Statistical Analysis of the Communities and Crime Data Set

Angelina Kolomoytseva

9/22/2021

Introduction

The following report presents the exploration and modeling of the *Communities and Crime* data set, acquired from the UCI machine learning repository website. The data set involves 2215 observations and 147 variables. One important advantage of this data set is that the data source includes a detailed explanation of the variable.

Decription of Variables in the *Communities and Crime* data set

- General Information (non-predictive) Variables
  - Community name (string)
  - U.S. state (categorical)
  - County Code (numeric)
  - Community Code (numeric)
  - Fold (numeric)
- Demographic Variables (i.e, population, age, ethnicity)
  - As total number or average (2 variables, continuous)
  - As percentage of population (8 variables, continuous)
- Income Variables (i.e, median household income, per capita income, etc.)
  - As total number or median (2 variables, continuous)
  - As percentage of population (6 variables, continuous)
  - Per capita (7 variables, continuous)
- Community Variables (i.e, the total number or percent of the population considered urban, the total number percentage of people under the poverty level, number of people in homeless shelters, number of homeless people counted in the street, etc.)
  - As total number (4 variables, continuous)
  - As percentage of population (6 variables, continuous)
- Education Variables (i.e, percentage of people 25 and over with a bachelor's degree or higher education, etc.)
  - As percentage of population (3 variables, continuous)
- Employment Status Variables (i.e, percentage of people 16 and over who are employed, percentage of people 16 and over who are employed in management or professional occupations, etc.)
  - As percentage (6 variables, continuous)
- Marital Status Variables (i.e, percentage of males/females who are divorced or have never married, percentage of population who are divorced)
  - As percentage of population (4 variables, continuous)
- Household/Family Characteristics Variables (i.e, mean number of people per family, percentage of kids in family housing with two parents, percentage of kids born to never married, percent of family households that are large, mean persons per household, etc.)
  - As total number or average (3 variables, continuous)
  - As percentage (9 variables, continuous)

- Foreign born/Immigrated Variables (i.e, total number of people known to be foreign born, percent of population who have immigrated within the last 5 years, percent of people who do not speak English well, etc.)
  - As total number (1 variable, continuous)
  - As percentage of population (11 variables, continuous)
- Housing Variables (i.e, mean persons per owner occupied/rental household, median number of bedrooms, number of vacant households, percent of vacant housing that has been vacant more than 6 months, median year housing units built, owner occupied housing - median value, rental housing - median rent, median gross rent as a percentage of household income, etc.)
  - As total number, mean, median, or lower/upper quartile (17 variables, continuous)
  - As percentage (9 variables, continuous)
- Law Enforcement Variables (i.e, number of sworn full time police officers, percent of sworn full time police officers on patrol, total requests for police, police average overtime worked, police operating budget, etc.)
  - As total number or average (12 variables, continuous)
  - As percentage of population (7 variables, continuous)
  - Per 100K population (4 variables, continuous)
- Other law enforcement Variables (i.e, population density in persons per square mile, percent of people using public transit for commuting, land area in square miles)
  - As total number (2 variables, continuous)
  - As percentage of population (1 variable, continuous)
- Crime Variables (i.e, number of robberies in 1995, number of robberies per 100K population, number of auto thefts in 1995, number of auto thefts per 100K population, total number of violent crimes per 100K population, total number of non-violent crimes per 100K population, etc.)
  - As total number (8 variables, continuous)
  - Per 100K population (10 variables, continuous)

Data Exploration

R command `read_csv("crimedata.csv")` allows importing data without any modifications as long as the current `.Rmd` file is placed in the same folder with the `crimedata.csv` file.

