Communities and Crime Rate

Brendan Dagys

6/27/2018

Data Import and Cleaning

```
library(corrplot)
library(caret)
library(rpart); library(party)
library(randomForest)
library(sqldf)
library(dplyr)
library(e1071)
library(neuralnet)
```

Column names to make the data frame more readable:

```
columns = c('state', 'county', 'community_int', 'community', 'fold',
'population', 'household_size', 'pct_black', 'pct_white',
            'pct_asian', 'pct_hispanic', 'age_12-21', 'age_12-29', 'age_16-
24', 'age_65+', 'num_urban', 'pct_urban', 'med_income',
            'pct_with_wage', 'pct_with_farm', 'pct_with_invest',
'pct_with_ss', 'pct_with_pub_assist', 'pct_with_retire_inc',
            'med_family_inc', 'per_cap_inc', 'white_per_cap_inc',
'black_per_cap_inc', 'indian_per_cap_inc', 'asian_per_cap_inc',
            'other_per_cap_inc', 'hisp_per_cap_inc', 'num_under_pov',
'pct_pop_under_pov', 'pct_less_9th_gr', 'pct_no_hs', 'pct_with_bach',
            'pct_unemp', 'pct_employ', 'pct_employ_manuf', 'pct_employ_prof',
'pct_employ_manuf', 'pct_employ_mgmt', 'pct_males_div',
            'pct_male_never_marr', 'pct_fem_div', 'pct_pop_div',
'ppl_per_family', 'pct_fam_2_parents', 'pct_kids_2_parents',
            'pct_kids<4_2_parents', 'pct_teens_2_parents',
'pct_work_mom_young_kids', 'pct_work_mom_kids', 'num_kids_to_unmarried',
            'pct_kids_to_unmarried', 'num_foreign_born', 'pct_immig_3_years',
'pct_immig_5_years', 'pct_immig_8_years', 'pct_immig_10_years',
            'pct_pop_immig_3_years', 'pct_pop_immig_5_years',
'pct_pop_immig_8_years', 'pct_pop_immig_10_years', 'pct_only_english',
            'pct_ESL', 'pct_large_household', 'pct_large_household',
'ppl_per_house', 'ppl_per_owner_occ_house', 'ppl_per_rented_house',
            'pct_ppl_in_owned_house', 'pct_ppl_dense_housing',
'pct_houses_less_3_bedrooms', 'med_num_bedrooms', 'num_vacant_households',
            'pct_houses_occ', 'pct_houses_owner_occ', 'pct_vacant_boarded',
'pct_vacant>6months', 'med_year_houses_built', 'pct_houses_no_phone',
            'pct_houses_no_plumb', 'owner_occ_low_quartile',
'owner_occ_med_quartile', 'owner_occ_high_quartile', 'rental_low_quartile',
```

```
            'rental_med_quartile', 'rental_high_quartile', 'med_rent',
'med_rent/income', 'med_owner_cost/income',
'med_owner_cost/income_no_mortgage',
            'num_in_shelters', 'num_homeless', 'pct_foreign_born',
'pct_born_same_state', 'pct_same_city_5_years', 'pct_same_city_5_years',
            'pct_same_state_5_years', 'full_time_cops',
'full_time_cops/100k', 'cops_in_field_ops',
'cops_in_field_ops/100k','tot_requests_for_police',
            'total_requests_for_police/100k',
'total_requests_for_police/officer', 'cops/100k', 'racial_match_pop_cops',
'pct_cops_white',
            'pct_cops_black', 'pct_cops_hisp', 'pct_cops_asian',
'pct_cops_minority', 'cops_drug_unit', 'num_kinds_drugs_seized',
'cops_avg_OT',
            'land_area_miles^2', 'ppl/mile^2', 'pct_ppl_use_transit_commute',
'num_police_cars', 'police_budget', 'pct_sworn_cops',
            'gang_unit', 'pct_cops_assigned_drug_unit', 'cop_budget_per_pop',
'violent_crime_100k')
```

Setting the working directory and loading the .txt file:

```
setwd('/')

crime = read.table('/Users/brendan/Desktop/Personal R Projects/Communities
and Crime Rate/communities.txt', sep = ',', na.strings = c('?', ''),
col.names = columns)
```

'fold' is non-predictive and 'pct_employ_manuf' is duplicated. There is also no description of the state numbers. We will remove these three columns:

```
crime = crime[, -c(1, 5, 42)]
```

