LSOA.final)[names(crime.change.16.by.LSOA.final)=="V
iolence and sexual offences"]<-"violence.sexual.offences"
names(crime.change.16.by.LSOA.final)[names(crime.change.16.by.LSOA.final)=="O
ther theft"]<-"other.theft"
names(crime.change.16.by.LSOA.final)[names(crime.change.16.by.LSOA.final)=="T
heft from the person"]<-"theft.from.person"

View(crime.change.16.by.LSOA.final)

# scaling the data
library(tidyverse)
crime.change.16.by.LSOA.final$crime.increased <-
as.factor(crime.change.16.by.LSOA.final$crime.increased)
x <- crime.change.16.by.LSOA.final %>% select(crime.increased) # save binary
crime condition
crime.change.16.by.LSOA.final <-
  scale(crime.change.16.by.LSOA.final[, -1]) # only apply to numeric

```



```

crime.change.16.by.LSOA.final <- cbind(crime.change.16.by.LSOA.final, x) #
add condition back

# Split data into training and test sets
library(caret)
anyNA(crime.change.16.by.LSOA.final) # check NA
set.seed(123)
crime.data.split <- crime.change.16.by.LSOA.final$crime.increased %>%
  createDataPartition(p = 0.8, list = FALSE)
crime.train.data <- crime.change.16.by.LSOA.final[crime.data.split, ]
crime.test.data <- crime.change.16.by.LSOA.final[-crime.data.split, ]

# 3.Exploring data

# perform classification
library(caret)
control <- trainControl(method = "cv", number = 10) # using 10-fold cross
validation
metric <- "Accuracy" # measure by accuracy

nb.model <- train(crime.increased~., data = crime.train.data, method = "nb",
metric = metric, trControl = control)
lda.model <- train(crime.increased~., data = crime.train.data, method =
"lda", metric = metric, trControl = control)
knn.model <- train(crime.increased~., data=crime.train.data, method="knn",
metric=metric, trControl=control)
svm.model <- train(crime.increased~., data=crime.train.data,
method="svmRadial", metric=metric, trControl=control)
rf.model <- train(crime.increased~., data=crime.train.data, method="rf",
metric=metric, trControl=control)

results <- resamples(list(nb = nb.model,
                        lda = lda.model,

```

```

        knn = knn.model,
        svm = svm.model,
        rf = rf.model)) # compare each results
summary(results) # k nearest neighbour (knn) has maxmium mean accuracy score

knn.predictions <- predict(knn.model, crime.test.data) # using test data
confusionMatrix(knn.predictions,
                 crime.test.data$crime.increased,
                 positive = "TRUE") # compare results

# decision tree
library(rpart)
set.seed(123)
rpart.model <- rpart(crime.increased~.,
                     data = crime.train.data,
                     method = "class") # simply build decision tree

par(xpd = NA) # prevent the text in plot overlapped
plot(rpart.model, main = "Classification Tree for crime (trainging data)")
text(rpart.model, digits = 2, cex = 0.6) # add text

```