Effect of Brexit on crime rates in the UK

Janos Strasszer

2017 January

Contents

1. Synopsis

In 2016 there was a significant event in the history of the European Union. One of the member states - The United Kingdom - held a referendum to decide whether leave the union or stay. There were several articles [1][2] and Facebook posts before, during and after the Brexit voting that reported violent acts and hate crimes against foreigners and emigrants in the country. These reports suggested that these acts happened mainly in pro-brexit territories.

Did hate crimes happen during Brexit voting in a volume that is concerning? Was this really true or just some kind of political sensationalism?

A statistical analysis of the crime reports from United Kingdom's police forces can even prove or confute this.

Conclusion: According to the following analysis and regression there is a connection. Pro-brexit regions encountered with higher crime rates in the months of the brexit referendum poll.

Note: To reproduce the result please visit the public github repository of this analysis: https://github.com/entelente/CEU-DA2-Term_Project

[1] The Guardian - Racist incidents feared to be linked to Brexit result (https://www.theguardian.com/politics/2016/jun/26/racist-incidents-feared-to-be-linked-to-brexit-result-reported-in-england-and-wales)

[2] Independent - Brexit: Wave of hate crime and racial abuse reported following EU referendum (http://www.independent.co.uk/news/uk/home-news/brexit-eu-referendum-racial-racism-abuse-hate-crime-reported-latest-leave-immigration-a7104191.html)

2. Understanding and processing the data

Brexit referendum data

EU referendum result contains the outcome of the poll for every governmental region of the UK. For further details about the data cleaning and the data itself please check Appendix A. and Appendix B.

Variables

Column	Format	Description
Region	Text	Big region of UK. E.g. "East"
Area	Text	The governmental region where the poll occurred
Votes_Cast	Numeric	Total number of valid votes
Remain	Numeric	No. votes against exit.
Leave	Numeric	No. votes for brexit.

Number of pro-brexit regions is 0 of 382 total regions.

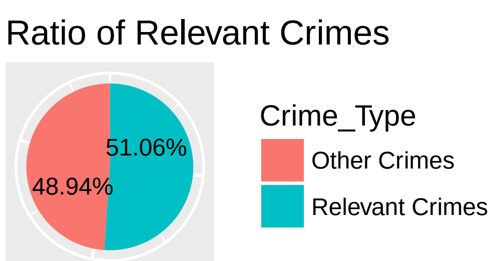
Crime data from UK Police forces

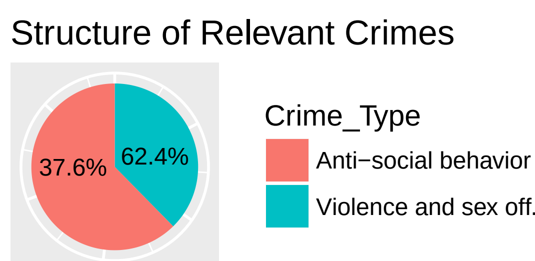
Crime data from UK Police forces contains every crime that occurred in the UK (except Scotland) from January 2015 to November 2016. For further details about the data cleaning and the data itself please check Appendix A. and Appendix B.

Variables

Column	Format	Description
Territory	Text	Identifies the police force where the crime occurred
Longitude	Numeric	Geographic coordinate of the crime scene
Latitude	Numeric	Geographic coordinate of the crime scene
Type	Text	The type of the crime
Month	Numeric	The month when the crime occurred
Year	Numeric	The year when the crime occurred

There are 11306179 observations of crimes. After filtering out non-relevant crimes keeping only **Anti-social behaviour** and **Violence and sexual offences** there are 5773121 crimes left.