The following R commands allow us to explore the dimension, variable names, and the number of the missing values in the *Communities and Crime* data set:

```
library(dplyr)
library(readxl)
library(readr)
crimedata <- read_csv("crimedata.csv", na= c("?", "NA"))

options(width = 80)
# structure
#str(crimedata)
dim(crimedata)
```

```
## [1] 2215  147
```

```
names(crimedata)
```

```
##   [1] "Communityname"         "state"                 "countyCode"
##   [4] "communityCode"         "fold"                  "population"
##   [7] "householdsize"         "racepctblack"          "racePctWhite"
##  [10] "racePctAsian"          "racePctHisp"           "agePct12t21"
##  [13] "agePct12t29"           "agePct16t24"           "agePct65up"
```

```
##  [16] "numbUrban"             "pctUrban"              "medIncome"
##  [19] "pctWWage"              "pctWFarmSelf"          "pctWInvInc"
##  [22] "pctWSocSec"            "pctWPubAsst"           "pctWRetire"
##  [25] "medFamInc"             "perCapInc"             "whitePerCap"
##  [28] "blackPerCap"           "indianPerCap"          "AsianPerCap"
##  [31] "OtherPerCap"           "HispPerCap"            "NumUnderPov"
##  [34] "PctPopUnderPov"        "PctLess9thGrade"       "PctNotHSGrad"
##  [37] "PctBSorMore"           "PctUnemployed"         "PctEmploy"
##  [40] "PctEmplManu"           "PctEmplProfServ"       "PctOccupManu"
##  [43] "PctOccupMgmtProf"      "MalePctDivorce"        "MalePctNevMarr"
##  [46] "FemalePctDiv"          "TotalPctDiv"           "PersPerFam"
##  [49] "PctFam2Par"            "PctKids2Par"           "PctYoungKids2Par"
##  [52] "PctTeen2Par"           "PctWorkMomYoungKids"   "PctWorkMom"
##  [55] "NumKidsBornNeverMar"   "PctKidsBornNeverMar"   "NumImmig"
##  [58] "PctImmigRecent"        "PctImmigRec5"          "PctImmigRec8"
##  [61] "PctImmigRec10"         "PctRecentImmig"        "PctRecImmig5"
##  [64] "PctRecImmig8"          "PctRecImmig10"         "PctSpeakEnglOnly"
##  [67] "PctNotSpeakEnglWell"   "PctLargHouseFam"       "PctLargHouseOccup"
##  [70] "PersPerOccupHous"      "PersPerOwnOccHous"     "PersPerRentOccHous"
##  [73] "PctPersOwnOccup"       "PctPersDenseHous"      "PctHousLess3BR"
##  [76] "MedNumBR"              "HousVacant"            "PctHousOccup"
##  [79] "PctHousOwnOcc"         "PctVacantBoarded"      "PctVacMore6Mos"
##  [82] "MedYrHousBuilt"        "PctHousNoPhone"        "PctWOFullPlumb"
##  [85] "OwnOccLowQuart"        "OwnOccMedVal"          "OwnOccHiQuart"
##  [88] "OwnOccQrange"          "RentLowQ"              "RentMedian"
##  [91] "RentHighQ"             "RentQrange"            "MedRent"
##  [94] "MedRentPctHousInc"     "MedOwnCostPctInc"      "MedOwnCostPctIncNoMtg"
##  [97] "NumInShelters"         "NumStreet"             "PctForeignBorn"
## [100] "PctBornSameState"      "PctSameHouse85"        "PctSameCity85"
## [103] "PctSameState85"        "LemasSwornFT"          "LemasSwFTPerPop"
## [106] "LemasSwFTFieldOps"     "LemasSwFTFieldPerPop"  "LemasTotalReq"
## [109] "LemasTotReqPerPop"     "PolicReqPerOffic"      "PolicPerPop"
## [112] "RacialMatchCommPol"    "PctPolicWhite"         "PctPolicBlack"
## [115] "PctPolicHisp"          "PctPolicAsian"         "PctPolicMinor"
## [118] "OfficAssgnDrugUnits"   "NumKindsDrugsSeiz"     "PolicAveOTWorked"
## [121] "LandArea"              "PopDens"               "PctUsePubTrans"
## [124] "PolicCars"             "PolicOperBudg"         "LemasPctPolicOnPatr"
## [127] "LemasGangUnitDeploy"   "LemasPctOfficDrugUn"   "PolicBudgPerPop"
## [130] "murders"               "murdPerPop"            "rapes"
## [133] "rapesPerPop"           "robberies"             "robbbPerPop"
## [136] "assaults"              "assaultPerPop"         "burglaries"
## [139] "burglPerPop"           "larcenies"             "larcPerPop"
## [142] "autoTheft"             "autoTheftPerPop"       "arsons"
## [145] "arsonsPerPop"          "ViolentCrimesPerPop"   "nonViolPerPop"
```

```
sum(is.na(crimedata))
```

```
## [1] 44592
```

```
colSums(is.na(crimedata))
```

```
##         Communityname                 state            countyCode
##                     0                     0                  1221
##         communityCode                  fold            population
##                  1224                     0                     0
##         householdsize          racepctblack          racePctWhite
```

```
##                   0                   0                   0
##         racePctAsian          racePctHisp          agePct12t21
##                   0                   0                   0
##          agePct12t29          agePct16t24          agePct65up
##                   0                   0                   0
##            numbUrban             pctUrban           medIncome
##                   0                   0                   0
##             pctWWage          pctWFarmSelf          pctWInvInc
##                   0                   0                   0
##           pctWSocSec          pctWPubAsst          pctWRetire
##                   0                   0                   0
##            medFamInc             perCapInc          whitePerCap
##                   0                   0                   0
##          blackPerCap          indianPerCap          AsianPerCap
##                   0                   0                   0
##          OtherPerCap            HispPerCap          NumUnderPov
##                   1                   0                   0
##        PctPopUnderPov       PctLess9thGrade         PctNotHSGrad
##                   0                   0                   0
##          PctBSorMore         PctUnemployed            PctEmploy
##                   0                   0                   0
##          PctEmplManu       PctEmplProfServ          PctOccupManu
##                   0                   0                   0
##     PctOccupMgmtProf        MalePctDivorce        MalePctNevMarr
##                   0                   0                   0
##          FemalePctDiv           TotalPctDiv           PersPerFam
##                   0                   0                   0
##            PctFam2Par           PctKids2Par     PctYoungKids2Par
##                   0                   0                   0
##          PctTeen2Par   PctWorkMomYoungKids           PctWorkMom
##                   0                   0                   0
##   NumKidsBornNeverMar   PctKidsBornNeverMar             NumImmig
##                   0                   0                   0
##        PctImmigRecent         PctImmigRec5         PctImmigRec8
##                   0                   0                   0
##         PctImmigRec10        PctRecentImmig         PctRecImmig5
##                   0                   0                   0
##         PctRecImmig8        PctRecImmig10     PctSpeakEnglOnly
##                   0                   0                   0
##   PctNotSpeakEnglWell      PctLargHouseFam     PctLargHouseOccup
##                   0                   0                   0
##      PersPerOccupHous     PersPerOwnOccHous    PersPerRentOccHous
##                   0                   0                   0
##       PctPersOwnOccup      PctPersDenseHous       PctHousLess3BR
##                   0                   0                   0
##             MedNumBR           HousVacant         PctHousOccup
##                   0                   0                   0
##         PctHousOwnOcc      PctVacantBoarded        PctVacMore6Mos
##                   0                   0                   0
##        MedYrHousBuilt        PctHousNoPhone        PctWOFullPlumb
##                   0                   0                   0
##        OwnOccLowQuart          OwnOccMedVal         OwnOccHiQuart
##                   0                   0                   0
##          OwnOccQrange             RentLowQ           RentMedian
```

```
##                   0                   0                   0
##           RentHighQ           RentQrange               MedRent
##                   0                   0                   0
##    MedRentPctHousInc     MedOwnCostPctInc MedOwnCostPctIncNoMtg
##                   0                   0                   0
##        NumInShelters           NumStreet       PctForeignBorn
##                   0                   0                   0
##     PctBornSameState       PctSameHouse85        PctSameCity85
##                   0                   0                   0
##       PctSameState85         LemasSwornFT      LemasSwFTPerPop
##                   0                1872                1872
##    LemasSwFTFieldOps  LemasSwFTFieldPerPop        LemasTotalReq
##                1872                1872                1872
##    LemasTotReqPerPop     PolicReqPerOffic          PolicPerPop
##                1872                1872                1872
##   RacialMatchCommPol        PctPolicWhite        PctPolicBlack
##                1872                1872                1872
##         PctPolicHisp        PctPolicAsian        PctPolicMinor
##                1872                1872                1872
##  OfficAssgnDrugUnits     NumKindsDrugsSeiz     PolicAveOTWorked
##                1872                1872                1872
##             LandArea             PopDens       PctUsePubTrans
##                   0                   0                   0
##            PolicCars        PolicOperBudg   LemasPctPolicOnPatr
##                1872                1872                1872
##  LemasGangUnitDeploy   LemasPctOfficDrugUn      PolicBudgPerPop
##                1872                   0                1872
##              murders           murdPerPop               rapes
##                   0                   0                 208
##          rapesPerPop            robberies          robbbPerPop
##                 208                   1                   1
##             assaults        assaultPerPop           burglaries
##                   13                  13                   3
##          burglPerPop            larcenies          larcPerPop
##                    3                   3                   3
##            autoTheft       autoTheftPerPop              arsons
##                    3                   3                  91
##         arsonsPerPop   ViolentCrimesPerPop        nonViolPerPop
##                   91                 221                  97
```

Unfortunately, many of the law enforcement variables starting from *LemasSwornFT* (i.e., number of sworn full time police officers) have a large number of missing values: 1872 out of 2215, i.e., almost 85% of the values are missing. The data source website includes the following explanation for this: "a limitation was that the LEMAS survey was of the police departments with at least 100 officers, plus a random sample of smaller departments. For our purposes, communities not found in both census and crime data sets were omitted. Many communities are missing LEMAS data". Several non-predictive variables, such as countryCode and communityCode have many missing values as well. However, those are not important for the analysis.

