Natalia Kusek - Formative feedback - Crime prediction

Goal: Predicting number of each crime occurences in London one month in advance.

Hypothese1: Number of each crimes in each location normalised per number of people living in this location depends on previous number of crime occuriences in this location. Hypothese2: Number of each crimes in each location normalised per number of people living in this location is higher where proportion of the population experiencing deprivation relating to low income and quality of the local environment is higher. Hypothese3: Number of each crimes in each location normalised per number of people living in this location is higher where population density, number of employee, number of properties, number of owned properties and number of properties in council tax band F-H houses per square meter (area) is higher.

Loading data sets

```
df1 = read.csv('/Users/nataliakusek/Middlesex/CST4070-R/Formative assignment-20191124/data/LondonCensus
df2 = read.csv('/Users/nataliakusek/Middlesex/CST4070-R/Formative assignment-20191124/data/MPS_Ward_Lev
```

```
str(df1)
```

```
## 'data.frame':    625 obs. of  20 variables:
##  $ WardCode  : Factor w/ 625 levels "E05000026","E05000027",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ WardName  : Factor w/ 604 levels "Abbey","Abbey Road",..: 1 10 23 92 176 177 218 225 270 338 ...
##  $ borough   : Factor w/ 32 levels "Barking and Dagenham",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ NESW      : Factor w/ 5 levels "Central","East",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ AreaSqKm  : num  1.3 1.4 1.3 3.4 3.5 1.4 1.1 1.3 2 1.6 ...
##  $ lon       : num  0.0779 0.1483 0.119 0.14 0.1736 ...
##  $ lat       : num  51.5 51.5 51.6 51.6 51.6 ...
##  $ IncomeScor: num  0.27 0.28 0.25 0.27 0.19 0.27 0.36 0.27 0.31 0.17 ...
##  $ LivingEnSc: num  42.8 28 31.6 34.8 21.2 ...
##  $ NoEmployee: int  7900 800 1100 1700 4000 1000 2800 1300 2500 1600 ...
##  $ GrenSpace : num  19.6 22.4 3 56.4 51.1 18.1 20.3 17.1 38.4 30.3 ...
##  $ PopDen    : num  9885 7464 8923 2971 3014 ...
##  $ BornUK    : int  5459 7824 8075 7539 8514 7880 6447 8244 8183 7660 ...
##  $ NotBornUK : int  7327 2561 3470 2482 1992 3744 6005 3023 2603 3818 ...
##  $ NoCTFtoH  : num  0.1 0.1 0.1 0.4 0.5 0 0.1 0.1 0 7.7 ...
##  $ NoDwelling: int  4733 4045 4378 4050 3976 4321 4662 4293 4409 3787 ...
##  $ NoFlats   : int  3153 574 837 1400 742 933 3368 657 1606 852 ...
##  $ NoHouses  : int  1600 3471 3541 2662 3235 3388 1343 3639 2812 2936 ...
##  $ NoOwndDwel: int  1545 1849 2093 2148 2646 1913 1233 1938 1832 2618 ...
##  $ MedHPrice : int  177000 160000 170000 195000 191750 167250 145000 155000 155000 250000 ...
```

```
str(df2)
```

```
## 'data.frame':    20126 obs. of  110 variables:
##  $ WardCode     : Factor w/ 629 levels "E05000026","E05000027",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ WardName     : Factor w/ 606 levels "Abbey","Abbey Road",..: 1 10 23 93 175 176 217 224 269 341 .
##  $ Borough      : Factor w/ 32 levels "Barking and Dagenham",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ MajorCategory: Factor w/ 9 levels "Burglary","Criminal Damage",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ MinorCategory: Factor w/ 32 levels "Assault with Injury",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ T_201004     : int  9 7 10 10 3 8 7 10 8 4 ...
##  $ T_201005     : int  4 3 6 10 5 3 6 3 7 3 ...
##  $ T_201006     : int  6 8 21 11 2 2 7 10 9 10 ...
##  $ T_201007     : int  6 12 7 8 6 12 8 9 10 6 ...
##  $ T_201008     : int  12 12 13 13 4 18 3 14 7 4 ...
##  $ T_201009     : int  5 14 8 7 4 12 2 9 3 5 ...
##  $ T_201010     : int  3 4 4 12 8 10 5 9 15 8 ...
```

```