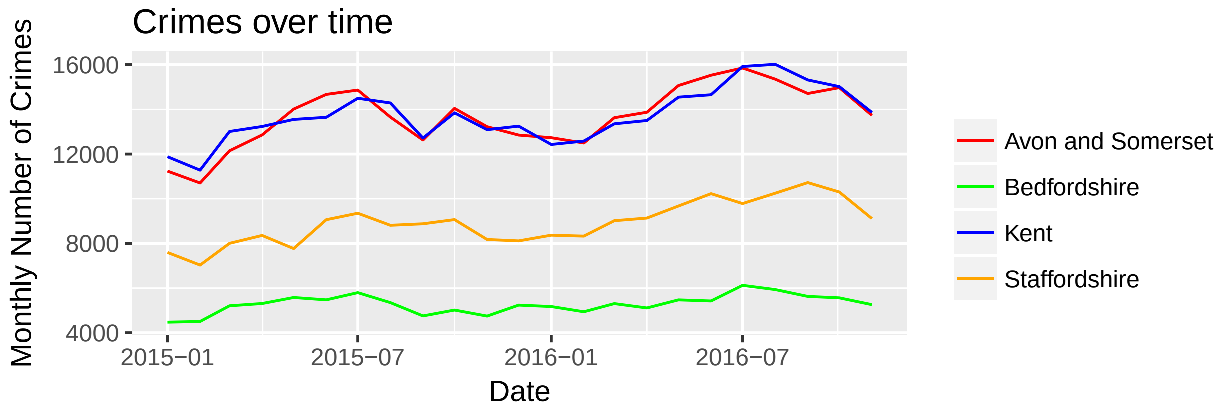




Aggregating crimes

In current analysis we take care only the monthly crime statistics not the individual crimes thus it has to be aggregated.

For demonstration purposes lets select a few police forces "Avon and Somerset Constabulary, Bedfordshire Police, Kent Police, Staffordshire Police" and visualize the number of crimes (Antisocial behavior and Violent) over time.



Looking at this plot it is clearly evident that the crime rates topped around the time of brexit referendum (2016-06) but it can be a coincident or a trend too. Further analysis could help to resolve this.

Cross referencing with vote data

Crime and brexit poll outcome data have be cross referenced wit the help of the xref data. (Please check Appendix A for further details). This xref data can connect Government regions that present in the vote dataset with police force regions those are in the crime dataset.

The problem is that a police region can consist multiple areas. There are two possible solutions for this:

	Solution
A	Use the same crime data for all the areas in the same police region
B	Aggregate the votes for each police regions

Solution A could jeopardize the the analysis because it is possible that a police region has one are with very low crime rates and an other area with large crime rates. Applying the same statistics for both region would be a mistake. Solution B is much better because there we will use valid vote counts but withing larger regions. The best solution would be to have more detailed crime data for each governmental areas. (Actually we have the data because a dataset with the geographical borders of each region is downloadable from the internet and every crime has a GPS coordinate pair, but this is out of the scope of this analysis) Please check Appendix B for details of the cross-referenced data. A binary variable Pro_brexit has to be calculated for each Police Force Areas to indicate whether the region is pro- or counter-brexit.

After the cross referencing the dataset has the following properties:

Variables:

Column	Format	Description
Territory	Text	Identifies the police force where the crime occurred
Year	Numeric	The year when the crime occurred
Month	Numeric	The month when the crime occurred

Column	Format	Description
Count	Numeric	The number of crimes
Votes_Cast	Numeric	Total number of valid votes
Pro_brexit	Binary	Indicates whether the region is pro-brexit

Statistics: In the dataset there are 23 months of data from 44 Police Force Areas. From those there are 828 pro-brexit areas.

3. Regression Analysis : Pro-brexit vs. Counter-brexit

In the first regression we will examine the effect of brexit by comparing the number of crimes in each regions in 2015 and 2016 using the aggregated crime statistics from May, June, July (This time period will cover the time frame of the Brexit hysteria). Using the Pro_brexit binary variable as the independent variable in the regression it is easy to find a correlation or the lack of correlation.

Regression A:

$$E[Delta\_crimes|Pro\_brexit] = \alpha + \beta * Pro\_brexit$$

There can be created a second regression with a control variable of total number of valid votes. This is representing the size of the region which can be also a serious player in the regression.

Regression B:

$$E[Delta\_crimes|Pro\_brexit, Votes\_Cast] = \alpha + \beta * Pro\_brexit + \gamma * Votes\_Cast$$

To do these regressions the following steps were made:

- Data filtering, only data from May, June and July was kept
- Delta_crimes variable was calculated by subtracting crime count in 2015 from crime count in 2016.