I exclude the variables with many missing values:

```
crimedata1 <- crimedata[ -c(1, 3:5, 104:120, 124:127, 129) ]
dim(crimedata1)
## [1] 2215  121
```

For the regression analysis, the *Violent Crime Rate* and *Non-violent Crime Rate* will be used as the response variables (separately). The data source provides two variables: *ViolentCrimesPerPop* and *nonViolPerPop* (i.e., total number of violent and non-violent crimes per 100K population, respectively). It also includes some detail on how those variables were generated. I have checked that the total number of different types of crimes divided by the total population for a community approximately matches the sum of these two variables divided by 100,000. Also, all of the percentage type variables in the data set are in the percentage form (not in decimal form). Thus, I define the *Viol.Rate* and *nonViol.Rate* as (*ViolentCrimesPerPop*/100,000 x 100) and (*nonViolPerPop*/100,000 x 100), respectively. The predictors will include all the other variables except for the individual types of different crimes in terms of the total number and the number that is given per 100K population. However, I will still use these individual types of crimes in the data exploration step.

```
Viol.Rate <- round(crimedata1$ViolentCrimesPerPop/1000, digits=5)
nonViol.Rate <- round(crimedata1$nonViolPerPop/1000, digits=5)
crimedata2 <- data.frame(crimedata1, Viol.Rate, nonViol.Rate)
head(crimedata2[,120:123])
##   ViolentCrimesPerPop nonViolPerPop Viol.Rate nonViol.Rate
## 1               41.02       1394.59   0.04102      1.39459
## 2              127.56       1955.95   0.12756      1.95595
## 3              218.59       6167.51   0.21859      6.16751
## 4              306.64            NA   0.30664           NA
## 5                  NA       9988.79        NA      9.98879
## 6              442.95       6867.42   0.44295      6.86742
crimedata2 <- crimedata2[ -c(120:121)]
dim(crimedata2)
## [1] 2215  121
```

Since the data set is quite large, examining pairwise linear associations between all variables is not very practical. Yet, investigating strong linear correlations that appear to be significant may be helpful. I use the following function to gain some insight:

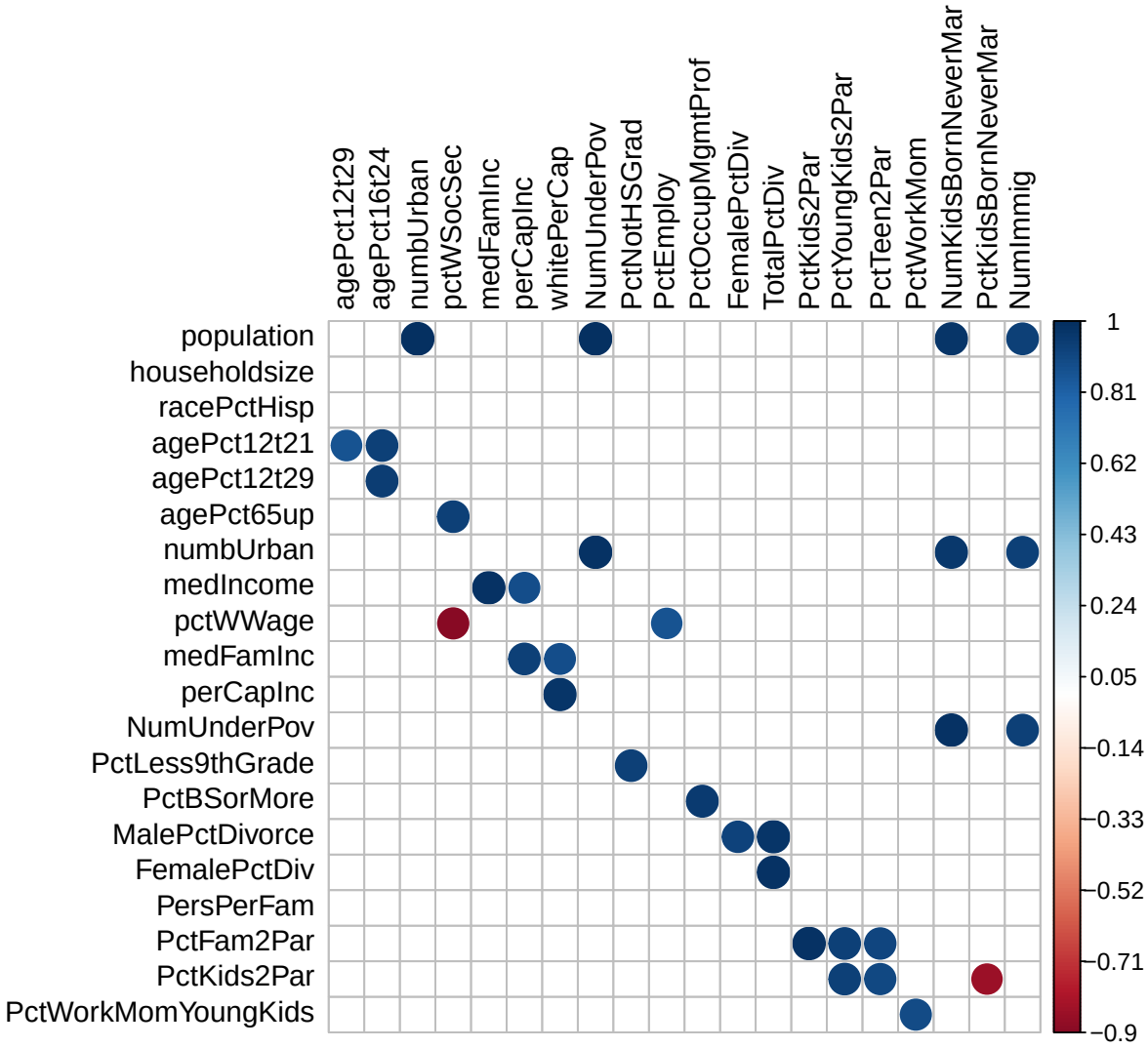
```
# A function to select significant correlations
#install.packages("corrplot")
library(corrplot)
## Warning: package 'corrplot' was built under R version 4.0.4

corr.fn <- function(data, sig){
  #convert data to numeric in order to run correlations
  #convert to factor first to keep the integrity of the data
  # each value will become a number rather than turn into NA
  df_cor <- data %>% mutate_if(is.character, as.factor)
  df_cor <- df_cor %>% mutate_if(is.factor, as.numeric)
  #run a correlation and drop the insignificant ones
  corr <- cor(df_cor, use="complete.obs")
  #prepare to drop duplicates and correlations of 1
  corr[lower.tri(corr, diag=TRUE)] <- NA
  #drop perfect correlations
  corr[corr == 1] <- NA
  #turn into a 3-column table
  corr <- as.data.frame(as.table(corr), row.names= NULL, optional = FALSE)
  #dim(corr)
  #remove the NA values from above
  corr <- na.omit(corr)
```

```
  #select significant values
  corr <- subset(corr, abs(Freq) > sig)
  #sort by highest correlation
  corr <- corr[order(-abs(corr$Freq)), ]
  #print table
  print(corr, row.names = FALSE)
  #turn corr back into matrix in order to plot with corrplot
  mtx_corr <- reshape2::acast(corr, Var1~Var2, value.var="Freq")

  mtx_corr1 <- mtx_corr[ 1:20, 1:20]
  #dim(mtx_corr1)
  #plot correlations visually
  corrplot(mtx_corr1, is.corr=FALSE, tl.col="black", na.label=" ")
  }

corr.fn(data=crimedata2, sig=0.85)
##                 Var1                Var2      Freq
##           population           numbUrban 0.9990379
##         PctRecImmig8       PctRecImmig10 0.9952219
##         PctRecImmig5        PctRecImmig8 0.9937611
##       OwnOccLowQuart        OwnOccMedVal 0.9914064
##       PctRecentImmig        PctRecImmig5 0.9898631
##           population         NumUnderPov 0.9891905
##           RentMedian             MedRent 0.9880038
##            numbUrban         NumUnderPov 0.9872327
##      PctLargHouseFam   PctLargHouseOccup 0.9864056
##           PctFam2Par         PctKids2Par 0.9858735
##         PctRecImmig5       PctRecImmig10 0.9851953
##         OwnOccMedVal       OwnOccHiQuart 0.9847504
##          NumUnderPov           autoTheft 0.9841697
##         FemalePctDiv         TotalPctDiv 0.9834567
##           burglaries           larcenies 0.9834540
##  NumKidsBornNeverMar           robberies 0.9829632
##           population           autoTheft 0.9829524
##          NumUnderPov NumKidsBornNeverMar 0.9827583
##      PctPersOwnOccup       PctHousOwnOcc 0.9822650
##            numbUrban           autoTheft 0.9817016
##  NumKidsBornNeverMar             murders 0.9799673
##           RentMedian           RentHighQ 0.9793943
##            medIncome           medFamInc 0.9793766
##       PctRecentImmig        PctRecImmig8 0.9786083
##             RentHighQ             MedRent 0.9768052
##            perCapInc         whitePerCap 0.9763308
##              murders           robberies 0.9761005
##       MalePctDivorce         TotalPctDiv 0.9755096
##          NumUnderPov             murders 0.9752823
##          NumUnderPov           robberies 0.9746141
##            robberies           autoTheft 0.9734297
##              murders           autoTheft 0.9727907
##           burglaries           autoTheft 0.9700459
##           population NumKidsBornNeverMar 0.9697116
##           population          burglaries 0.9693797
##             assaults           autoTheft 0.9688670
```

```
##            numbUrban          burglaries  0.9688543
##  NumKidsBornNeverMar           autoTheft  0.9687544
##              murders            assaults  0.9679052
##            numbUrban NumKidsBornNeverMar  0.9678098
##           population           robberies  0.9668093
##           population             murders  0.9664251
##          NumUnderPov          burglaries  0.9663998
##       PctRecentImmig       PctRecImmig10  0.9663462
##            numbUrban           robberies  0.9646403
##            numbUrban             murders  0.9645544
##       OwnOccLowQuart       OwnOccHiQuart  0.9632361
##           population           larcenies  0.9627810
##            numbUrban           larcenies  0.9624167
##             RentLowQ          RentMedian  0.9621057
##            robberies            assaults  0.9620588
##           population            assaults  0.9603662
##             NumImmig           robberies  0.9597605
##        PctRecImmig10      PctForeignBorn  0.9596369
##              assaults          burglaries  0.9594091
##            numbUrban            assaults  0.9588176
##          NumUnderPov            assaults  0.9580432
##                rapes          burglaries  0.9549604
##          NumUnderPov           larcenies  0.9549468
##            NumStreet           robberies  0.9548143
##              murders          burglaries  0.9546704
##        NumInShelters           NumStreet  0.9529759
##          PctBSorMore    PctOccupMgmtProf  0.9511871
##         PctRecImmig8      PctForeignBorn  0.9498705
##             larcenies           autoTheft  0.9496309
##        NumInShelters           robberies  0.9488286
##             RentLowQ             MedRent  0.9483720
##         PctImmigRec8       PctImmigRec10  0.9479333
##                rapes           larcenies  0.9468639
##        assaultPerPop           Viol.Rate  0.9453698
##          agePct12t29         agePct16t24  0.9453249
##  NumKidsBornNeverMar            assaults  0.9437438
##           HousVacant          burglaries  0.9435201
##           HousVacant           larcenies  0.9434554
##             NumImmig            assaults  0.9429655
##  NumKidsBornNeverMar       NumInShelters  0.9429131
##           larcPerPop        nonViol.Rate  0.9425977
##             NumImmig           autoTheft  0.9413200
##              assaults           larcenies  0.9405139
##  NumKidsBornNeverMar          burglaries  0.9396543
##           population             NumImmig  0.9386170
##           PersPerFam    PersPerOccupHous  0.9384103
##           PctFam2Par    PctYoungKids2Par  0.9369880
##             robberies          burglaries  0.9367634
##            numbUrban             NumImmig  0.9366282
##         PctImmigRec5        PctImmigRec8  0.9357105
##             NumImmig           NumStreet  0.9355435
##           HousVacant               rapes  0.9351301
##         PctRecImmig5      PctForeignBorn  0.9344108
```