##  $ T_201011     : int  11 8 12 16 3 4 5 18 10 7 ...
##  $ T_201012     : int  8 7 9 4 6 5 2 8 8 9 ...
##  $ T_201101     : int  8 8 16 18 10 10 11 7 13 17 ...
##  $ T_201102     : int  7 8 9 8 4 8 9 4 8 18 ...
##  $ T_201103     : int  4 10 21 18 7 4 5 12 12 10 ...
##  $ T_201104     : int  7 12 10 9 7 6 5 6 15 5 ...
##  $ T_201105     : int  13 8 6 6 4 6 7 17 9 12 ...
##  $ T_201106     : int  5 8 7 6 2 9 9 3 6 8 ...
##  $ T_201107     : int  4 11 10 7 3 9 5 5 4 5 ...
##  $ T_201108     : int  5 9 10 13 4 4 3 3 4 4 ...
##  $ T_201109     : int  8 5 15 7 6 4 8 2 2 8 ...
##  $ T_201110     : int  8 3 10 10 8 8 11 7 17 5 ...
##  $ T_201111     : int  7 9 13 11 16 10 11 3 16 11 ...
##  $ T_201112     : int  7 4 12 15 14 9 9 4 16 18 ...
##  $ T_201201     : int  9 11 18 14 9 3 6 9 11 19 ...
##  $ T_201202     : int  6 15 3 9 10 12 13 3 9 15 ...
##  $ T_201203     : int  5 17 14 13 8 9 4 12 11 19 ...
##  $ T_201204     : int  6 10 10 10 8 10 10 8 3 9 ...
##  $ T_201205     : int  9 9 14 15 11 6 7 3 5 11 ...
##  $ T_201206     : int  6 8 6 8 6 10 7 6 5 5 ...
##  $ T_201207     : int  6 6 14 14 3 11 11 4 4 2 ...
##  $ T_201208     : int  4 5 11 10 4 6 10 7 5 5 ...
##  $ T_201209     : int  5 3 8 7 5 14 10 1 6 8 ...
##  $ T_201210     : int  5 6 12 10 5 4 6 15 10 14 ...
##  $ T_201211     : int  11 7 12 14 6 15 4 11 19 12 ...
##  $ T_201212     : int  1 5 18 13 8 19 8 11 11 9 ...
##  $ T_201301     : int  6 10 13 13 11 11 6 17 10 15 ...
##  $ T_201302     : int  9 14 9 10 10 10 5 15 12 13 ...
##  $ T_201303     : int  9 4 12 8 5 12 4 13 18 9 ...
##  $ T_201304     : int  1 2 20 11 8 6 4 11 10 4 ...
##  $ T_201305     : int  6 6 9 10 8 4 5 9 10 7 ...
##  $ T_201306     : int  4 3 7 10 5 2 7 8 5 7 ...
##  $ T_201307     : int  5 3 5 5 7 7 9 4 3 1 ...
##  $ T_201308     : int  5 2 4 8 3 10 4 4 10 5 ...
##  $ T_201309     : int  2 1 7 7 2 4 1 4 9 10 ...
##  $ T_201310     : int  2 6 16 11 6 4 5 10 8 15 ...
##  $ T_201311     : int  6 3 15 10 5 11 3 12 10 10 ...
##  $ T_201312     : int  3 5 10 7 5 4 3 8 9 1 ...
##  $ T_201401     : int  5 7 6 12 6 5 3 8 1 10 ...
##  $ T_201402     : int  5 4 6 7 5 7 6 5 4 13 ...
##  $ T_201403     : int  9 11 8 6 5 12 7 7 8 10 ...
##  $ T_201404     : int  2 7 2 7 1 10 3 7 5 3 ...
##  $ T_201405     : int  6 4 10 6 4 3 3 6 10 2 ...
##  $ T_201406     : int  3 4 11 5 2 8 4 3 8 7 ...
##  $ T_201407     : int  4 5 6 11 13 8 1 8 7 1 ...
##  $ T_201408     : int  2 5 13 6 4 13 4 16 5 8 ...
##  $ T_201409     : int  2 0 9 5 8 5 3 9 6 4 ...
##  $ T_201410     : int  1 6 11 7 10 5 9 3 6 6 ...
##  $ T_201411     : int  6 10 8 13 8 8 5 8 17 21 ...
##  $ T_201412     : int  5 7 9 6 7 10 6 15 4 15 ...
##  $ T_201501     : int  0 9 19 6 6 15 11 17 11 6 ...
##  $ T_201502     : int  7 4 7 10 0 9 4 4 3 10 ...
##  $ T_201503     : int  1 5 2 9 6 11 11 6 5 7 ...
##  $ T_201504     : int  7 3 5 3 0 5 5 2 4 6 ...
```

```
##  $ T_201505     : int  4 1 8 2 2 5 6 2 2 1 ...
##  $ T_201506     : int  2 1 2 7 5 10 3 3 1 4 ...
##  $ T_201507     : int  2 5 7 7 5 3 10 5 6 5 ...
##  $ T_201508     : int  6 2 5 6 5 4 4 3 4 4 ...
##  $ T_201509     : int  3 5 6 4 3 5 5 2 5 4 ...
##  $ T_201510     : int  2 4 10 6 4 11 7 4 6 7 ...
##  $ T_201511     : int  7 3 3 12 5 12 5 8 4 13 ...
##  $ T_201512     : int  3 8 9 10 7 8 8 11 4 9 ...
##  $ T_201601     : int  4 6 8 7 5 4 5 9 3 7 ...
##  $ T_201602     : int  6 5 7 3 7 5 4 0 4 6 ...
##  $ T_201603     : int  5 7 6 4 1 3 2 2 8 6 ...
##  $ T_201604     : int  2 2 4 1 3 3 2 3 1 2 ...
##  $ T_201605     : int  2 5 2 7 3 2 7 1 5 4 ...
##  $ T_201606     : int  10 4 6 2 1 5 3 3 2 4 ...
##  $ T_201607     : int  3 5 4 5 4 3 1 2 1 1 ...
##  $ T_201608     : int  2 0 4 2 2 3 4 2 2 5 ...
##  $ T_201609     : int  3 0 4 4 7 2 3 3 2 1 ...
##  $ T_201610     : int  1 2 4 3 8 5 0 5 6 6 ...
##  $ T_201611     : int  3 2 6 4 2 7 3 4 2 1 ...
##  $ T_201612     : int  3 5 5 11 7 8 6 2 3 11 ...
##  $ T_201701     : int  3 4 12 10 2 6 8 8 2 3 ...
##  $ T_201702     : int  1 8 15 6 4 14 2 9 6 12 ...
##  $ T_201703     : int  7 3 4 7 7 9 14 7 8 2 ...
##  $ T_201704     : int  12 5 5 7 2 11 6 2 9 5 ...
##  $ T_201705     : int  12 5 14 4 8 6 10 4 10 4 ...
##  $ T_201706     : int  3 11 5 2 3 4 4 6 6 2 ...
##  $ T_201707     : int  5 9 10 2 5 8 6 6 10 2 ...
##  $ T_201708     : int  4 4 5 4 3 4 8 4 5 2 ...
##  $ T_201709     : int  3 0 7 3 5 10 7 3 4 3 ...
##  $ T_201710     : int  3 4 15 5 3 8 10 7 17 3 ...
##  $ T_201711     : int  4 1 5 4 3 10 2 6 3 9 ...
##  $ T_201712     : int  3 1 11 11 2 9 2 13 5 8 ...
##  $ T_201801     : int  7 8 7 12 11 15 2 4 4 12 ...
##   [list output truncated]
```