Interpreting results

The first regression didn't provide an usable result because of the huge standard error the slope coefficient is not statistically significant, interpreting the coefficients has no use. Fortunately the second regression worked better and provided statistically significant results.

Intercept: In the examined 3 months time period the average number of crimes is lower by 2822.76 in average in 2016 than in 2015 in regions where 0 votes were cast. This don't have any human interpret-able meaning it just provides an offset for the result. The confidence interval is [-4,713.48, -932.03] and the intercept is in this interval with 95% probability.

Slope coefficiant of Pro_brexit: In the examined 3 months time period the average number of crimes is higher by 1691.3 (in 2016 compared to 2015) in pro-brexit regions compared to counter-brexit regions with the same number of votes cast in average. The confidence interval is [34.16, 3,348.45] and the coefficient is in this interval with 95% probability. The p-value is 0.057 which is the chance of the null hypothesis (the slope to be zero) to be zero. I will reject the null hypothesis because there is a 94.3% probability of the positive correlation.

Slope coefficiant of Votes_Cast: In the examined 3 months time period the average number of crimes is higher by 0.004 (in 2016 compared to 2015) in average for each valid vote comparing counter-brexit regions. The confidence interval is [0.003, 0.01] and the coefficient is in this interval with 95% probability. The p-value is lesser than 0.01 which is the chance of the null hypothesis (the slope to be zero) to be zero. I will reject the null hypothesis and accept the positive correlation.

Table 5: Comparing regressions

	Dependent variable:	
	Delta_crime	
	A	B
	(1)	(2)
Pro_brexit	−392.19	1,691.30
	(−2,735.09, 1,950.71)	(34.16, 3,348.45)
	p = 0.75	p = 0.06*
Votes_Cast		0.004
		(0.003, 0.01)
		p = 0.00***
Constant	1,878.75	−2,822.76
	(−240.48, 3,997.98)	(−4,713.48, −932.03)
	p = 0.09*	p = 0.01***
Observations	44	44
R^2	0.003	0.57
Adjusted R^2	−0.02	0.55
Residual Std. Error	3,058.27 (df = 42)	2,037.07 (df = 41)
F Statistic	0.11 (df = 1; 42)	26.95*** (df = 2; 41)
Note:		*p<0.1; **p<0.05; ***p<0.01

Conclusion

According to the regressions brexit had an effect on the crime rates in the UK. The first regression alone wasn't enough because regions with different population had not comparable changes in the crime rates. After initiating a control variable for the population (in this case the total votes cast) the result showed a positive correlation between the crime rate increase and the fact to be a pro-brexit region.

Appendix A - Data Cleaning

Crime data

Getting the data

UK crime data was acquired from (link) data.police.uk selecting all available areas and all month from January 2015 to November 2016. (November was the last available month at the time of this analysis). The site generated one big zip file. Each month is represented by one directory. Within the directories CSV files contain the actual crime reports. One CSV fire was generated for each police force for each month.

Challenges

To make the data usable it was needed to merge all the files to one dataframe but initially it was very slow so some improvements were introduced:

- Slowness of R built in CSV parser –> read_csv was used from readr package (about 10x speedup)
- Single threaded nature of R –> doParallel package was used to process the input files on multiple threads (8)
- During merging the loaded data frames rbind always reallocated the memory for each bind and made the process quite slow –> rbindlist could preallocate all the necessary memory and merge all the data frames almost instantly

Size of data

	Size
Downloaded Zip	450 Mb
Uncompressed CSVs	2284 Mb
Cleaned DF in rdata format	85 Mb
No. variables	12
No. variables (cleaned)	6
No. observations	11,306,179