```
##           agePct65up          pctWSocSec  0.9342738
##          NumUnderPov           NumImmig  0.9340218
##            medFamInc           perCapInc  0.9333352
##              murders           larcenies  0.9330481
##          PctKids2Par    PctYoungKids2Par  0.9319745
##      PctLess9thGrade        PctNotHSGrad  0.9315462
##          agePct12t21         agePct16t24  0.9315409
##           population       NumInShelters  0.9307511
##            numbUrban       NumInShelters  0.9287034
##          NumUnderPov       NumInShelters  0.9285607
##     PctSpeakEnglOnly PctNotSpeakEnglWell -0.9283435
##           population          HousVacant  0.9278608
##            numbUrban          HousVacant  0.9272262
##             RentLowQ           RentHighQ  0.9271934
##     PersPerOccupHous   PersPerOwnOccHous  0.9267253
##  NumKidsBornNeverMar           NumStreet  0.9254262
##             NumImmig             murders  0.9253019
##           population           NumStreet  0.9236478
##            NumStreet           autoTheft  0.9231994
##       MalePctDivorce        FemalePctDiv  0.9214840
##        NumInShelters           autoTheft  0.9211524
##            numbUrban           NumStreet  0.9210091
##           population               rapes  0.9208450
##            numbUrban               rapes  0.9205963
##          NumUnderPov           NumStreet  0.9205091
##        NumInShelters             murders  0.9189098
##          NumUnderPov          HousVacant  0.9188153
##  NumKidsBornNeverMar             NumImmig  0.9180700
##           PctFam2Par         PctTeen2Par  0.9177485
##          NumUnderPov               rapes  0.9172619
##       PctRecentImmig      PctForeignBorn  0.9151890
##                rapes           autoTheft  0.9151587
##          racePctHisp    PctSpeakEnglOnly -0.9150000
##  NumKidsBornNeverMar           larcenies  0.9139868
##           HousVacant           autoTheft  0.9131935
##            robberies           larcenies  0.9117803
##            autoTheft              arsons  0.9081622
##          PctKids2Par         PctTeen2Par  0.9073772
##       PctImmigRecent        PctImmigRec5  0.9069431
##                rapes            assaults  0.9049437
##              murders              arsons  0.9046879
##              murders               rapes  0.9036819
##           PersPerFam   PctLargHouseOccup  0.9031602
##          NumUnderPov              arsons  0.9027053
##             pctWWage          pctWSocSec -0.9020459
##              assaults              arsons  0.9016917
##           HousVacant             murders  0.9007747
##           PersPerFam   PersPerOwnOccHous  0.8997207
##            NumStreet             murders  0.8995213
##           population              arsons  0.8979823
##  PctWorkMomYoungKids          PctWorkMom  0.8977099
##                rapes              arsons  0.8969499
##            numbUrban              arsons  0.8968283
```

```
##  PctNotSpeakEnglWell    PctPersDenseHous  0.8954701
##        NumInShelters             assaults  0.8940684
##            NumStreet             assaults  0.8935399
##       OwnOccLowQuart          RentMedian  0.8931923
##          racePctHisp PctNotSpeakEnglWell  0.8930663
##        NumInShelters           burglaries  0.8906021
##             NumImmig       NumInShelters  0.8900060
##  NumKidsBornNeverMar           HousVacant  0.8899949
##       OwnOccLowQuart            RentHighQ  0.8886046
##           burglaries              arsons  0.8881046
##           HousVacant             assaults  0.8878922
##            medIncome           perCapInc  0.8877038
##            medFamInc          whitePerCap  0.8861821
##  NumKidsBornNeverMar                rapes  0.8854679
##         OwnOccMedVal            RentHighQ  0.8852853
##        OwnOccHiQuart         OwnOccQrange  0.8852040
##         OwnOccMedVal           RentMedian  0.8844948
##        OwnOccHiQuart            RentHighQ  0.8841514
##             NumImmig           burglaries  0.8820732
##           PersPerFam      PctLargHouseFam  0.8812537
##        NumInShelters             larcenies  0.8805579
##        householdsize     PersPerOccupHous  0.8789565
##            larcenies                arsons  0.8783992
##      PctLargHouseFam     PctPersDenseHous  0.8753536
##        OwnOccHiQuart           RentMedian  0.8721058
##  NumKidsBornNeverMar                arsons  0.8708835
##         PctImmigRec5       PctImmigRec10  0.8690227
##          agePct12t21         agePct12t29  0.8687202
##             NumImmig             larcenies  0.8672843
##          racePctHisp     PctPersDenseHous  0.8659072
##     PctSpeakEnglOnly       PctForeignBorn -0.8645246
##            robberies                arsons  0.8641708
##            NumStreet           burglaries  0.8641678
##       OwnOccLowQuart              MedRent  0.8640840
##             NumImmig                arsons  0.8625458
##                rapes             robberies  0.8614949
##           HousVacant                arsons  0.8612484
##           HousVacant             robberies  0.8604947
##             pctWWage             PctEmploy  0.8603104
##  PctNotSpeakEnglWell       PctForeignBorn  0.8596108
##          PctKids2Par PctKidsBornNeverMar -0.8590727
##            medIncome              MedRent  0.8587439
##        PctRecImmig10 PctNotSpeakEnglWell  0.8555506
##         OwnOccMedVal              MedRent  0.8554999
##            medIncome           RentMedian  0.8544983
```