changing dataframe into datatable

```
library(data.table)
df1 = as.data.table(df1)
df2 = as.data.table(df2)
```

```
head(df1)
```

```
##     WardCode       WardName              borough NESW AreaSqKm      lon
## 1: E05000026          Abbey Barking and Dagenham East      1.3 0.077935
## 2: E05000027         Alibon Barking and Dagenham East      1.4 0.148270
## 3: E05000028      Becontree Barking and Dagenham East      1.3 0.118957
## 4: E05000029 Chadwell Heath Barking and Dagenham East      3.4 0.139985
## 5: E05000030      Eastbrook Barking and Dagenham East      3.5 0.173581
## 6: E05000031       Eastbury Barking and Dagenham East      1.4 0.105683
##         lat IncomeScor LivingEnSc NoEmployee GrenSpace PopDen BornUK
## 1: 51.53971       0.27      42.76       7900      19.6 9884.6   5459
## 2: 51.54559       0.28      27.96        800      22.4 7464.3   7824
## 3: 51.55453       0.25      31.59       1100       3.0 8923.1   8075
```

```
## 4: 51.58475       0.27      34.78       1700      56.4 2970.6   7539
## 5: 51.55365       0.19      21.25       4000      51.1 3014.3   8514
## 6: 51.53590       0.27      31.16       1000      18.1 8357.1   7880
##    NotBornUK NoCTFtoH NoDwelling NoFlats NoHouses NoOwndDwel MedHPrice
## 1:      7327      0.1       4733    3153     1600        1545    177000
## 2:      2561      0.1       4045     574     3471        1849    160000
## 3:      3470      0.1       4378     837     3541        2093    170000
## 4:      2482      0.4       4050    1400     2662        2148    195000
## 5:      1992      0.5       3976     742     3235        2646    191750
## 6:      3744      0.0       4321     933     3388        1913    167250
```

```
head(df2)
```

```
##     WardCode       WardName              Borough MajorCategory
## 1: E05000026          Abbey Barking and Dagenham      Burglary
## 2: E05000027         Alibon Barking and Dagenham      Burglary
## 3: E05000028      Becontree Barking and Dagenham      Burglary
## 4: E05000029 Chadwell Heath Barking and Dagenham      Burglary
## 5: E05000030      Eastbrook Barking and Dagenham      Burglary
## 6: E05000031       Eastbury Barking and Dagenham      Burglary
##             MinorCategory T_201004 T_201005 T_201006 T_201007 T_201008
## 1: Burglary In A Dwelling        9        4        6        6       12
## 2: Burglary In A Dwelling        7        3        8       12       12
## 3: Burglary In A Dwelling       10        6       21        7       13
## 4: Burglary In A Dwelling       10       10       11        8       13
## 5: Burglary In A Dwelling        3        5        2        6        4
## 6: Burglary In A Dwelling        8        3        2       12       18
##    T_201009 T_201010 T_201011 T_201012 T_201101 T_201102 T_201103 T_201104
## 1:        5        3       11        8        8        7        4        7
## 2:       14        4        8        7        8        8       10       12
## 3:        8        4       12        9       16        9       21       10
## 4:        7       12       16        4       18        8       18        9
## 5:        4        8        3        6       10        4        7        7
## 6:       12       10        4        5       10        8        4        6
##    T_201105 T_201106 T_201107 T_201108 T_201109 T_201110 T_201111 T_201112
## 1:       13        5        4        5        8        8        7        7
## 2:        8        8       11        9        5        3        9        4
## 3:        6        7       10       10       15       10       13       12
## 4:        6        6        7       13        7       10       11       15
## 5:        4        2        3        4        6        8       16       14
## 6:        6        9        9        4        4        8       10        9
##    T_201201 T_201202 T_201203 T_201204 T_201205 T_201206 T_201207 T_201208
## 1:        9        6        5        6        9        6        6        4
## 2:       11       15       17       10        9        8        6        5
## 3:       18        3       14       10       14        6       14       11
## 4:       14        9       13       10       15        8       14       10
## 5:        9       10        8        8       11        6        3        4
## 6:        3       12        9       10        6       10       11        6
##    T_201209 T_201210 T_201211 T_201212 T_201301 T_201302 T_201303 T_201304
## 1:        5        5       11        1        6        9        9        1
## 2:        3        6        7        5       10       14        4        2
## 3:        8       12       12       18       13        9       12       20
## 4:        7       10       14       13       13       10        8       11
## 5:        5        5        6        8       11       10        5        8
## 6:       14        4       15       19       11       10       12        6
```

```
##    T_201305 T_201306 T_201307 T_201308 T_201309 T_201310 T_201311 T_201312
## 1:        6        4        5        5        2        2        6        3
## 2:        6        3        3        2        1        6        3        5
## 3:        9        7        5        4        7       16       15       10
## 4:       10       10        5        8        7       11       10        7
## 5:        8        5        7        3        2        6        5        5
## 6:        4        2        7       10        4        4       11        4
##    T_201401 T_201402 T_201403 T_201404 T_201405 T_201406 T_201407 T_201408
## 1:        5        5        9        2        6        3        4        2
## 2:        7        4       11        7        4        4        5        5
## 3:        6        6        8        2       10       11        6       13
## 4:       12        7        6        7        6        5       11        6
## 5:        6        5        5        1        4        2       13        4
## 6:        5        7       12       10        3        8        8       13
##    T_201409 T_201410 T_201411 T_201412 T_201501 T_201502 T_201503 T_201504
## 1:        2        1        6        5        0        7        1        7
## 2:        0        6       10        7        9        4        5        3
## 3:        9       11        8        9       19        7        2        5
## 4:        5        7       13        6        6       10        9        3
## 5:        8       10        8        7        6        0        6        0
## 6:        5        5        8       10       15        9       11        5
##    T_201505 T_201506 T_201507 T_201508 T_201509 T_201510 T_201511 T_201512
## 1:        4        2        2        6        3        2        7        3
## 2:        1        1        5        2        5        4        3        8
## 3:        8        2        7        5        6       10        3        9
## 4:        2        7        7        6        4        6       12       10
## 5:        2        5        5        5        3        4        5        7
## 6:        5       10        3        4        5       11       12        8
##    T_201601 T_201602 T_201603 T_201604 T_201605 T_201606 T_201607 T_201608
## 1:        4        6        5        2        2       10        3        2
## 2:        6        5        7        2        5        4        5        0
## 3:        8        7        6        4        2        6        4        4
## 4:        7        3        4        1        7        2        5        2
## 5:        5        7        1        3        3        1        4        2
## 6:        4        5        3        3        2        5        3        3
##    T_201609 T_201610 T_201611 T_201612 T_201701 T_201702 T_201703 T_201704
## 1:        3        1        3        3        3        1        7       12
## 2:        0        2        2        5        4        8        3        5
## 3:        4        4        6        5       12       15        4        5
## 4:        4        3        4       11       10        6        7        7
## 5:        7        8        2        7        2        4        7        2
## 6:        2        5        7        8        6       14        9       11
##    T_201705 T_201706 T_201707 T_201708 T_201709 T_201710 T_201711 T_201712
## 1:       12        3        5        4        3        3        4        3
## 2:        5       11        9        4        0        4        1        1
## 3:       14        5       10        5        7       15        5       11
## 4:        4        2        2        4        3        5        4       11
## 5:        8        3        5        3        5        3        3        2
## 6:        6        4        8        4       10        8       10        9
##    T_201801 T_201802 T_201803 T_201804 T_201805 T_201806 T_201807 T_201808
## 1:        7        6        8        5        5        4        6        3
## 2:        8        8        7        1        5        3        2        4
## 3:        7        6        7        8        6        4        7        3
## 4:       12       10        7        5        8        6        5        4
```

```
## 5:       11        2        5        2        1        2        3        3
## 6:       15       15       13        9        8        9        8        6
##    T_201809 T_201810 T_201811 T_201812
## 1:        6        6        4        8
## 2:        4        8        4       10
## 3:        6        5       10        9
## 4:        8        5       10        8
## 5:        6        6       11        2
## 6:        9        6       16        8
```