Checking the structure: 1, 2 are 'int', 3 is 'factor'. Everything else is numeric.

```
str(crime)

## 'data.frame':    1994 obs. of  125 variables:
##  $ county                       : int  NA NA NA 5 95 NA 7 NA NA NA ...
##  $ community_int                : int  NA NA NA 81440 6096 NA 41500 NA
NA NA ...
##  $ community                    : Factor w/ 1828 levels
"Aberdeencity",..: 796 1626 2 1788 142 1520 840 1462 669 288 ...
##  $ population                   : num  0.19 0 0 0.04 0.01 0.02 0.01
0.01 0.03 0.01 ...
##  $ household_size               : num  0.33 0.16 0.42 0.77 0.55 0.28
0.39 0.74 0.34 0.4 ...
##  $ pct_black                    : num  0.02 0.12 0.49 1 0.02 0.06 0
0.03 0.2 0.06 ...
##  $ pct_white                    : num  0.9 0.74 0.56 0.08 0.95 0.54
0.98 0.46 0.84 0.87 ...
##  $ pct_asian                    : num  0.12 0.45 0.17 0.12 0.09 1 0.06
```

```
0.2 0.02 0.3 ...
##  $ pct_hispanic                     : num  0.17 0.07 0.04 0.1 0.05 0.25
0.02 1 0 0.03 ...
##  $ age_12.21                        : num  0.34 0.26 0.39 0.51 0.38 0.31
0.3 0.52 0.38 0.9 ...
##  $ age_12.29                        : num  0.47 0.59 0.47 0.5 0.38 0.48
0.37 0.55 0.45 0.82 ...
##  $ age_16.24                        : num  0.29 0.35 0.28 0.34 0.23 0.27
0.23 0.36 0.28 0.8 ...
##  $ age_65.                          : num  0.32 0.27 0.32 0.21 0.36 0.37
0.6 0.35 0.48 0.39 ...
##  $ num_urban                        : num  0.2 0.02 0 0.06 0.02 0.04 0.02
0 0.04 0.02 ...
##  $ pct_urban                        : num  1 1 0 1 0.9 1 0.81 0 1 1 ...
##  $ med_income                       : num  0.37 0.31 0.3 0.58 0.5 0.52
0.42 0.16 0.17 0.54 ...
##  $ pct_with_wage                    : num  0.72 0.72 0.58 0.89 0.72 0.68
0.5 0.44 0.47 0.59 ...
##  $ pct_with_farm                    : num  0.34 0.11 0.19 0.21 0.16 0.2
0.23 1 0.36 0.22 ...
##  $ pct_with_invest                  : num  0.6 0.45 0.39 0.43 0.68 0.61
0.68 0.23 0.34 0.86 ...
##  $ pct_with_ss                      : num  0.29 0.25 0.38 0.36 0.44 0.28
0.61 0.53 0.55 0.42 ...
##  $ pct_with_pub_assist              : num  0.15 0.29 0.4 0.2 0.11 0.15
0.21 0.97 0.48 0.02 ...
##  $ pct_with_retire_inc              : num  0.43 0.39 0.84 0.82 0.71 0.25
0.54 0.41 0.43 0.31 ...
##  $ med_family_inc                   : num  0.39 0.29 0.28 0.51 0.46 0.62
0.43 0.15 0.21 0.85 ...
##  $ per_cap_inc                      : num  0.4 0.37 0.27 0.36 0.43 0.72
0.47 0.1 0.23 0.89 ...
##  $ white_per_cap_inc                : num  0.39 0.38 0.29 0.4 0.41 0.76
0.44 0.12 0.23 0.94 ...
##  $ black_per_cap_inc                : num  0.32 0.33 0.27 0.39 0.28 0.77
0.4 0.08 0.19 0.11 ...
##  $ indian_per_cap_inc               : num  0.27 0.16 0.07 0.16 0 0.28 0.24
0.17 0.1 0.09 ...
##  $ asian_per_cap_inc                : num  0.27 0.3 0.29 0.25 0.74 0.52
0.86 0.27 0.26 0.33 ...
##  $ other_per_cap_inc                : num  0.36 0.22 0.28 0.36 0.51 0.48
0.24 0.18 0.29 0.17 ...
##  $ hisp_per_cap_inc                 : num  0.41 0.35 0.39 0.44 0.48 0.6
0.36 0.21 0.22 0.8 ...
##  $ num_under_pov                    : num  0.08 0.01 0.01 0.01 0 0.01 0.01
0.03 0.04 0 ...
##  $ pct_pop_under_pov                : num  0.19 0.24 0.27 0.1 0.06 0.12
0.11 0.64 0.45 0.11 ...
##  $ pct_less_9th_gr                  : num  0.1 0.14 0.27 0.09 0.25 0.13
0.29 0.96 0.52 0.04 ...
```

```
##  $ pct_no_hs                       : num  0.18 0.24 0.43 0.25 0.3 0.12 
0.41 0.82 0.59 0.03 ...
##  $ pct_with_bach                   : num  0.48 0.3 0.19 0.31 0.33 0.8 
0.36 0.12 0.17 1 ...
##  $ pct_unemp                       : num  0.27 0.27 0.36 0.33 0.12 0.1 
0.28 1 0.55 0.11 ...
##  $ pct_employ                      : num  0.68 0.73 0.58 0.71 0.65 0.65 
0.54 0.26 0.43 0.44 ...
##  $ pct_employ_manuf                : num  0.23 0.57 0.32 0.36 0.67 0.19 
0.44 0.43 0.59 0.2 ...
##  $ pct_employ_prof                 : num  0.41 0.15 0.29 0.45 0.38 0.77 
0.53 0.34 0.36 1 ...
##  $ pct_employ_mgmt                 : num  0.52 0.36 0.32 0.39 0.46 0.91 
0.49 0.18 0.29 0.96 ...
##  $ pct_males_div                   : num  0.68 1 0.63 0.34 0.22 0.49 0.25 
0.38 0.62 0.3 ...
##  $ pct_male_never_marr             : num  0.4 0.63 0.41 0.45 0.27 0.57 
0.34 0.47 0.26 0.85 ...
##  $ pct_fem_div                     : num  0.75 0.91 0.71 0.49 0.2 0.61 
0.28 0.59 0.66 0.39 ...
##  $ pct_pop_div                     : num  0.75 1 0.7 0.44 0.21 0.58 0.28 
0.52 0.67 0.36 ...
##  $ ppl_per_family                  : num  0.35 0.29 0.45 0.75 0.51 0.44 
0.42 0.78 0.37 0.31 ...
##  $ pct_fam_2_parents               : num  0.55 0.43 0.42 0.65 0.91 0.62 
0.77 0.45 0.51 0.65 ...
##  $ pct_kids_2_parents              : num  0.59 0.47 0.44 0.54 0.91 0.69 
0.81 0.43 0.55 0.73 ...
##  $ pct_kids.4_2_parents            : num  0.61 0.6 0.43 0.83 0.89 0.87 
0.79 0.34 0.58 0.78 ...
##  $ pct_teens_2_parents             : num  0.56 0.39 0.43 0.65 0.85 0.53 
0.74 0.34 0.47 0.67 ...
##  $ pct_work_mom_young_kids         : num  0.74 0.46 0.71 0.85 0.4 0.3 
0.57 0.29 0.65 0.72 ...
##  $ pct_work_mom_kids               : num  0.76 0.53 0.67 0.86 0.6 0.43 
0.62 0.27 0.64 0.71 ...
##  $ num_kids_to_unmarried           : num  0.04 0 0.01 0.03 0 0 0 0.02 
0.02 0 ...
##  $ pct_kids_to_unmarried           : num  0.14 0.24 0.46 0.33 0.06 0.11 
0.13 0.5 0.29 0.07 ...
##  $ num_foreign_born                : num  0.03 0.01 0 0.02 0 0.04 0.01 
0.02 0 0.01 ...
##  $ pct_immig_3_years               : num  0.24 0.52 0.07 0.11 0.03 0.3 0 
0.5 0.12 0.41 ...
##  $ pct_immig_5_years               : num  0.27 0.62 0.06 0.2 0.07 0.35 
0.02 0.59 0.09 0.44 ...
##  $ pct_immig_8_years               : num  0.37 0.64 0.15 0.3 0.2 0.43 
0.02 0.65 0.07 0.52 ...
##  $ pct_immig_10_years              : num  0.39 0.63 0.19 0.31 0.27 0.47 
0.1 0.59 0.13 0.48 ...
```

```
##  $ pct_pop_immig_3_years          : num  0.07 0.25 0.02 0.05 0.01 0.5 0
0.69 0 0.22 ...
##  $ pct_pop_immig_5_years          : num  0.07 0.27 0.02 0.08 0.02 0.5
0.01 0.72 0 0.21 ...
##  $ pct_pop_immig_8_years          : num  0.08 0.25 0.04 0.11 0.04 0.56
0.01 0.71 0 0.22 ...
##  $ pct_pop_immig_10_years         : num  0.08 0.23 0.05 0.11 0.05 0.57
0.03 0.6 0 0.19 ...
##  $ pct_only_english               : num  0.89 0.84 0.88 0.81 0.88 0.45
0.73 0.12 0.99 0.85 ...
##  $ pct_ESL                        : num  0.06 0.1 0.04 0.08 0.05 0.28
0.05 0.93 0.01 0.03 ...
##  $ pct_large_household            : num  0.14 0.16 0.2 0.56 0.16 0.25
0.12 0.74 0.12 0.09 ...
##  $ pct_large_household.1          : num  0.13 0.1 0.2 0.62 0.19 0.19
0.13 0.75 0.12 0.06 ...
##  $ ppl_per_house                  : num  0.33 0.17 0.46 0.85 0.59 0.29
0.42 0.8 0.35 0.15 ...
##  $ ppl_per_owner_occ_house        : num  0.39 0.29 0.52 0.77 0.6 0.53
0.54 0.68 0.38 0.34 ...
##  $ ppl_per_rented_house           : num  0.28 0.17 0.43 1 0.37 0.18 0.24
0.92 0.33 0.05 ...
##  $ pct_ppl_in_owned_house         : num  0.55 0.26 0.42 0.94 0.89 0.39
0.65 0.39 0.5 0.48 ...
##  $ pct_ppl_dense_housing          : num  0.09 0.2 0.15 0.12 0.02 0.26
0.03 0.89 0.1 0.03 ...
##  $ pct_houses_less_3_bedrooms     : num  0.51 0.82 0.51 0.01 0.19 0.73
0.46 0.66 0.64 0.58 ...
##  $ med_num_bedrooms               : num  0.5 0 0.5 0.5 0.5 0 0.5 0 0 0
...
##  $ num_vacant_households          : num  0.21 0.02 0.01 0.01 0.01 0.02
0.01 0.01 0.04 0.02 ...
##  $ pct_houses_occ                 : num  0.71 0.79 0.86 0.97 0.89 0.84
0.89 0.91 0.72 0.72 ...
##  $ pct_houses_owner_occ           : num  0.52 0.24 0.41 0.96 0.87 0.3
0.57 0.46 0.49 0.38 ...
##  $ pct_vacant_boarded             : num  0.05 0.02 0.29 0.6 0.04 0.16
0.09 0.22 0.05 0.07 ...
##  $ pct_vacant.6months             : num  0.26 0.25 0.3 0.47 0.55 0.28
0.49 0.37 0.49 0.47 ...
##  $ med_year_houses_built          : num  0.65 0.65 0.52 0.52 0.73 0.25
0.38 0.6 0.5 0.04 ...
##  $ pct_houses_no_phone            : num  0.14 0.16 0.47 0.11 0.05 0.02
0.05 0.28 0.57 0.01 ...
##  $ pct_houses_no_plumb            : num  0.06 0 0.45 0.11 0.14 0.05 0.05
0.23 0.22 0 ...
##  $ owner_occ_low_quartile         : num  0.22 0.21 0.18 0.24 0.31 0.94
0.37 0.15 0.07 0.63 ...
##  $ owner_occ_med_quartile         : num  0.19 0.2 0.17 0.21 0.31 1 0.38
0.13 0.07 0.71 ...
```

```
##  $ owner_occ_high_quartile          : num  0.18 0.21 0.16 0.19 0.3 1 0.39
0.13 0.08 0.79 ...
##  $ rental_low_quartile              : num  0.36 0.42 0.27 0.75 0.4 0.67
0.26 0.21 0.14 0.44 ...
##  $ rental_med_quartile              : num  0.35 0.38 0.29 0.7 0.36 0.63
0.35 0.24 0.17 0.42 ...
##  $ rental_high_quartile             : num  0.38 0.4 0.27 0.77 0.38 0.68
0.42 0.25 0.16 0.47 ...
##  $ med_rent                         : num  0.34 0.37 0.31 0.89 0.38 0.62
0.35 0.24 0.15 0.41 ...
##  $ med_rent.income                  : num  0.38 0.29 0.48 0.63 0.22 0.47
0.46 0.64 0.38 0.23 ...
##  $ med_owner_cost.income            : num  0.46 0.32 0.39 0.51 0.51 0.59
0.44 0.59 0.13 0.27 ...
##  $ med_owner_cost.income_no_mortgage: num  0.25 0.18 0.28 0.47 0.21 0.11
0.31 0.28 0.36 0.28 ...
##  $ num_in_shelters                  : num  0.04 0 0 0 0 0 0 0 0.01 0 ...
##  $ num_homeless                     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ pct_foreign_born                 : num  0.12 0.21 0.14 0.19 0.11 0.7
0.15 0.59 0.01 0.22 ...
##  $ pct_born_same_state              : num  0.42 0.5 0.49 0.3 0.72 0.42
0.81 0.58 0.78 0.42 ...
##  $ pct_same_city_5_years            : num  0.5 0.34 0.54 0.73 0.64 0.49
0.77 0.52 0.48 0.34 ...
##  $ pct_same_city_5_years.1          : num  0.51 0.6 0.67 0.64 0.61 0.73
0.91 0.79 0.79 0.23 ...
##  $ pct_same_state_5_years           : num  0.64 0.52 0.56 0.65 0.53 0.64
0.84 0.78 0.75 0.09 ...
##  $ full_time_cops                   : num  0.03 NA NA NA NA NA NA NA NA NA
...
##   [list output truncated]
```