Based on the fact that many variables are provided as a total number and also as a percentage of the population, per capita or per 100K population value, etc., one can expect some of the predictors to be highly correlated. As such, we notice, for example, that the percentage of immigrants who immigrated within the last 8 years has a strong linear positive correlation with the percentage of immigrants who immigrated within the last 10 years. The same is true for other immigration-related variables, housing and rent variables, etc. Some other interesting correlations include: *population* (population of a community) and *NumUnderPov* (the number of people under the poverty level) with the correlation of *0.9892*; *NumUnderPov* (the number of people under the poverty level) and *NumKidsBornNeverMar* (the number of kids born to never married) with the correlation of *0.9828*; *NumImmig* (the total number of people known to be foreign-born) and *NumStreet* (the number of homeless people counted in the street) with the correlation of *0.9355*; *racePctHisp* (the percentage of the population that is of Hispanic heritage) and *PctSpeakEnglOnly* (the percent of people who speak only English) with the negative correlation coefficient of *-0.915*; *pctWWage* (the percentage of households with wage or salary income) has a negative correlation of *-0.902* with *pctWSocSec* (the percentage of households with social security income).

Many of the crime variables have high significant positive correlations among themselves (i.e, *burglaries* and *larcenies*, *robberies* and *autoTheft*, etc.).

Several crime types, including *robberies*, *autoTheft*, *murders*, *burglaries*, *assaults*, *larcenies*, and *rapes* have a strong positive linear association with one or more of the following variables: *NumUnderPov* (the number

of people under the poverty level), *NumKidsBornNeverMar* (the number of kids born to never married), *population* (population of a community), *numbUrban* (number of people living in areas classified as urban), *NumImmig* (the total number of people known to be foreign-born), *NumStreet* (the number of homeless people counted in the street), *NumInShelters* (the number of people in homeless shelters), and *HousVacant* (the number of vacant households).

The dark blue circles on the plot indicate some of the strong positive linear associations while the dark red circles display some of the strong negative associations among the variables.