Code

The following R code was used to load, clean and save the data in rdata format.

```
library(readr)
library(foreach)
library(doParallel)
library(data.table)


# Automatic download is not working because of Onedrive limitations.
# Please download the file manually and copy it to the right location
RAW_FILE_URL = paste("https://onedrive.live.com/download?",
                     "cid=757EB1DDF6DF0848&resid=757EB1DD",
                     "F6DF0848%214618&authkey=AE479QFm0CzYd3w",
                     sep = "")

RAW_FILE = "data/raw_crime_data.zip"
```

```r
TEMP_DIR = "temp_crime"

# Create temp directory and unzip data
if(!dir.exists(TEMP_DIR)) {
  dir.create(TEMP_DIR)
  unzip( zipfile = RAW_FILE, exdir = TEMP_DIR)
}

# Iterate trough all directory to get all the filenames in a list
files <- lapply(list.files(TEMP_DIR), function(d) {
  lapply(list.files( paste(TEMP_DIR, "/", d, sep = "") ),
         function(f) paste(TEMP_DIR, "/", d, "/", f, sep = "") )
})
files <- unlist(files) # Flatten list

# Create a local cluster for parallel processing
cl <- makeCluster(8)
registerDoParallel(cl)


crime <- foreach(i=1:length(files), .packages = c("readr")) %dopar% {
          df_part <- data.frame(read_csv(files[i])) # Read CSV to DF
          # Remove unused columns
          df_part <- df_part[,c("Falls.within",
                                "Longitude",
                                "Latitude",
                                "Crime.type",
                                "Month")]
          # create year and month columns
          df_part$Year <-
            as.numeric(gsub("(\\d\\d\\d\\d)-\\d\\d", "\\1", x = df_part$Month))
          df_part$Month <-
            as.numeric(gsub("\\d\\d\\d\\d-(\\d\\d)", "\\1", x = df_part$Month))
          colnames(df_part) <-
            c("Territory", "Longitude", "Latitude", "Type", "Month", "Year")
          return(df_part)
        }
# Merge all dataframes to a big one
crime <- rbindlist(crime) # rbindlist is much much faster than rbind

stopCluster(cl)

save(crime, file = "data/crime.rdata")
```

Vote data

Getting the data

Brexit poll results were downloaded from (link) www.electoralcommission.org.uk in CSV format.

Size of data

	Size
Downloaded CSV	63 Kb
Cleaned DF in rdata format	6.5 Kb
No. variables	21
No. variables (cleaned)	5
No. observations	382

Code

The following R code was used to load, clean and save the data in rdata format.

```r
library(readr)

# Source: http://www.electoralcommission.org.uk/find-information-by-subject/
#         elections-and-referendums/upcoming-elections-and-referendums/
#         eu-referendum/electorate-and-count-information

RAW_FILE_URL = paste("http://www.electoralcommission.org.uk/__data/assets/",
                     "file/0014/212135/EU-referendum-result-data.csv", sep = "")
RAW_FILE = "data/raw_vote_data.csv"

# Download file from the internet
if(!file.exists(RAW_FILE)) {
  download.file(RAW_FILE_URL, RAW_FILE, method = "curl")
}

vote <- read_csv("data/raw_vote_data.csv")

# Keep only important columns
vote <- vote[, c("Region", "Area", "Votes_Cast", "Remain", "Leave")]

save(vote, file = "data/vote.rdata")
```

Cross referencing police force and voting regions

Getting the data

It required extensive research on the web to find data that contains both the governmental areas of the UK and also the police force regions too. (link) data.gov.uk.

Challenges

- Different naming for police forces –> It was corrected manually
- Missing date –> North Ireland police force were added manually, Scotland police forces were skipped because there were no crime data for them in the crimes dataset.