merging datasets

```
df = merge(df1, df2, by=c('WardCode'), all.y=TRUE)
```

```
dim(df)
```

```
## [1] 20126   129
```

removing NA values

```
data = df[(complete.cases(df))]
dim(data)
```

```
## [1] 18238   129
```

```
new_no  = (dim(df)[1] - dim(data)[1])/dim(df)[1]
print(paste('Percentage of observation removed:', round(new_no*100, 1)))
```

```
## [1] "Percentage of observation removed: 9.4"
```

```
levels(data[, MajorCategory])
```

```
## [1] "Burglary"                    "Criminal Damage"
## [3] "Drugs"                       "Fraud or Forgery"
## [5] "Other Notifiable Offences"   "Robbery"
## [7] "Sexual Offences"             "Theft and Handling"
## [9] "Violence Against The Person"
```

```
class(data)
```

```
## [1] "data.table" "data.frame"
```

changing WardCode factor into numerical

```
data[, WardCode := as.integer(gsub("E", "", WardCode))]
```

changing MajorCategory into binomial values

```
library(fastDummies)
data = dummy_cols(data, select_columns = 'MajorCategory')
```

transposing T__ columns into rows

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data = melt(data, measure.var = c(colnames(select(data ,contains('T_')))))
colnames(data)[colnames(data)=="variable"] = "CrimeMonth"
colnames(data)[colnames(data)=="value"] = "NoOfCrimes"
```

changing CrimeMonth column into date

```
data[, CrimeMonth:=as.Date(paste(substr(CrimeMonth, 3, 8), "01", sep=""), "%Y%m%d")]
```

```
colnames(data)
```

```
##  [1] "WardCode"
##  [2] "WardName.x"
##  [3] "borough"
##  [4] "NESW"
##  [5] "AreaSqKm"
##  [6] "lon"
##  [7] "lat"
##  [8] "IncomeScor"
##  [9] "LivingEnSc"
## [10] "NoEmployee"
## [11] "GrenSpace"
## [12] "PopDen"
## [13] "BornUK"
## [14] "NotBornUK"
## [15] "NoCTFtoH"
## [16] "NoDwelling"
## [17] "NoFlats"
## [18] "NoHouses"
## [19] "NoOwndDwel"
## [20] "MedHPrice"
## [21] "WardName.y"
## [22] "Borough"
## [23] "MajorCategory"
## [24] "MinorCategory"
## [25] "MajorCategory_Burglary"
## [26] "MajorCategory_Criminal Damage"
## [27] "MajorCategory_Drugs"
## [28] "MajorCategory_Fraud or Forgery"
## [29] "MajorCategory_Other Notifiable Offences"
## [30] "MajorCategory_Robbery"
## [31] "MajorCategory_Sexual Offences"
```

```
## [32] "MajorCategory_Theft and Handling"
## [33] "MajorCategory_Violence Against The Person"
## [34] "CrimeMonth"
## [35] "NoOfCrimes"
```

deleting unneeded columns

```
data[, c("MinorCategory", "MajorCategory", "WardName.x", "borough" , "lon", "lat", "WardName.y", "Borou
```

summing NoOfCrimes for MajorCrimes after removing MinorCategory column

```
data = data[, sum(NoOfCrimes), by=c(colnames(data)[colnames(data) != "NoOfCrimes"])]
```

```
colnames(data)[colnames(data) == "V1"] = "NoOfCrimes"
```

calculating CrimerRatio

```
data[, CrimeRatio:= round(NoOfCrimes/(BornUK + NotBornUK), 10) ]
```

```
data[, NoOfCrimes :=NULL ]
```

computing ratios for number of employee, number of properties, number of owned properties and number of properties in council tax band F-H houses per square meter

```
data[, NoEmployeeRatio:=NoEmployee/AreaSqKm]
data[, NoDwellingRatio:=NoDwelling/AreaSqKm]
data[, NoOwndDwelNoRatio:=NoOwndDwel/AreaSqKm]
data[, NoCTFtoHNoRatio:=NoCTFtoH/AreaSqKm]
```

deleting columns