There are 1675 rows with missing values, but only 1994 rows. We therefore can't delete observations, but we also can't impute the mean.

```
sum(complete.cases(crime)) # 123 complete cases. We'll have to remove the
columns.

## [1] 123

sapply(crime, function (x) sum(is.na(x)))

##                               county                      community_int
##                                 1174                               1177
##                            community                         population
##                                    0                                  0
##                       household_size                          pct_black
##                                    0                                  0
##                            pct_white                          pct_asian
##                                    0                                  0
##                         pct_hispanic                          age_12.21
```

```
##                            0                            0
##                    age_12.29                    age_16.24
##                            0                            0
##                      age_65.                    num_urban
##                            0                            0
##                    pct_urban                   med_income
##                            0                            0
##                pct_with_wage                pct_with_farm
##                            0                            0
##              pct_with_invest                  pct_with_ss
##                            0                            0
##          pct_with_pub_assist          pct_with_retire_inc
##                            0                            0
##               med_family_inc                  per_cap_inc
##                            0                            0
##            white_per_cap_inc            black_per_cap_inc
##                            0                            0
##           indian_per_cap_inc            asian_per_cap_inc
##                            0                            0
##            other_per_cap_inc             hisp_per_cap_inc
##                            1                            0
##                num_under_pov            pct_pop_under_pov
##                            0                            0
##              pct_less_9th_gr                    pct_no_hs
##                            0                            0
##                pct_with_bach                    pct_unemp
##                            0                            0
##                   pct_employ             pct_employ_manuf
##                            0                            0
##              pct_employ_prof              pct_employ_mgmt
##                            0                            0
##                pct_males_div          pct_male_never_marr
##                            0                            0
##                  pct_fem_div                  pct_pop_div
##                            0                            0
##               ppl_per_family            pct_fam_2_parents
##                            0                            0
##           pct_kids_2_parents         pct_kids.4_2_parents
##                            0                            0
##          pct_teens_2_parents     pct_work_mom_young_kids
##                            0                            0
##            pct_work_mom_kids         num_kids_to_unmarried
##                            0                            0
##        pct_kids_to_unmarried             num_foreign_born
##                            0                            0
##            pct_immig_3_years            pct_immig_5_years
##                            0                            0
##            pct_immig_8_years           pct_immig_10_years
##                            0                            0
##        pct_pop_immig_3_years        pct_pop_immig_5_years
```

```
##                                 0                                 0
##              pct_pop_immig_8_years             pct_pop_immig_10_years
##                                 0                                 0
##                  pct_only_english                           pct_ESL
##                                 0                                 0
##                pct_large_household             pct_large_household.1
##                                 0                                 0
##                     ppl_per_house             ppl_per_owner_occ_house
##                                 0                                 0
##               ppl_per_rented_house             pct_ppl_in_owned_house
##                                 0                                 0
##              pct_ppl_dense_housing         pct_houses_less_3_bedrooms
##                                 0                                 0
##                  med_num_bedrooms             num_vacant_households
##                                 0                                 0
##                    pct_houses_occ               pct_houses_owner_occ
##                                 0                                 0
##                pct_vacant_boarded                 pct_vacant.6months
##                                 0                                 0
##              med_year_houses_built                pct_houses_no_phone
##                                 0                                 0
##                pct_houses_no_plumb              owner_occ_low_quartile
##                                 0                                 0
##             owner_occ_med_quartile             owner_occ_high_quartile
##                                 0                                 0
##                rental_low_quartile                 rental_med_quartile
##                                 0                                 0
##               rental_high_quartile                           med_rent
##                                 0                                 0
##                   med_rent.income              med_owner_cost.income
##                                 0                                 0
## med_owner_cost.income_no_mortgage                    num_in_shelters
##                                 0                                 0
##                      num_homeless                   pct_foreign_born
##                                 0                                 0
##                pct_born_same_state              pct_same_city_5_years
##                                 0                                 0
##            pct_same_city_5_years.1             pct_same_state_5_years
##                                 0                                 0
##                    full_time_cops                full_time_cops.100k
##                              1675                              1675
##                 cops_in_field_ops             cops_in_field_ops.100k
##                              1675                              1675
##            tot_requests_for_police    total_requests_for_police.100k
##                              1675                              1675
## total_requests_for_police.officer                          cops.100k
##                              1675                              1675
##              racial_match_pop_cops                    pct_cops_white
##                              1675                              1675
##                    pct_cops_black                     pct_cops_hisp
```

```
##                               1675                              1675
##                     pct_cops_asian                 pct_cops_minority
##                               1675                              1675
##                     cops_drug_unit            num_kinds_drugs_seized
##                               1675                              1675
##                        cops_avg_OT                  land_area_miles.2
##                               1675                                 0
##                         ppl.mile.2       pct_ppl_use_transit_commute
##                                  0                                 0
##                    num_police_cars                     police_budget
##                               1675                              1675
##                     pct_sworn_cops                         gang_unit
##                               1675                              1675
##       pct_cops_assigned_drug_unit                 cop_budget_per_pop
##                                  0                              1675
##                  violent_crime_100k
##                                  0
```