Code

The following R code was used to load, clean and save the data in rdata format.

```r
# Source: http://geoportal.statistics.gov.uk/datasets/296f0ff0013b4db484bacf2e2a8dd613_0.csv
RAW_FILE_URL = "http://geoportal.statistics.gov.uk/datasets/296f0ff0013b4db484bacf2e2a8dd613_0.csv"
RAW_FILE = "data/raw_area_police_xref.csv"

# Download file from the internet
if(!file.exists(RAW_FILE)) {
  download.file(RAW_FILE_URL, RAW_FILE, method = "curl")
}

xref <- read_csv("data/raw_area_police_xref.csv")

xref <- xref[, c(2, 6)]
colnames(xref) <- c("Local_Authority_District", "Police_Force_Area")

xref$Police_Force_Area[xref$Police_Force_Area =="Avon and Somerset"] <-
  "Avon and Somerset Constabulary"
xref$Police_Force_Area[xref$Police_Force_Area == "Bedfordshire"] <-
  "Bedfordshire Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Cambridgeshire"] <-
  "Cambridgeshire Constabulary"
xref$Police_Force_Area[xref$Police_Force_Area == "Cheshire"] <-
  "Cheshire Constabulary"
xref$Police_Force_Area[xref$Police_Force_Area == "London, City of"] <-
  "City of London Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Cleveland"] <-
  "Cleveland Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Cumbria"] <-
  "Cumbria Constabulary"
xref$Police_Force_Area[xref$Police_Force_Area == "Derbyshire"] <-
  "Derbyshire Constabulary"
xref$Police_Force_Area[xref$Police_Force_Area == "Devon & Cornwall"] <-
  "Devon & Cornwall Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Dorset"] <-
  "Dorset Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Durham"] <-
  "Durham Constabulary"
```

```r
xref$Police_Force_Area[xref$Police_Force_Area == "Dyfed-Powys"] <-
  "Dyfed-Powys Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Essex"] <-
  "Essex Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Gloucestershire"] <-
  "Gloucestershire Constabulary"
xref$Police_Force_Area[xref$Police_Force_Area == "Greater Manchester"] <-
  "Greater Manchester Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Gwent"] <-
  "Gwent Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Hampshire"] <-
  "Hampshire Constabulary"
xref$Police_Force_Area[xref$Police_Force_Area == "Hertfordshire"] <-
  "Hertfordshire Constabulary"
xref$Police_Force_Area[xref$Police_Force_Area == "Humberside"] <-
  "Humberside Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Kent"] <-
  "Kent Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Lancashire"] <-
  "Lancashire Constabulary"
xref$Police_Force_Area[xref$Police_Force_Area == "Leicestershire"] <-
  "Leicestershire Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Lincolnshire"] <-
  "Lincolnshire Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Merseyside"] <-
  "Merseyside Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Metropolitan Police"] <-
  "Metropolitan Police Service"
xref$Police_Force_Area[xref$Police_Force_Area == "Norfolk"] <-
  "Norfolk Constabulary"
xref$Police_Force_Area[xref$Police_Force_Area == "North Wales"] <-
  "North Wales Police"
xref$Police_Force_Area[xref$Police_Force_Area == "North Yorkshire"] <-
  "North Yorkshire Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Northamptonshire"] <-
  "Northamptonshire Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Northumbria"] <-
  "Northumbria Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Nottinghamshire"] <-
  "Nottinghamshire Police"
xref$Police_Force_Area[xref$Police_Force_Area == "South Wales"] <-
  "South Wales Police"
xref$Police_Force_Area[xref$Police_Force_Area == "South Yorkshire"] <-
  "South Yorkshire Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Staffordshire"] <-
  "Staffordshire Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Suffolk"] <-
  "Suffolk Constabulary"
xref$Police_Force_Area[xref$Police_Force_Area == "Surrey"] <-
  "Surrey Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Sussex"] <-
  "Sussex Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Thames Valley"] <-
```

```
  "Thames Valley Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Warwickshire"] <-
  "Warwickshire Police"
xref$Police_Force_Area[xref$Police_Force_Area == "West Mercia"] <-
  "West Mercia Police"
xref$Police_Force_Area[xref$Police_Force_Area == "West Midlands"] <-
  "West Midlands Police"
xref$Police_Force_Area[xref$Police_Force_Area == "West Yorkshire"] <-
  "West Yorkshire Police"
xref$Police_Force_Area[xref$Police_Force_Area == "Wiltshire"] <-
  "Wiltshire Police"

# Manually add Police Serrvice of Northern Ireland
xref <- rbind(xref, c("Northern Ireland", "Police Service of Northern Ireland"))

save(xref, file = "data/xref.rdata")
```