```
data[, c('BornUK', 'NotBornUK', 'NoEmployee', 'NoFlats', 'NoHouses', 'NoOwndDwel',
         'NoCTFtoH', 'GrenSpace', 'NoDwelling', 'AreaSqKm', 'NESW', 'MedHPrice'):=NULL]
```

creating PrevCrimeRatio

```
data = data[order(WardCode, MajorCategory_Burglary, `MajorCategory_Criminal Damage`, MajorCategory_Drug
data[, PreviousCrimeRatio:= shift(CrimeRatio), by=c('WardCode', 'MajorCategory_Burglary', 'MajorCategor
```

```
data = data[(complete.cases(data))]
```

creating train test sets

```
dataTrain = data[CrimeMonth >= '2010-05-01' & CrimeMonth <= '2017-12-01', ]
dataTest = data[CrimeMonth >='2018-01-01', ]
```

```
head(dataTrain)
```

```
##    WardCode IncomeScor LivingEnSc PopDen MajorCategory_Burglary
## 1:  5000026       0.27      42.76 9884.6                      0
```

```
## 2:  5000026       0.27      42.76 9884.6                      0
## 3:  5000026       0.27      42.76 9884.6                      0
## 4:  5000026       0.27      42.76 9884.6                      0
## 5:  5000026       0.27      42.76 9884.6                      0
## 6:  5000026       0.27      42.76 9884.6                      0
##    MajorCategory_Criminal Damage MajorCategory_Drugs
## 1:                             0                   0
## 2:                             0                   0
## 3:                             0                   0
## 4:                             0                   0
## 5:                             0                   0
## 6:                             0                   0
##    MajorCategory_Fraud or Forgery MajorCategory_Other Notifiable Offences
## 1:                              0                                       0
## 2:                              0                                       0
## 3:                              0                                       0
## 4:                              0                                       0
## 5:                              0                                       0
## 6:                              0                                       0
##    MajorCategory_Robbery MajorCategory_Sexual Offences
## 1:                     0                             0
## 2:                     0                             0
## 3:                     0                             0
## 4:                     0                             0
## 5:                     0                             0
## 6:                     0                             0
##    MajorCategory_Theft and Handling
## 1:                                0
## 2:                                0
## 3:                                0
## 4:                                0
## 5:                                0
## 6:                                0
##    MajorCategory_Violence Against The Person CrimeMonth  CrimeRatio
## 1:                                         1 2010-05-01 0.004145159
## 2:                                         1 2010-06-01 0.004849054
## 3:                                         1 2010-07-01 0.003519474
## 4:                                         1 2010-08-01 0.003597685
## 5:                                         1 2010-09-01 0.002424527
## 6:                                         1 2010-10-01 0.003441264
##    NoEmployeeRatio NoDwellingRatio NoOwndDwelNoRatio NoCTFtoHNoRatio
## 1:        6076.923        3640.769          1188.462      0.07692308
## 2:        6076.923        3640.769          1188.462      0.07692308
## 3:        6076.923        3640.769          1188.462      0.07692308
## 4:        6076.923        3640.769          1188.462      0.07692308
## 5:        6076.923        3640.769          1188.462      0.07692308
## 6:        6076.923        3640.769          1188.462      0.07692308
##    PreviousCrimeRatio
## 1:        0.005240106
## 2:        0.004145159
## 3:        0.004849054
## 4:        0.003519474
## 5:        0.003597685
## 6:        0.002424527
```

```
head(dataTest)
```

```
##    WardCode IncomeScor LivingEnSc PopDen MajorCategory_Burglary
## 1:  5000026       0.27      42.76 9884.6                      0
## 2:  5000026       0.27      42.76 9884.6                      0
## 3:  5000026       0.27      42.76 9884.6                      0
## 4:  5000026       0.27      42.76 9884.6                      0
## 5:  5000026       0.27      42.76 9884.6                      0
## 6:  5000026       0.27      42.76 9884.6                      0
##    MajorCategory_Criminal Damage MajorCategory_Drugs
## 1:                             0                   0
## 2:                             0                   0
## 3:                             0                   0
## 4:                             0                   0
## 5:                             0                   0
## 6:                             0                   0
##    MajorCategory_Fraud or Forgery MajorCategory_Other Notifiable Offences
## 1:                              0                                       0
## 2:                              0                                       0
## 3:                              0                                       0
## 4:                              0                                       0
## 5:                              0                                       0
## 6:                              0                                       0
##    MajorCategory_Robbery MajorCategory_Sexual Offences
## 1:                     0                             0
## 2:                     0                             0
## 3:                     0                             0
## 4:                     0                             0
## 5:                     0                             0
## 6:                     0                             0
##    MajorCategory_Theft and Handling
## 1:                                0
## 2:                                0
## 3:                                0
## 4:                                0
## 5:                                0
## 6:                                0
##    MajorCategory_Violence Against The Person CrimeMonth  CrimeRatio
## 1:                                         1 2018-01-01 0.004536211
## 2:                                         1 2018-02-01 0.005005475
## 3:                                         1 2018-03-01 0.005552948
## 4:                                         1 2018-04-01 0.006100422
## 5:                                         1 2018-05-01 0.004692633
## 6:                                         1 2018-06-01 0.004770843
##    NoEmployeeRatio NoDwellingRatio NoOwndDwelNoRatio NoCTFtoHNoRatio
## 1:        6076.923        3640.769          1188.462      0.07692308
## 2:        6076.923        3640.769          1188.462      0.07692308
## 3:        6076.923        3640.769          1188.462      0.07692308
## 4:        6076.923        3640.769          1188.462      0.07692308
## 5:        6076.923        3640.769          1188.462      0.07692308
## 6:        6076.923        3640.769          1188.462      0.07692308
##    PreviousCrimeRatio
## 1:        0.005396527
## 2:        0.004536211
```

```
## 3:        0.005005475
## 4:        0.005552948
## 5:        0.006100422
## 6:        0.004692633
```

linear model