Removing 25 columns:

```r
keep = sapply(crime, function (x) !any(is.na(x)))
crime = crime[, keep]
```

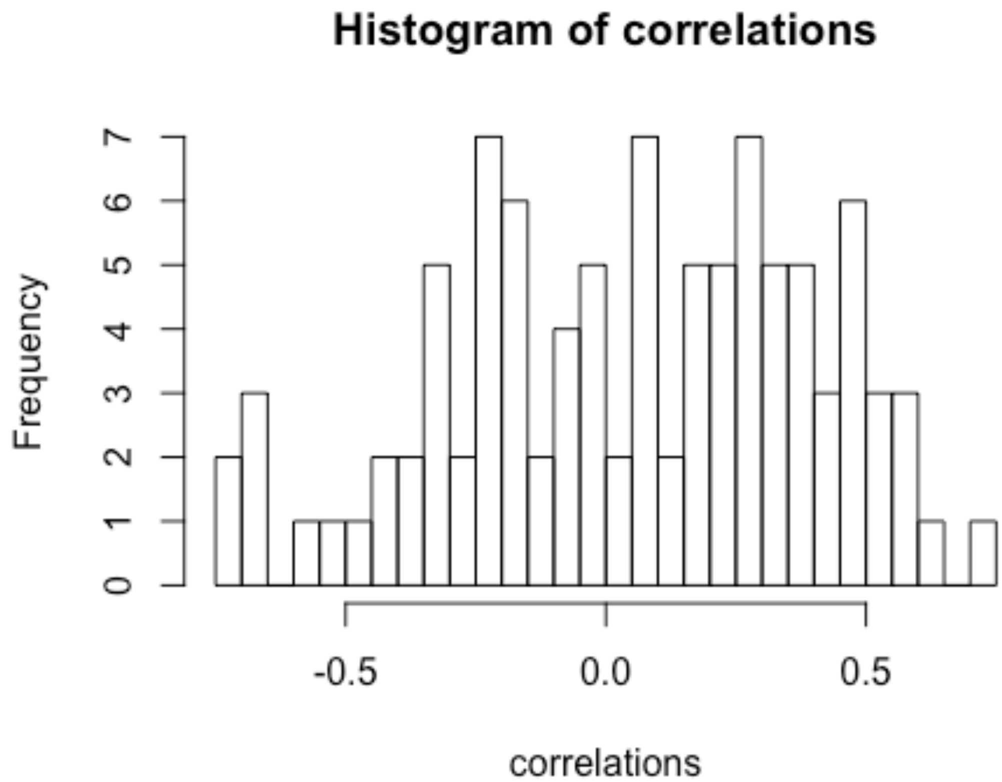
Now there are no missing values!

```r
sum(is.na(crime))

## [1] 0
```

Initial Exploration and Feature Selection

Excluding ‘community_name’ and ‘crime_per_100k’ for correlation calculation:

```r
correlations = cor(crime[-c(1, 100)], crime[100])
hist(correlations, breaks = 50)
```



To re-include ‘community_name’ and ‘crime_per_100k’ for the next step:

```
correlations = c(1, correlations, 1)
```