Appendix B - Data

Police forces

Number of police forces: 45Number of crimes: 11306179Number of crime types: 14

Avon and Somerset Constabulary	Greater Manchester Police	Police Service of Northern Ireland
Bedfordshire Police	Gwent Police	Northumbria Police
British Transport Police	Hampshire Constabulary	Nottinghamshire Police
Cambridgeshire Constabulary	Hertfordshire Constabulary	South Wales Police
Cheshire Constabulary	Humberside Police	South Yorkshire Police
City of London Police	Kent Police	Staffordshire Police
Cleveland Police	Lancashire Constabulary	Suffolk Constabulary
Cumbria Constabulary	Leicestershire Police	Surrey Police
Derbyshire Constabulary	Lincolnshire Police	Sussex Police
Devon & Cornwall Police	Merseyside Police	Thames Valley Police
Dorset Police	Metropolitan Police Service	Warwickshire Police
Durham Constabulary	Norfolk Constabulary	West Mercia Police
Dyfed-Powys Police	North Wales Police	West Midlands Police
Essex Police	North Yorkshire Police	West Yorkshire Police
Gloucestershire Constabulary	Northamptonshire Police	Wiltshire Police

Crime types

Anti-social behaviour	Drugs	Possession of weapons
Burglary	Shoplifting	Bicycle theft
Criminal damage and arson	Violence and sexual offences	Other crime
Other theft	Public order	Robbery
Vehicle crime	Theft from the person	

Govermental Areas

Adur	Bradford	Chelmsford	Derbyshire Dales
Allerdale	Braintree	Cheltenham	Doncaster
Amber Valley	Breckland	Cherwell	Dover
Arun	Brent	Cheshire East	Dudley
Ashfield	Brentwood	Cheshire West and Chester	Ealing
Ashford	Bridgend	Chesterfield	East Cambridgeshire
Aylesbury Vale	Brighton and Hove	Chichester	East Devon
Babergh	Bristol, City of	Chiltern	East Dorset
Barking and Dagenham	Broadland	Chorley	East Hampshire
Barnet	Bromley	Christchurch	East Hertfordshire
Barnsley	Bromsgrove	City of London	East Lindsey
Barrow-in-Furness	Broxbourne	Colchester	East Northamptonshire
Basildon	Broxtowe	Conwy	East Riding of Yorkshire
Basingstoke and Deane	Burnley	Copeland	East Staffordshire
Bassetlaw	Bury	Corby	Eastbourne
Bath and North East Somerset	Caerphilly	Cornwall	Eastleigh
Bedford	Calderdale	Cotswold	Eden
Bexley	Cambridge	County Durham	Elmbridge
Birmingham	Camden	Coventry	Enfield
Blaby	Cannock Chase	Craven	Epping Forest
Blackburn with Darwen	Canterbury	Crawley	Epsom and Ewell
Blackpool	Cardiff	Croydon	Erewash
Blaenau Gwent	Carlisle	Dacorum	Exeter
Bolsover	Carmarthenshire	Darlington	Fareham
Bolton	Castle Point	Dartford	Fenland
Boston	Central Bedfordshire	Daventry	Flintshire
Bournemouth	Ceredigion	Denbighshire	Forest Heath
Bracknell Forest	Charnwood	Derby	Forest of Dean