```
model = lm(CrimeRatio ~ ., data=dataTrain)
```

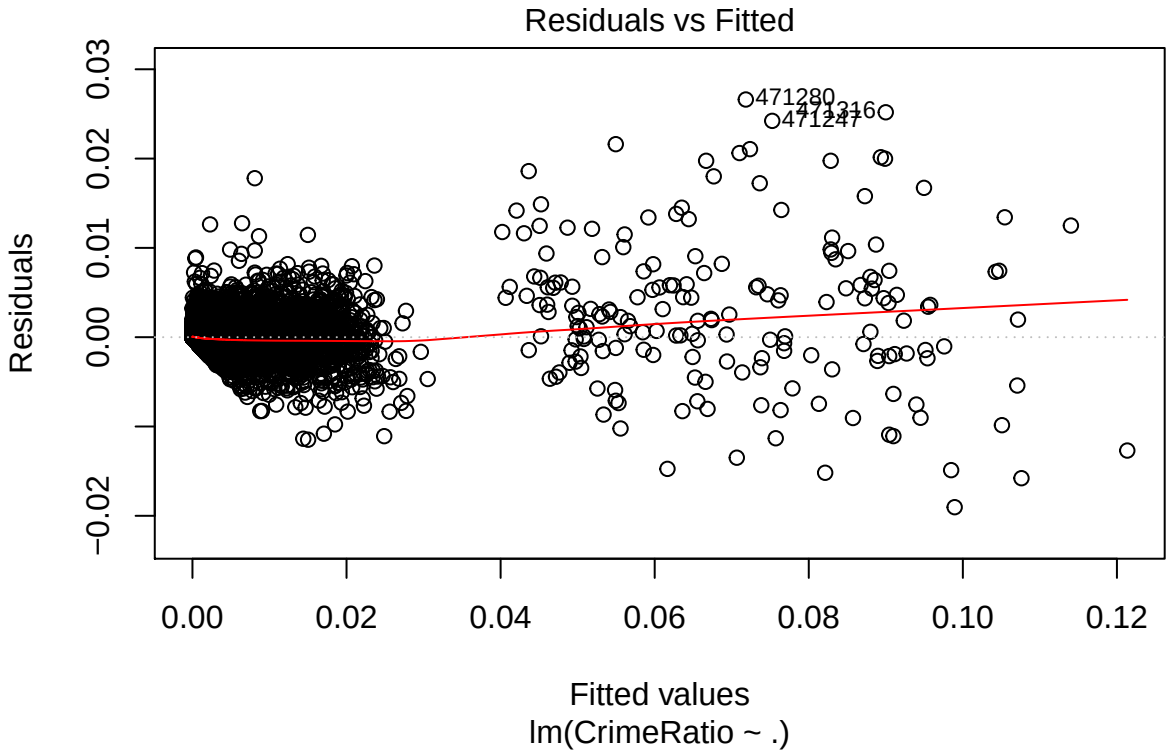
```
summary(model)
```

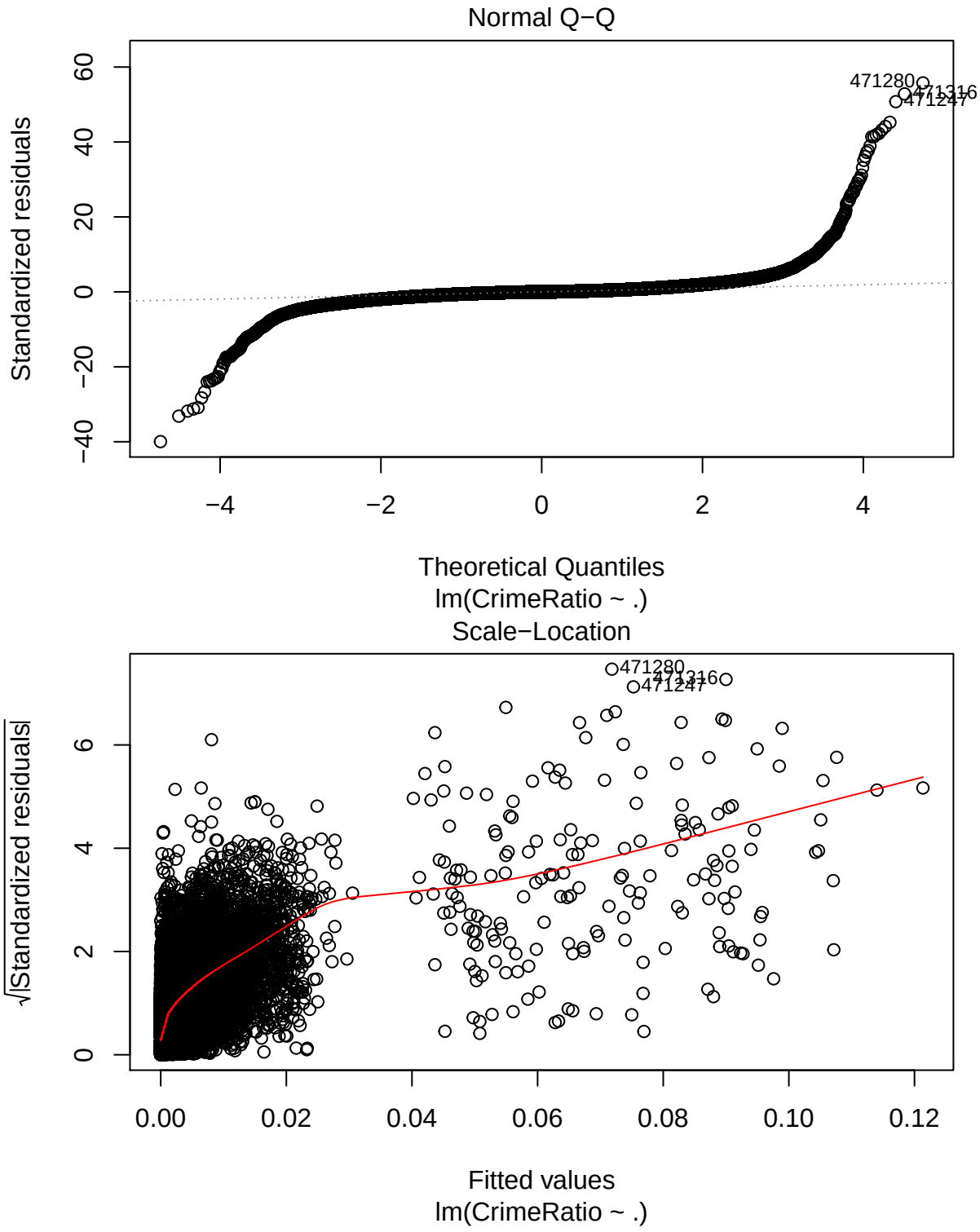
```
##
## Call:
## lm(formula = CrimeRatio ~ ., data = dataTrain)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0190337 -0.0001573  0.0000020  0.0001496  0.0266119
##
## Coefficients: (1 not defined because of singularities)
##                                          Estimate Std. Error  t value
## (Intercept)                             -2.592e-02  2.061e-02   -1.258
## WardCode                                 5.188e-09  4.122e-09    1.258
## IncomeScor                               3.052e-05  1.344e-05    2.271
## LivingEnSc                               8.054e-07  9.578e-08    8.409
## PopDen                                  -1.686e-09  7.850e-10   -2.147
## MajorCategory_Burglary                  -5.010e-05  2.985e-06  -16.785
## `MajorCategory_Criminal Damage`         -6.331e-05  2.998e-06  -21.117
## MajorCategory_Drugs                     -7.127e-05  3.014e-06  -23.645
## `MajorCategory_Fraud or Forgery`        -8.654e-05  3.050e-06  -28.378
## `MajorCategory_Other Notifiable Offences` -8.284e-05  3.046e-06  -27.194
## MajorCategory_Robbery                   -7.648e-05  3.031e-06  -25.234
## `MajorCategory_Sexual Offences`         -8.268e-05  3.047e-06  -27.134
## `MajorCategory_Theft and Handling`       3.795e-05  2.988e-06   12.700
## `MajorCategory_Violence Against The Person`         NA         NA       NA
## CrimeMonth                               3.052e-09  8.609e-10    3.545
## NoEmployeeRatio                          3.129e-09  1.038e-10   30.145
## NoDwellingRatio                         -4.840e-10  