Only keeping columns that have a correlation with the class variable greater than 0.3:

```
crime_important = crime[, correlations > 0.3]
str(crime_important)

## 'data.frame':    1994 obs. of  29 variables:
##  $ community                  : Factor w/ 1828 levels "Aberdeencity",..: 
796 1626 2 1788 142 1520 840 1462 669 288 ...
##  $ population                 : num  0.19 0 0 0.04 0.01 0.02 0.01 0.01 
0.03 0.01 ...
##  $ pct_black                  : num  0.02 0.12 0.49 1 0.02 0.06 0 0.03 0.2 
0.06 ...
##  $ num_urban                  : num  0.2 0.02 0 0.06 0.02 0.04 0.02 0 0.04 
0.02 ...
##  $ pct_with_pub_assist        : num  0.15 0.29 0.4 0.2 0.11 0.15 0.21 0.97 
0.48 0.02 ...
##  $ num_under_pov              : num  0.08 0.01 0.01 0.01 0 0.01 0.01 0.03 
0.04 0 ...
##  $ pct_pop_under_pov          : num  0.19 0.24 0.27 0.1 0.06 0.12 0.11 
0.64 0.45 0.11 ...
```

```
##  $ pct_less_9th_gr           : num  0.1 0.14 0.27 0.09 0.25 0.13 0.29
0.96 0.52 0.04 ...
##  $ pct_no_hs                 : num  0.18 0.24 0.43 0.25 0.3 0.12 0.41
0.82 0.59 0.03 ...
##  $ pct_unemp                 : num  0.27 0.27 0.36 0.33 0.12 0.1 0.28 1
0.55 0.11 ...
##  $ pct_males_div             : num  0.68 1 0.63 0.34 0.22 0.49 0.25 0.38
0.62 0.3 ...
##  $ pct_male_never_marr       : num  0.4 0.63 0.41 0.45 0.27 0.57 0.34
0.47 0.26 0.85 ...
##  $ pct_fem_div               : num  0.75 0.91 0.71 0.49 0.2 0.61 0.28
0.59 0.66 0.39 ...
##  $ pct_pop_div               : num  0.75 1 0.7 0.44 0.21 0.58 0.28 0.52
0.67 0.36 ...
##  $ num_kids_to_unmarried     : num  0.04 0 0.01 0.03 0 0 0 0.02 0.02 0
...
##  $ pct_kids_to_unmarried     : num  0.14 0.24 0.46 0.33 0.06 0.11 0.13
0.5 0.29 0.07 ...
##  $ pct_ESL                   : num  0.06 0.1 0.04 0.08 0.05 0.28 0.05
0.93 0.01 0.03 ...
##  $ pct_large_household       : num  0.14 0.16 0.2 0.56 0.16 0.25 0.12
0.74 0.12 0.09 ...
##  $ pct_ppl_dense_housing     : num  0.09 0.2 0.15 0.12 0.02 0.26 0.03
0.89 0.1 0.03 ...
##  $ pct_houses_less_3_bedrooms : num  0.51 0.82 0.51 0.01 0.19 0.73 0.46
0.66 0.64 0.58 ...
##  $ num_vacant_households     : num  0.21 0.02 0.01 0.01 0.01 0.02 0.01
0.01 0.04 0.02 ...
##  $ pct_vacant_boarded        : num  0.05 0.02 0.29 0.6 0.04 0.16 0.09
0.22 0.05 0.07 ...
##  $ pct_houses_no_phone       : num  0.14 0.16 0.47 0.11 0.05 0.02 0.05
0.28 0.57 0.01 ...
##  $ pct_houses_no_plumb       : num  0.06 0 0.45 0.11 0.14 0.05 0.05 0.23
0.22 0 ...
##  $ med_rent.income           : num  0.38 0.29 0.48 0.63 0.22 0.47 0.46
0.64 0.38 0.23 ...
##  $ num_in_shelters           : num  0.04 0 0 0 0 0 0 0 0.01 0 ...
##  $ num_homeless              : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ pct_cops_assigned_drug_unit: num  0.32 0 0 0 0 0 0 0 0 0 ...
##  $ violent_crime_100k        : num  0.2 0.67 0.43 0.12 0.03 0.14 0.03
0.55 0.53 0.15 ...
```

Correlation of everything but factor variable with the class variable:

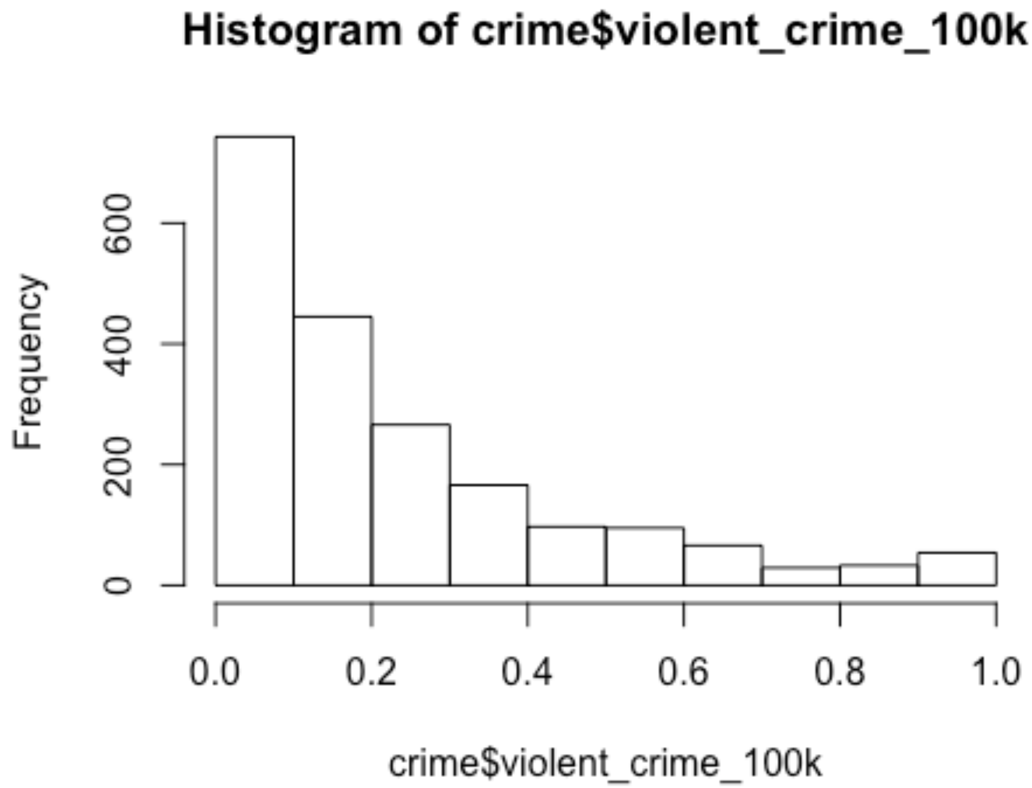
```r
new_correlations = cor(crime_important[-c(1, 29)], crime_important[29])

# corrplot(new_correlations) # Single row
# corrplot(cor(crime_important[-1])) # Matrix
```

Rename to ‘crime’ for simplicity and remove the only non-numeric column:

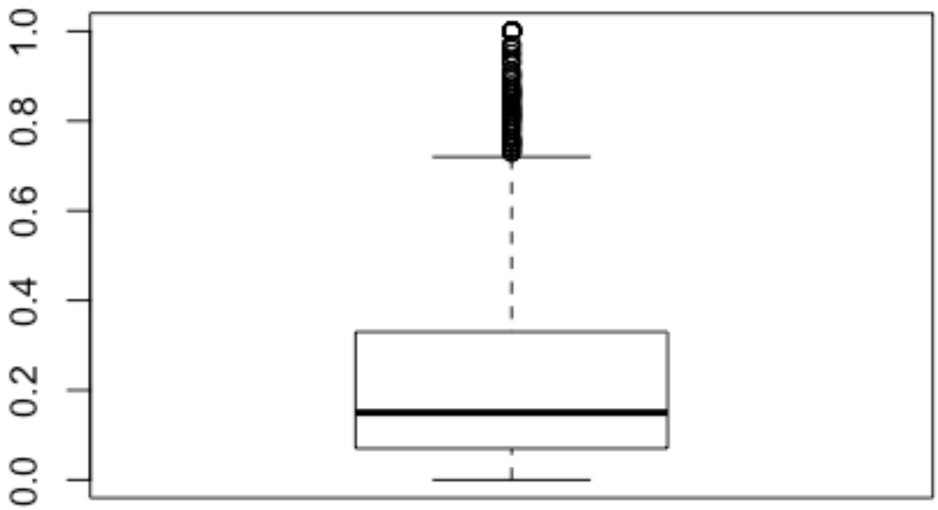
```r
crime = crime_important[-1]
cor(crime[-28], crime[28])
```

```
##                              violent_crime_100k
## population                            0.3671574
## pct_black                             0.6312636
## num_urban                             0.3628974
## pct_with_pub_assist                   0.5746653
## num_under_pov                         0.4475816
## pct_pop_under_pov                     0.5218765
## pct_less_9th_gr                       0.4110955
## pct_no_hs                             0.4833659
## pct_unemp                             0.5042346
## pct_males_div                         0.5254073
## pct_male_never_marr                   0.3045829
## pct_fem_div                           0.5560319
## pct_pop_div                           0.5527774
## num_kids_to_unmarried                 0.4710281
## pct_kids_to_unmarried                 0.7379565
## pct_ESL                               0.3000190
## pct_large_household                   0.3834797
## pct_ppl_dense_housing                 0.4529009
## pct_houses_less_3_bedrooms            0.4744899
## num_vacant_households                 0.4213958
## pct_vacant_boarded                    0.4828158
## pct_houses_no_phone                   0.4882435
## pct_houses_no_plumb                   0.3644539
## med_rent.income                       0.3250453
## num_in_shelters                       0.3757542
## num_homeless                          0.3402768
## pct_cops_assigned_drug_unit           0.3486273
```

```r
hist(crime$violent_crime_100k)
```




```
boxplot(crime$violent_crime_100k)
```



Partitioning

Using a 70% training set partition:

```
set.seed(7)
index = sample(nrow(crime), 0.7 * nrow(crime))
train = crime[index,]
test = crime[-index,]
train_labels = train[, 28] # for kNN
test_labels = test[, 28] # for kNN
```

Creating a function to predict RMSE:

```
my_rmse = function (predicted, actual) return(sqrt(mean((predicted -
actual)^2)))
```

Linear Regression

RMSE: 0.1329, R-squared: 0.6807, adjusted R-squared: 0.6744

```r
linear_model = lm(violent_crime_100k ~ ., data = train)
summary(linear_model)
```

```
##
## Call:
## lm(formula = violent_crime_100k ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56186 -0.07105 -0.01039  0.04780  0.79562
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -0.09484    0.02342  -4.049 5.43e-05 ***
## population                   -0.95838    0.28505  -3.362 0.000795 ***
## pct_black                     0.21486    0.03140   6.842 1.18e-11 ***
## num_urban                     0.68778    0.26657   2.580 0.009979 **
## pct_with_pub_assist           0.14494    0.03978   3.643 0.000279 ***
## num_under_pov                -0.06071    0.15265  -0.398 0.690889
## pct_pop_under_pov            -0.10119    0.04819  -2.100 0.035937 *
## pct_less_9th_gr              -0.23042    0.06636  -3.472 0.000532 ***
## pct_no_hs                     0.22043    0.07155   3.081 0.002105 **
## pct_unemp                    -0.05443    0.03864  -1.409 0.159149
## pct_males_div                 0.11998    0.24450   0.491 0.623721
## pct_male_never_marr           0.02441    0.03415   0.715 0.474921
## pct_fem_div                  -0.02868    0.30772  -0.093 0.925750
## pct_pop_div                   0.05244    0.51557   0.102 0.918998
## num_kids_to_unmarried        -0.05014    0.12226  -0.410 0.681787
## pct_kids_to_unmarried         0.25932    0.04625   5.607 2.48e-08 ***
## pct_ESL                       0.03479    0.04572   0.761 0.446805
## pct_large_household          -0.02873    0.04462  -0.644 0.519806
## pct_ppl_dense_housing         0.22790    0.06139   3.712 0.000214 ***
## pct_houses_less_3_bedrooms    0.02609    0.03777   0.691 0.489857
## num_vacant_households         0.34312    0.06331   5.420 7.03e-08 ***
## pct_vacant_boarded            0.04908    0.02326   2.110 0.035066 *
## pct_houses_no_phone           0.08785    0.03732   2.354 0.018705 *
## pct_houses_no_plumb          -0.02679    0.02272  -1.179 0.238473
## med_rent.income               0.07919    0.02738   2.892 0.003883 **
## num_in_shelters               0.18389    0.07261   2.532 0.011439 *
## num_homeless                  0.18707    0.05457   3.428 0.000625 ***
## pct_cops_assigned_drug_unit   0.03417    0.01862   1.835 0.066724 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1329 on 1367 degrees of freedom
## Multiple R-squared:  0.6807, Adjusted R-squared:  0.6744
## F-statistic: 107.9 on 27 and 1367 DF,  p-value: < 2.2e-16
```

Most important features are:

‘pct_black’ ‘pct_kids_to_unmarried’ ‘num_vacant_households’ ‘pct_ppl_dense_housing’ ‘pct_with_pub_assist’ ‘num_homeless’ ‘pct_less_9th_grade’ ‘population’ ‘pct_no_hs’

```
varImp(linear_model)
```

```
##                               Overall
## population                 3.36214484
## pct_black                  6.84174206
## num_urban                  2.58016893
## pct_with_pub_assist        3.64340904
## num_under_pov              0.39773320
## pct_pop_under_pov          2.09971032
## pct_less_9th_gr            3.47216704
## pct_no_hs                  3.08084683
## pct_unemp                  1.40870774
## pct_males_div              0.49069496
## pct_male_never_marr        0.71469321
## pct_fem_div                0.09321101
## pct_pop_div                0.10171488
## num_kids_to_unmarried      0.41011333
## pct_kids_to_unmarried      5.60748199
## pct_ESL                    0.76097163
## pct_large_household        0.64381041
## pct_ppl_dense_housing      3.71215211
## pct_houses_less_3_bedrooms 0.69072332
## num_vacant_households      5.42007242
## pct_vacant_boarded         2.10969004
## pct_houses_no_phone        2.35416012
## pct_houses_no_plumb        1.17932648
## med_rent.income            2.89245733
## num_in_shelters            2.53242986
## num_homeless               3.42841228
## pct_cops_assigned_drug_unit 1.83499560
```