Fylde
Gateshead
Gedling
Gloucester
Gosport
Gravesham
Great Yarmouth
Greenwich
Guildford
Gwynedd
Hackney
Halton
Hambleton
Hammersmith and Fulham
Harborough
Haringey
Harlow
Harrogate
Harrow
Hart
Hartlepool
Hastings
Havant
Havering
Herefordshire, County of
Hertsmere
High Peak
Hillingdon
Hinckley and Bosworth
Horsham
Hounslow
Huntingdonshire
Hyndburn
Ipswich
Isle of Anglesey
Isle of Wight
Isles of Scilly
Islington
Kensington and Chelsea
Kettering
King's Lynn and West Norfolk
Kingston upon Hull, City of
Kingston upon Thames
Kirklees
Knowsley
Lambeth
Lancaster
Leeds
Leicester
Lewes
Lewisham
Lichfield
Lincoln
Liverpool
Luton
Maidstone
Maldon
Malvern Hills
Manchester
Mansfield
Medway
Melton
Mendip
Merthyr Tydfil
Merton
Mid Devon
Mid Suffolk
Mid Sussex
Middlesbrough
Milton Keynes
Mole Valley
Monmouthshire
Neath Port Talbot
New Forest
Newark and Sherwood
Newcastle-under-Lyme
Newcastle upon Tyne
Newham
Newport
North Devon
North Dorset
North East Derbyshire
North East Lincolnshire
North Hertfordshire
North Kesteven
North Lincolnshire
North Norfolk
North Somerset
North Tyneside
North Warwickshire
North West Leicestershire
Northampton
Northern Ireland
Northumberland
Norwich
Nottingham
Nuneaton and Bedworth
Oadby and Wigston
Oldham
Oxford
Pembrokeshire
Pendle
Peterborough
Plymouth
Poole
Portsmouth
Powys
Preston
Purbeck
Reading
Redbridge
Redcar and Cleveland
Redditch
Reigate and Banstead
Rhondda Cynon Taf
Ribble Valley
Richmond upon Thames
Richmondshire
Rochdale
Rochford
Rossendale
Rother
Rotherham
Rugby
Runnymede
Rushcliffe
Rushmoor
Rutland
Ryedale
Salford
Sandwell
Scarborough
Sedgemoor
Sefton
Selby
Sevenoaks
Sheffield
Shepway
Shropshire
Slough
Solihull
South Bucks
South Cambridgeshire
South Derbyshire
South Gloucestershire
South Hams
South Holland
South Kesteven
South Lakeland
South Norfolk
South Northamptonshire
South Oxfordshire
South Ribble
South Somerset
South Staffordshire
South Tyneside
Southampton
Southend-on-Sea
Southwark
Spelthorne
St Albans
St Edmundsbury
St. Helens
Stafford
Staffordshire Moorlands
Stevenage
Stockport
Stockton-on-Tees
Stoke-on-Trent
Stratford-on-Avon
Stroud
Suffolk Coastal
Sunderland
Surrey Heath
Sutton
Swale
Swansea
Swindon
Tameside
Tamworth
Tandridge
Taunton Deane
Teignbridge
Telford and Wrekin
Tendring
Test Valley
Tewkesbury
Thanet
Three Rivers
Thurrock
Tonbridge and Malling
Torbay
Torfaen
Torridge
Tower Hamlets
Trafford
Tunbridge Wells
Uttlesford
Vale of Glamorgan
Vale of White Horse
Wakefield
Walsall
Waltham Forest
Wandsworth
Warrington
Warwick
Watford
Waveney
Waverley
Wealden
Wellingborough
Welwyn Hatfield
West Berkshire
West Devon
West Dorset
West Lancashire
West Lindsey
West Oxfordshire
West Somerset
Westminster
Weymouth and Portland
Wigan
Wiltshire
Winchester
Windsor and Maidenhead
Wirral
Woking
Wokingham
Wolverhampton
Worcester
Worthing
Wrexham
Wychavon
Wycombe
Wyre
Wyre Forest
York