Summary Statistics

To explore the relationship between the predictors and the response variables, I first look at the descriptive statistics of the reduced data set, excluding the rows with the remaining missing values:

```
sum(is.na(crimedata2))
## [1] 963
crimedata3 <- na.omit(crimedata2)
dim(crimedata3)
## [1] 1901  121
```

The final data set for the analysis contains 1901 rows, which is 314 observation less than the original data set.

```
summary(crimedata3)
##     state              population      householdsize    racepctblack
##  Length:1901        Min.   :  10005   Min.   :1.600   Min.   : 0.000
##  Class :character   1st Qu.:  14312   1st Qu.:2.500   1st Qu.: 0.930
##  Mode  :character   Median :  22686   Median :2.660   Median : 3.040
##                     Mean   :  52500   Mean   :2.712   Mean   : 9.359
##                     3rd Qu.:  43264   3rd Qu.:2.860   3rd Qu.:11.430
##                     Max.   :7322564   Max.   :5.280   Max.   :96.670
##   racePctWhite    racePctAsian     racePctHisp      agePct12t21
##  Min.   : 2.68   Min.   : 0.060   Min.   : 0.120   Min.   : 4.58
##  1st Qu.:75.77   1st Qu.: 0.630   1st Qu.: 0.950   1st Qu.:12.21
##  Median :89.61   Median : 1.270   Median : 2.430   Median :13.62
##  Mean   :83.47   Mean   : 2.823   Mean   : 8.719   Mean   :14.41
##  3rd Qu.:95.96   3rd Qu.: 2.880   3rd Qu.: 8.920   3rd Qu.:15.41
##  Max.   :99.63   Max.   :57.460   Max.   :95.290   Max.   :54.40
##   agePct12t29     agePct16t24      agePct65up      numbUrban
##  Min.   : 9.38   Min.   : 4.64   Min.   : 1.66   Min.   :      0
##  1st Qu.:24.37   1st Qu.:11.31   1st Qu.: 8.84   1st Qu.:      0
##  Median :26.78   Median :12.52   Median :11.83   Median :  17336
##  Mean   :27.60   Mean   :13.97   Mean   :11.98   Mean   :  46972
##  3rd Qu.:29.20   3rd Qu.:14.37   3rd Qu.:14.51   3rd Qu.:  41958
##  Max.   :70.51   Max.   :63.62   Max.   :52.77   Max.   :7322564
##     pctUrban     medIncome         pctWWage      pctWFarmSelf     pctWInvInc
##  Min.   :  0   Min.   : 12908   Min.   :31.68   Min.   :0.000   Min.   : 9.02
##  1st Qu.:  0   1st Qu.: 23726   1st Qu.:73.45   1st Qu.:0.470   1st Qu.:34.21
##  Median :100   Median : 31270   Median :78.55   Median :0.700   Median :42.44
##  Mean   : 70   Mean   : 33956   Mean   :78.19   Mean   :0.888   Mean   :43.47
##  3rd Qu.:100   3rd Qu.: 41489   3rd Qu.:83.76   3rd Qu.:1.100   3rd Qu.:52.55
##  Max.   :100   Max.   :123625   Max.   :96.62   Max.   :6.530   Max.   :89.04
##    pctWSocSec     pctWPubAsst       pctWRetire      medFamInc
```

```
##  Min.   : 4.81   Min.   : 0.500   Min.   : 3.46   Min.   : 14257
##  1st Qu.:20.90   1st Qu.: 3.360   1st Qu.:13.00   1st Qu.: 29345
##  Median :26.66   Median : 5.620   Median :15.70   Median : 36533
##  Mean   :26.58   Mean   : 6.755   Mean   :16.09   Mean   : 39769
##  3rd Qu.:31.72   3rd Qu.: 9.090   3rd Qu.:18.79   3rd Qu.: 46847
##  Max.   :76.39   Max.   :26.920   Max.   :45.51   Max.   :131315
##    perCapInc      whitePerCap     blackPerCap      indianPerCap
##  Min.   : 5237   Min.   : 5472   Min.   :     0   Min.   :     0
##  1st Qu.:11563   1st Qu.:12643   1st Qu.:  6748   1st Qu.:  6405
##  Median :14087   Median :15087   Median :  9784   Median :  9943
##  Mean   :15604   Mean   :16616   Mean   : 11583   Mean   : 12329
##  3rd Qu.:17910   3rd Qu.:18710   3rd Qu.: 14549   3rd Qu.: 14807
##  Max.   :63302   Max.   :68850   Max.   :212120   Max.   :480000
##   AsianPerCap      OtherPerCap       HispPerCap     NumUnderPov
##  Min.   :     0   Min.   :     0   Min.   :    0   Min.   :    78
##  1st Qu.:  8542   1st Qu.:  5615   1st Qu.: 7288   1st Qu.:   905
##  Median : 12393   Median :  8205   Median : 9709   Median :  2197
##  Mean   : 14293   Mean   :  9480   Mean   :11037   Mean   :  7369
##  3rd Qu.: 17351   3rd Qu.: 11471   3rd Qu.:13431   3rd Qu.:  5054
##  Max.   :106165   Max.   :137000   Max.   :54648   Max.   :1384994
##  PctPopUnderPov  PctLess9thGrade   PctNotHSGrad    PctBSorMore
##  Min.   : 0.64   Min.   : 0.200   Min.   : 2.09   Min.   : 1.63
##  1st Qu.: 4.63   1st Qu.: 4.720   1st Qu.:14.16   1st Qu.:14.08
##  Median : 9.38   Median : 7.850   Median :21.54   Median :19.69
##  Mean   :11.66   Mean   : 9.429   Mean   :22.66   Mean   :23.03
##  3rd Qu.:17.04   3rd Qu.:12.140   3rd Qu.:29.59   3rd Qu.:29.00
##  Max.   :48.82   Max.   :49.890   Max.   :73.66   Max.   :73.63
##  PctUnemployed     PctEmploy      PctEmplManu    PctEmplProfServ
##  Min.   : 1.32   Min.   :24.82   Min.   : 2.05   Min.   : 8.69
##  1st Qu.: 4.09   1st Qu.:56.47   1st Qu.:11.98   1st Qu.:20.06
##  Median : 5.47   Median :62.47   Median :16.67   Median :23.29
##  Mean   : 6.01   Mean   :61.87   Mean   :17.80   Mean   :24.47
##  3rd Qu.: 7.41   3rd Qu.:67.55   3rd Qu.:22.72   3rd Qu.:27.59
##  Max.   :23.83   Max.   :84.67   Max.   :50.03   Max.   :62.67
##   PctOccupManu   PctOccupMgmtProf MalePctDivorce   MalePctNevMarr
##  Min.   : 1.37   Min.   : 6.48    Min.   : 2.130   Min.   :12.06
##  1st Qu.: 9.00   1st Qu.:21.85    1st Qu.: 7.110   1st Qu.:25.45
##  Median :13.00   Median :26.32    Median : 9.200   Median :29.02
##  Mean   :13.70   Mean   :28.28    Mean   : 9.172   Mean   :30.65
##  3rd Qu.:17.38   3rd Qu.:33.09    3rd Qu.:11.130   3rd Qu.:33.44
##  Max.   :44.27   Max.   :64.97    Max.   :19.090   Max.   :76.32
##   FemalePctDiv    TotalPctDiv      PersPerFam      PctFam2Par
##  Min.   : 3.35   Min.   : 2.83   Min.   :2.290   Min.   :32.24
##  1st Qu.: 9.87   1st Qu.: 8.59   1st Qu.:2.990   1st Qu.:67.82
##  Median :12.63   Median :11.03   Median :3.100   Median :74.91
##  Mean   :12.39   Mean   :10.87   Mean   :3.132   Mean   :74.04
##  3rd Qu.:14.87   3rd Qu.:13.08   3rd Qu.:3.220   3rd Qu.:81.78
##  Max.   :23.46   Max.   :19.11   Max.   :4.640   Max.   :93.60
##   PctKids2Par    PctYoungKids2Par  PctTeen2Par    PctWorkMomYoungKids
##  Min.   :26.11   Min.   : 27.43   Min.   :30.64   Min.   :24.42
##  1st Qu.:63.73   1st Qu.: 74.82   1st Qu.:69.94   1st Qu.:55.30
##  Median :72.23   Median : 83.92   Median :76.74   Median :60.62
##  Mean   :71.03   Mean   : 81.90   Mean   :75.44   Mean   :60.32
```

```
##  3rd Qu.:80.00   3rd Qu.: 91.54   3rd Qu.:82.69   3rd Qu.:65.65
##  Max.   :92.58   Max.   :100.00   Max.   :97.34   Max.   :87.97
##    PctWorkMom    NumKidsBornNeverMar PctKidsBornNeverMar    NumImmig
##  Min.   :41.95   Min.   :     0      Min.   : 0.000      Min.   :     20
##  1st Qu.:64.92   1st Qu.:   145      1st Qu.: 1.070      1st Qu.:    421
##  Median :69.23   Median :   357      Median : 2.060      Median :   1082
##  Mean   :68.74   Mean   :  2033      Mean   : 3.109      Mean   :   6466
##  3rd Qu.:73.20   3rd Qu.:  1061      3rd Qu.: 3.930      3rd Qu.:   3461
##  Max.   :89.37   Max.   :527557      Max.   :24.190      Max.   :2082931
##  PctImmigRecent   PctImmigRec5    PctImmigRec8   PctImmigRec10
##  Min.   : 0.00   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  1st Qu.: 6.97   1st Qu.:11.72   1st Qu.:18.04   1st Qu.:23.60
##  Median :12.48   Median :19.74   Median :27.60   Median :35.58
##  Mean   :13.69   Mean   :20.78   Mean   :28.16   Mean   :35.45
##  3rd Qu.:18.05   3rd Qu.:27.63   3rd Qu.:37.12   3rd Qu.:46.70
##  Max.   :64.29   Max.   :76.16   Max.   :80.81   Max.   :88.00
##  PctRecentImmig    PctRecImmig5     PctRecImmig8    PctRecImmig10
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 0.190   1st Qu.: 0.300   1st Qu.: 0.440   1st Qu.: 0.560