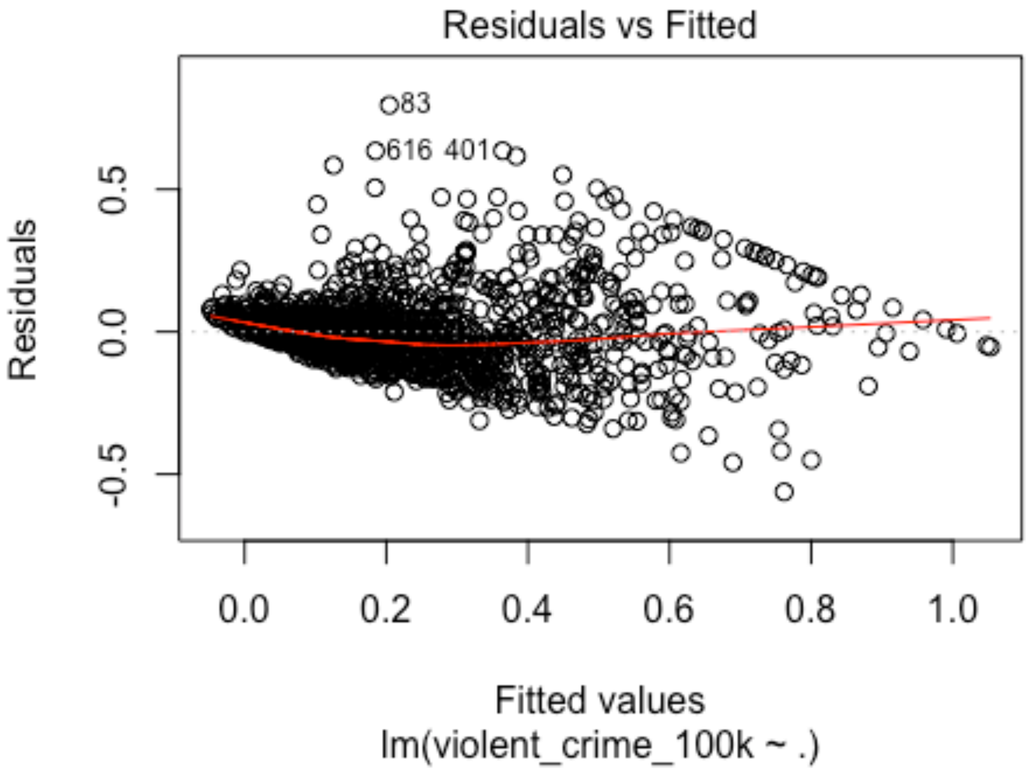
```
linear_pred = predict(linear_model, test)
```

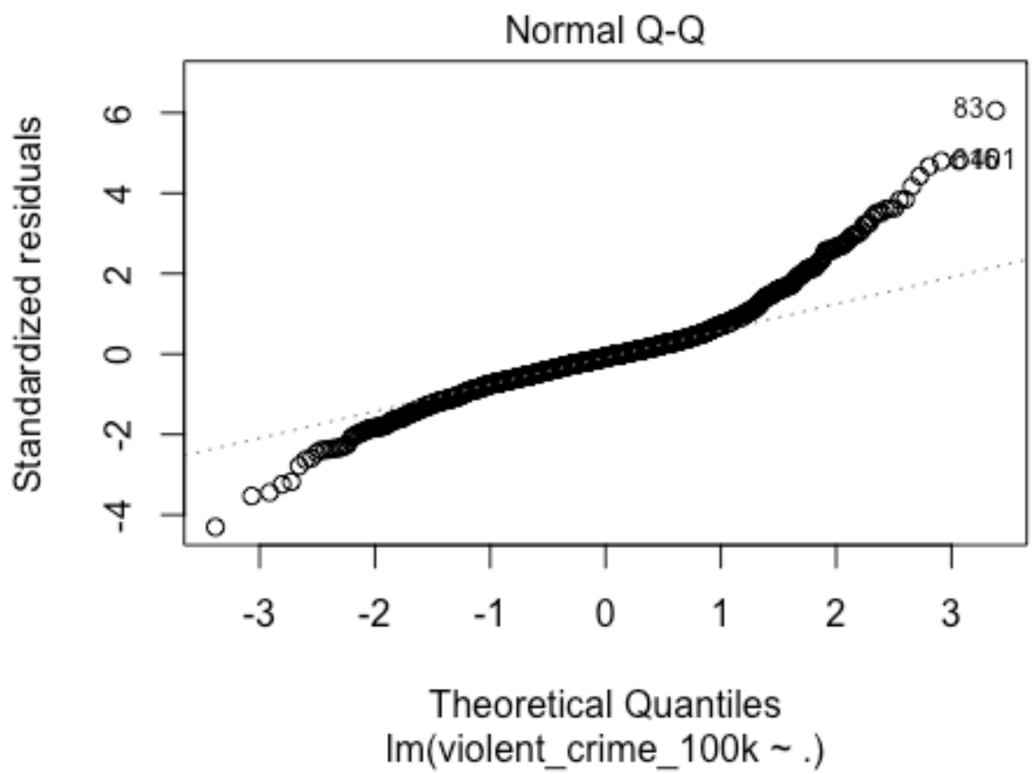
RMSE: 0.1479

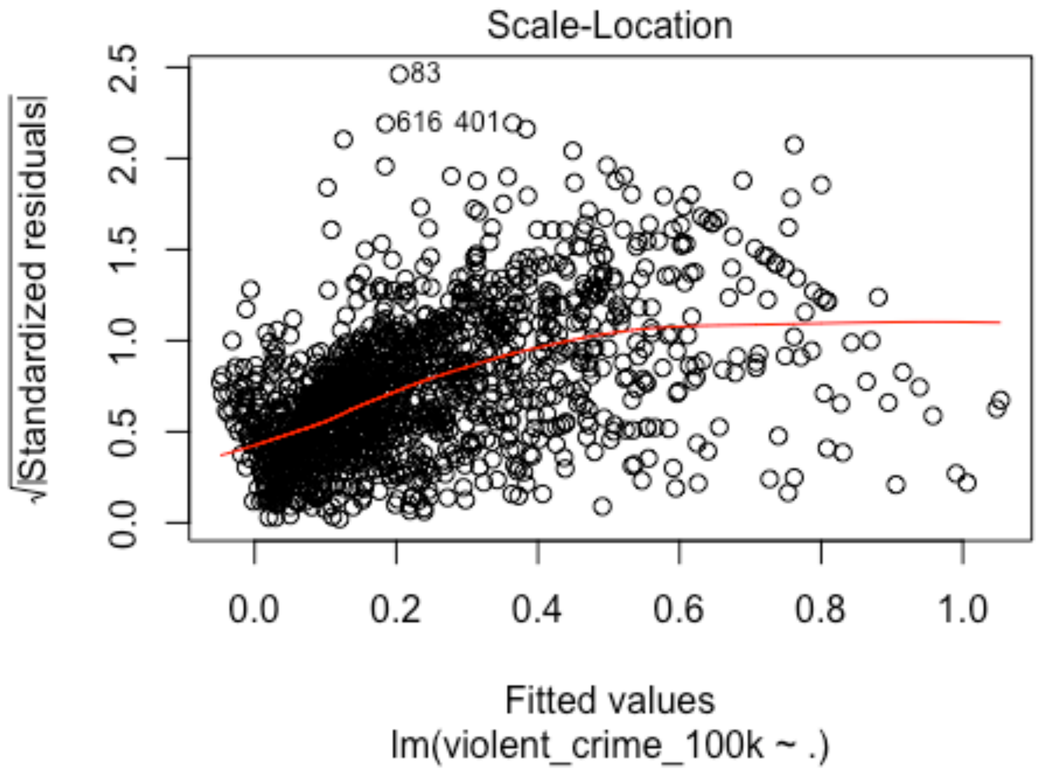
```
my_rmse(linear_pred, test$violent_crime_100k)
```