1.764e-09   -0.274
## NoOwndDwelNoRatio                       -2.933e-09  2.359e-09   -1.243
## NoCTFtoHNoRatio                         -2.055e-07  9.876e-08   -2.081
## PreviousCrimeRatio                       9.556e-01  4.366e-04 2188.532
##                                         Pr(>|t|)
## (Intercept)                             0.208550
## WardCode                                0.208259
## IncomeScor                              0.023142 *
## LivingEnSc                               < 2e-16 ***
## PopDen                                  0.031779 *
## MajorCategory_Burglary                   < 2e-16 ***
## `MajorCategory_Criminal Damage`          < 2e-16 ***
## MajorCategory_Drugs                      < 2e-16 ***
## `MajorCategory_Fraud or Forgery`         < 2e-16 ***
## `MajorCategory_Other Notifiable Offences`  < 2e-16 ***
## MajorCategory_Robbery                    < 2e-16 ***
## `MajorCategory_Sexual Offences`          < 2e-16 ***
```

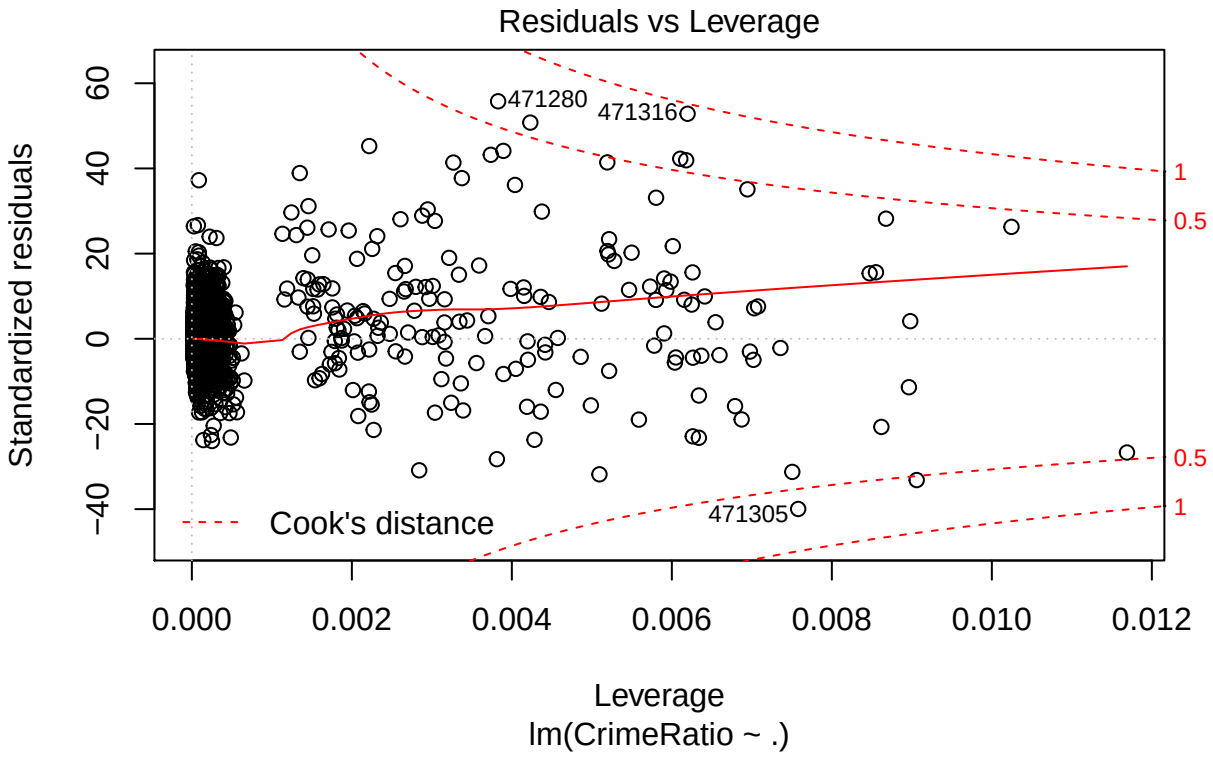
```
## `MajorCategory_Theft and Handling`           < 2e-16 ***
## `MajorCategory_Violence Against The Person`       NA
## CrimeMonth                                  0.000393 ***
## NoEmployeeRatio                              < 2e-16 ***
## NoDwellingRatio                             0.783821
## NoOwndDwelNoRatio                           0.213801
## NoCTFtoHNoRatio                             0.037463 *
## PreviousCrimeRatio                           < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0004781 on 471941 degrees of freedom
## Multiple R-squared:  0.9393, Adjusted R-squared:  0.9393
## F-statistic: 4.055e+05 on 18 and 471941 DF,  p-value: < 2.2e-16
```

ploting results