Appendix C - Code

Chapter 2 - Understanding the data

```r
load("data/crime.rdata") # Load crime data
load("data/vote.rdata") # Load vote data
load("data/xref.rdata") # Load police/govermental xref data

relevant_crime <- subset(crime, Type=="Anti-social behaviour" |
                           Type == "Violence and sexual offences" )

num_crimes = nrow(crime)
num_relevant_crimes = nrow(relevant_crime)
num_antisocial = nrow(subset(relevant_crime, Type=="Anti-social behaviour"))
num_violence = nrow(subset(relevant_crime, Type=="Violence and sexual offences"))


library(ggplot2)
library(scales)
library(gridExtra)

p1 <- ggplot(data=data.frame(Crime_Type=c("Relevant Crimes","Other Crimes"),
                             y=c(num_relevant_crimes, num_crimes-num_relevant_crimes)),
             aes(x="", y=y, fill=Crime_Type)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start=0) +
  theme(axis.text.x=element_blank()) +
  geom_text(aes(y = y/3 + c(0, cumsum(y)[-length(y)]),
            label = percent(y/nrow(crime))), size=3) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
        axis.ticks = element_blank()) +
  ggtitle("Ratio of Relevant Crimes")

p2 <- ggplot(data=data.frame(Crime_Type=c("Anti-social behavior","Violence and sex off."),
                             y=c(num_antisocial, num_violence)),
             aes(x="", y=y, fill=Crime_Type)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start=0) +
  theme(axis.text.x=element_blank()) +
  geom_text(aes(y = y/3 + c(0, cumsum(y)[-length(y)]),
            label = percent(y/num_relevant_crimes)), size=3) +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank(),
        axis.ticks = element_blank()) +
  ggtitle("Structure of Relevant Crimes")

grid.arrange(p1, p2, ncol=2)


crime$Count <- 1
aggregated_crime <- aggregate(Count ~ Territory + Year + Month, data=crime, FUN = sum)

aggregated_crime$date <- as.Date( paste( aggregated_crime$Year, aggregated_crime$Month,
                                         "1" , sep = "." )  , format = "%Y.%m.%d" )
```

```
ggplot(data=aggregated_crime, aes(x=date, y=Count)) +
  geom_line(data = subset(aggregated_crime, Territory == "Avon and Somerset Constabulary"),
            aes(colour = "Avon and Somerset")) +
  geom_line(data = subset(aggregated_crime, Territory == "Bedfordshire Police"),
            aes(colour = "Bedfordshire")) +
  geom_line(data = subset(aggregated_crime, Territory == "Kent Police"),
            aes(colour = "Kent")) +
  geom_line(data = subset(aggregated_crime, Territory == "Staffordshire Police"),
            aes(colour = "Staffordshire")) +
  scale_colour_manual("",
                breaks = c("Avon and Somerset", "Bedfordshire", "Kent", "Staffordshire"),
                values = c("red", "green", "blue", "orange")) +
  ggtitle("Crimes over time") + xlab("Date") + ylab("Monthly Number of Crimes")


vote <- merge(x = vote, y = xref,
              by.x = "Area",
              by.y = "Local_Authority_District",
              all.x = TRUE)
# Remove areas that don't have Police Force Area because those are in Scotland
# and we don't have crime data for them
vote <- subset(vote, !is.na(Police_Force_Area))
# Aggregate votes per Police Force Areas
aggregated_vote <- aggregate(cbind(Remain, Leave, Votes_Cast) ~ Police_Force_Area,
                             data = vote, FUN = sum)
# Recalculate Pro_brexit binary variable for each areas.
aggregated_vote$Pro_brexit <- as.numeric(aggregated_vote$Leave > aggregated_vote$Remain)
aggregated_vote <- aggregated_vote[, c(1,4,5)]

# Merge vote and crime data
crime_vote <- merge(x = aggregated_crime, y=aggregated_vote,
                    by.x="Territory", by.y="Police_Force_Area", all.x=TRUE)
# Filtering out Police Force Areas that don't have vote data.
# These are special police forces that are not binded to areas like "British Transport Police"
crime_vote <- subset(crime_vote, !is.na(Pro_brexit))
```

Chapter 3 - Regression Analysis

```
# Keeping only months (May, June, July)
reg_data <- subset(crime_vote, Month == 5 | Month == 6 | Month == 7)
# Separating data from 2015 and 2016 and merge them calculating the difference
reg_2015 <- subset(reg_data, Year == 2015)
reg_2016 <- subset(reg_data, Year == 2016)
reg_2015 <- reg_2015[, c(1, 4)]
colnames(reg_2015) <- c("Territory", "Count2015")
reg_2015 <- aggregate(Count2015 ~ Territory, data = reg_2015, FUN = sum)
reg_2016 <- aggregate(Count ~ Territory + Votes_Cast + Pro_brexit,
                      data = reg_2016, FUN = sum)
reg_data <- merge(x = reg_2016, y=reg_2015, by="Territory", all.x=TRUE)
reg_data$Delta_crime <- reg_data$Count - reg_data$Count2015
```

```r
reg1 <- lm(Delta_crime ~ Pro_brexit, data = reg_data)
reg2 <- lm(Delta_crime ~ Pro_brexit + Votes_Cast, data = reg_data)

library(stargazer)

stargazer(list(reg1, reg2), column.labels = c("A", "B"),
          digits=2, ci=TRUE, report = "vcsp*",
          title="Comparing regressions",
          type = "latex", header=FALSE)
```