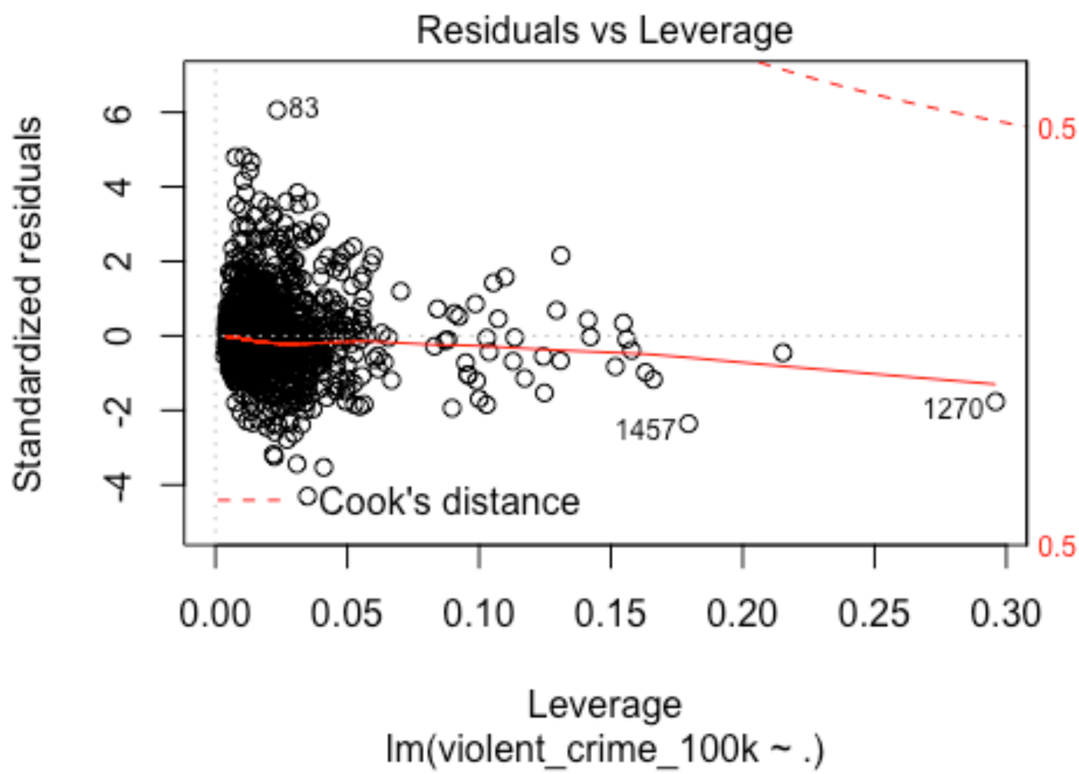
```
## [1] 0.1479097
```

```
plot(linear_model)
```


Residuals vs Fitted
83
616
401
Residuals
0.5
0.0
-0.5
0.0
0.2
0.4
0.6
0.8
1.0
Fitted values
lm(violent_crime_100k ~ .)


Normal Q-Q
Standardized residuals
-4
-2
0
2
4
6
83
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(violent_crime_100k ~ .)


Scale-Location
83
616
401
$\sqrt{|\text{Standardized residuals}|}$
0.0
0.5
1.0
1.5
2.0
2.5
0.0
0.2
0.4
0.6
0.8
1.0
Fitted values
lm(violent_crime_100k ~ .)



KNN Regression

```
knn_model = knnregTrain(train[-28], test[-28], train_labels, 10)
```

RMSE: 0.1519

```
my_rmse(knn_model, test_labels)

## [1] 0.1519897
```

Decision Tree

```
tree_model = rpart(violent_crime_100k ~ ., data = train)
```

Let's take a look at the tree that was generated:

```
# plot(tree_model)
# text(tree_model, use.n = 0, cex = 0.8)
```

RMSE: 0.1703

```
tree_pred = predict(tree_model, test)
my_rmse(tree_pred, test_labels)
```

```
## [1] 0.1702926
```

Random Forest

```
random_forest_model = randomForest(violent_crime_100k ~ ., data = train)
random_forest_pred = predict(random_forest_model, test)
```

RMSE: 0.1444

```
my_rmse(random_forest_pred, test_labels)

## [1] 0.1446817
```

Support Vector Machine

```
svm_model = svm(violent_crime_100k ~ ., data = train)
svm_pred = predict(svm_model, newdata = test)
```

RMSE: 0.1447

```
my_rmse(svm_pred, test_labels)

## [1] 0.1446891
```

Neural Net

```
neural_vars = c('pct_black', 'pct_kids_to_unmarried',
'num_vacant_households',
                'pct_ppl_dense_housing', 'pct_with_pub_assist',
'num_homeless',
                'pct_less_9th_gr', 'population', 'pct_no_hs',
'violent_crime_100k')

neural_train = train[, neural_vars]
neural_test = test[, neural_vars]
```

Creating a neural network model using the variables that are most correlated to the class variable:

```
nn_model = neuralnet(violent_crime_100k ~ pct_black + pct_kids_to_unmarried +
num_vacant_households +
                                          pct_ppl_dense_housing +
pct_with_pub_assist +
                                          num_homeless + pct_less_9th_gr +
population +
                                          pct_no_hs, data = neural_train)
```

Cannot have class variable here:

```
nn_pred = compute(nn_model, neural_test[-10])
```

RMSE: 0.1521

```
my_rmse(nn_pred$net.result, neural_test$violent_crime_100k)

## [1] 0.1520672817
```

Overall, the linear regression model worked the best, with the lowest RMSE value of 0.1329