```
plot(model)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(CrimeRatio ~ .)
471280
471316
471247
Scale−Location
√|Standardized residuals|
Fitted values
lm(CrimeRatio ~ .)



prediction

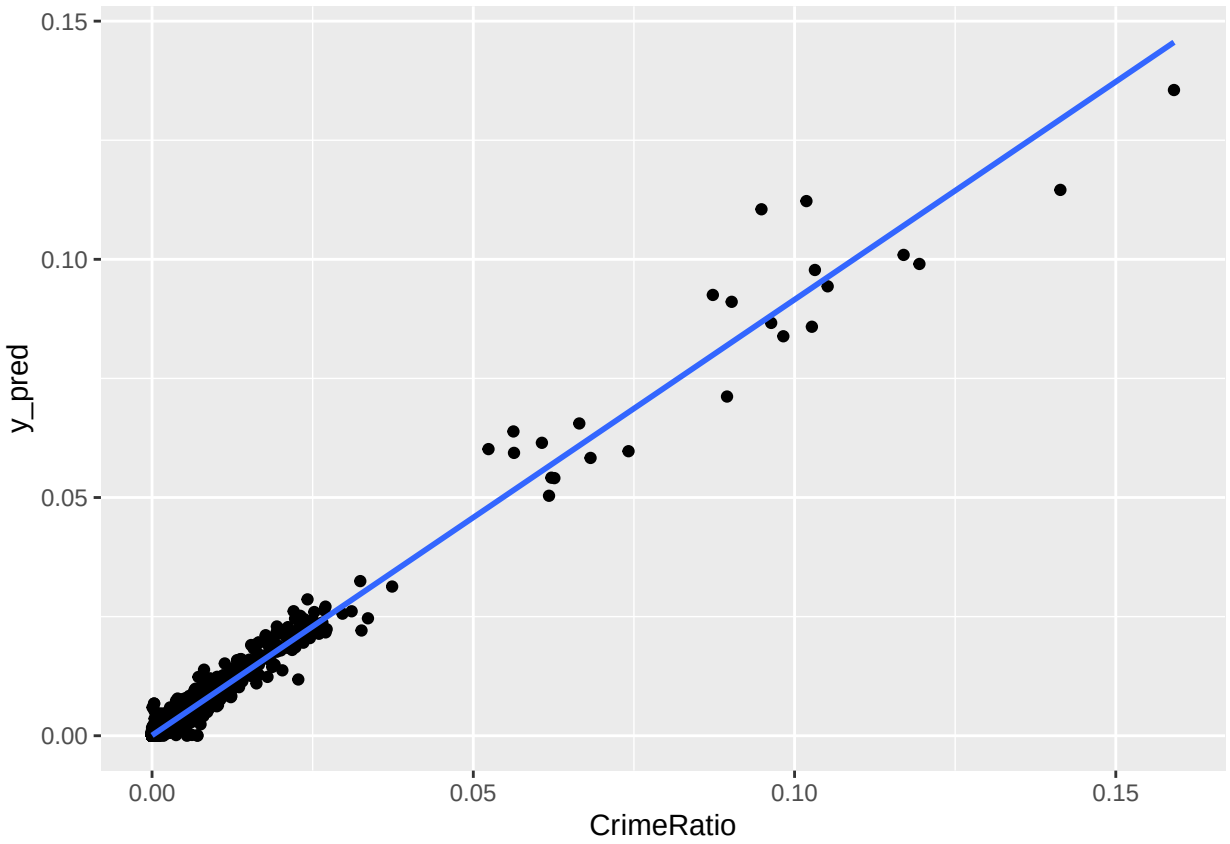
```
dataTest[, y_pred := predict(model, dataTest)]
```

```
## Warning in predict.lm(model, dataTest): prediction from a rank-deficient
## fit may be misleading
```

```
head(dataTest[, c('CrimeRatio', 'y_pred')])
```

```
##     CrimeRatio      y_pred
## 1: 0.004536211 0.005265678
## 2: 0.005005475 0.004443661
## 3: 0.005552948 0.004892171
## 4: 0.006100422 0.005415428
## 5: 0.004692633 0.005938681
## 6: 0.004770843 0.004593502
```

```
library(ggplot2)
ggplot(dataTest, aes(x=CrimeRatio, y=y_pred)) + geom_point() + geom_smooth(method = "lm")
```




```
actual = dataTest[, CrimeRatio]
preds = dataTest[, y_pred]
rss = sum((preds - actual) ^ 2)
tss = sum((actual - mean(actual)) ^ 2)
rsq = 1 - rss/tss
print(paste("R2 train: ", summary(model)$r.squared))
```

```
## [1] "R2 train:  0.939269396288181"
```

```
print(paste("R2 test: ", rsq))
```

```
## [1] "R2 test:  0.950248004806369"
```

Model is able to explain around 95% of dependent variable. Model also has statistical significance. So our hypothese1 and hypothese2 is true - features are statistically significant, but hypothese3 is not good especially “NoDwellingRatio” and “NoOwndDwelNoRatio” which can be removed from the model. Residuals are randomly distributed, have mean zero and have “bell” shape. Next step will be to run model without using third hypotese and see if this improve model.