##  Median : 0.540   Median : 0.810   Median : 1.110   Median : 1.430
##  Mean   : 1.174   Mean   : 1.823   Mean   : 2.482   Mean   : 3.169
##  3rd Qu.: 1.420   3rd Qu.: 2.210   3rd Qu.: 3.020   3rd Qu.: 3.820
##  Max.   :13.710   Max.   :19.930   Max.   :25.340   Max.   :32.630
##  PctSpeakEnglOnly PctNotSpeakEnglWell PctLargHouseFam  PctLargHouseOccup
##  Min.   : 6.15    Min.   : 0.000      Min.   : 0.960   Min.   : 0.440
##  1st Qu.:83.48    1st Qu.: 0.520      1st Qu.: 3.420   1st Qu.: 2.370
##  Median :91.55    Median : 0.990      Median : 4.300   Median : 3.070
##  Mean   :86.29    Mean   : 2.602      Mean   : 5.515   Mean   : 4.021
##  3rd Qu.:95.26    3rd Qu.: 2.550      3rd Qu.: 6.030   3rd Qu.: 4.300
##  Max.   :98.98    Max.   :38.330      Max.   :34.870   Max.   :30.870
##  PersPerOccupHous PersPerOwnOccHous PersPerRentOccHous PctPersOwnOccup
##  Min.   :1.580    Min.   :1.610     Min.   :1.580      Min.   :13.93
##  1st Qu.:2.400    1st Qu.:2.540     1st Qu.:2.130      1st Qu.:56.56
##  Median :2.560    Median :2.710     Median :2.300      Median :65.04
##  Mean   :2.619    Mean   :2.738     Mean   :2.389      Mean   :65.56
##  3rd Qu.:2.770    3rd Qu.:2.900     3rd Qu.:2.550      3rd Qu.:75.55
##  Max.   :4.520    Max.   :4.480     Max.   :4.730      Max.   :96.59
##  PctPersDenseHous PctHousLess3BR     MedNumBR       HousVacant
##  Min.   : 0.15    Min.   : 3.06   Min.   :1.000   Min.   :    36
##  1st Qu.: 1.31    1st Qu.:37.67   1st Qu.:2.000   1st Qu.:   309
##  Median : 2.49    Median :46.83   Median :3.000   Median :   584
##  Mean   : 4.40    Mean   :45.85   Mean   :2.621   Mean   :  1734
##  3rd Qu.: 4.96    3rd Qu.:54.22   3rd Qu.:3.000   3rd Qu.:  1281
##  Max.   :59.49    Max.   :95.34   Max.   :4.000   Max.   :172768
##   PctHousOccup   PctHousOwnOcc   PctVacantBoarded PctVacMore6Mos
##  Min.   :37.47   Min.   :16.86   Min.   : 0.00    Min.   : 3.12
##  1st Qu.:90.95   1st Qu.:54.22   1st Qu.: 0.76    1st Qu.:24.55
##  Median :94.01   Median :62.11   Median : 1.72    Median :34.31
##  Mean   :92.69   Mean   :62.74   Mean   : 2.77    Mean   :35.01
##  3rd Qu.:95.94   3rd Qu.:71.83   3rd Qu.: 3.48    3rd Qu.:44.13
##  Max.   :99.00   Max.   :96.36   Max.   :39.89    Max.   :82.13
##  MedYrHousBuilt PctHousNoPhone   PctWOFullPlumb   OwnOccLowQuart
##  Min.   :1939   Min.   : 0.000   Min.   :0.0000   Min.   : 15700
```

```
##  1st Qu.:1956   1st Qu.: 0.940   1st Qu.:0.1700   1st Qu.: 42200
##  Median :1964   Median : 2.970   Median :0.3300   Median : 66900
##  Mean   :1963   Mean   : 4.411   Mean   :0.4328   Mean   : 92683
##  3rd Qu.:1971   3rd Qu.: 7.080   3rd Qu.:0.5600   3rd Qu.:128100
##  Max.   :1987   Max.   :23.630   Max.   :5.3300   Max.   :500001
##   OwnOccMedVal    OwnOccHiQuart     OwnOccQrange       RentLowQ
##  Min.   : 26600   Min.   : 36700   Min.   :     0   Min.   :  99.0
##  1st Qu.: 57100   1st Qu.: 75500   1st Qu.: 33100   1st Qu.: 215.0
##  Median : 86000   Median :111800   Median : 44700   Median : 308.0
##  Mean   :118021   Mean   :151346   Mean   : 58663   Mean   : 332.2
##  3rd Qu.:159800   3rd Qu.:196400   3rd Qu.: 69200   3rd Qu.: 426.0
##  Max.   :500001   Max.   :500001   Max.   :331000   Max.   :1001.0
##    RentMedian       RentHighQ        RentQrange     MedRent
##  Min.   : 139.0   Min.   : 203.0   Min.   :  0   Min.   : 192.0
##  1st Qu.: 290.0   1st Qu.: 367.0   1st Qu.:140   1st Qu.: 366.0
##  Median : 400.0   Median : 494.0   Median :175   Median : 473.0
##  Mean   : 433.4   Mean   : 534.2   Mean   :202   Mean   : 507.6
##  3rd Qu.: 552.0   3rd Qu.: 675.0   3rd Qu.:243   3rd Qu.: 627.0
##  Max.   :1001.0   Max.   :1001.0   Max.   :803   Max.   :1001.0
##  MedRentPctHousInc MedOwnCostPctInc MedOwnCostPctIncNoMtg NumInShelters
##  Min.   :14.90     Min.   :14.10    Min.   :10.10         Min.   :    0.00
##  1st Qu.:24.40     1st Qu.:19.20    1st Qu.:11.90         1st Qu.:    0.00
##  Median :26.20     Median :21.40    Median :12.80         Median :    0.00
##  Mean   :26.35     Mean   :21.31    Mean   :13.02         Mean   :   67.61
##  3rd Qu.:28.10     3rd Qu.:23.30    3rd Qu.:13.80         3rd Qu.:   24.00
##  Max.   :35.10     Max.   :32.70    Max.   :23.40         Max.   :23383.00
##    NumStreet        PctForeignBorn   PctBornSameState PctSameHouse85
##  Min.   :    0.00   Min.   : 0.190   Min.   : 6.75    Min.   :11.83
##  1st Qu.:    0.00   1st Qu.: 2.180   1st Qu.:48.18    1st Qu.:44.56
##  Median :    0.00   Median : 4.580   Median :61.84    Median :51.82
##  Mean   :   19.38   Mean   : 7.789   Mean   :60.00    Mean   :51.28
##  3rd Qu.:    1.00   3rd Qu.: 9.950   3rd Qu.:73.90    3rd Qu.:58.62
##  Max.   :10447.00   Max.   :60.400   Max.   :93.14    Max.   :78.56
##  PctSameCity85   PctSameState85     LandArea          PopDens
##  Min.   :27.95   Min.   :32.83   Min.   :   0.90   Min.   :   10
##  1st Qu.:71.74   1st Qu.:84.68   1st Qu.:   7.30   1st Qu.: 1176
##  Median :79.13   Median :89.53   Median :  13.70   Median : 2004
##  Mean   :76.99   Mean   :87.65   Mean   :  28.21   Mean   : 2804
##  3rd Qu.:84.67   3rd Qu.:92.70   3rd Qu.:  25.60   3rd Qu.: 3278
##  Max.   :96.59   Max.   :99.90   Max.   :3569.80   Max.   :44230
##  PctUsePubTrans   LemasPctOfficDrugUn    murders           murdPerPop
##  Min.   : 0.000   Min.   : 0.000      Min.   :   0.000   Min.   : 0.000
##  1st Qu.: 0.360   1st Qu.: 0.000      1st Qu.:   0.000   1st Qu.: 0.000
##  Median : 1.240   Median : 0.000      Median :   1.000   Median : 2.460
##  Mean   : 3.075   Mean   : 1.011      Mean   :   7.392   Mean   : 5.966
##  3rd Qu.: 3.440   3rd Qu.: 0.000      3rd Qu.:   3.000   3rd Qu.: 8.640
##  Max.   :54.330   Max.   :48.440      Max.   :1946.000   Max.   :91.090
##      rapes          rapesPerPop       robberies        robbbPerPop
##  Min.   :   0.00   Min.   :  0.00   Min.   :    0.0   Min.   :   0.00
##  1st Qu.:   2.00   1st Qu.: 11.57   1st Qu.:    6.0   1st Qu.:  29.00
##  Median :   7.00   Median : 27.05   Median :   19.0   Median :  78.98
##  Mean   :  27.28   Mean   : 36.23   Mean   :  234.4   Mean   : 166.76
##  3rd Qu.:  19.00   3rd Qu.: 51.17   3rd Qu.:   73.0   3rd Qu.: 194.30
```

```
##  Max.   :2818.00   Max.   :401.35   Max.   :86001.0   Max.   :2264.13
##     assaults       assaultPerPop       burglaries     burglPerPop
##  Min.   :    0.0   Min.   :   0.00   Min.   :    2   Min.   :   16.92
##  1st Qu.:   19.0   1st Qu.:  95.46   1st Qu.:   95   1st Qu.:  520.42
##  Median :   58.0   Median : 234.36   Median :  213   Median :  858.80
##  Mean   :  318.9   Mean   : 374.76   Mean   :  758   Mean   : 1056.58
##  3rd Qu.:  183.0   3rd Qu.: 511.45   3rd Qu.:  537   3rd Qu.: 1373.68
##  Max.   :62778.0   Max.   :3486.14   Max.   :99207   Max.   :11881.02
##    larcenies        larcPerPop         autoTheft        autoTheftPerPop
##  Min.   :    10   Min.   :   77.86   Min.   :     1.0   Min.   :   6.55
##  1st Qu.:   382   1st Qu.: 1996.46   1st Qu.:    31.0   1st Qu.: 162.59
##  Median :   745   Median : 3057.33   Median :    77.0   Median : 311.04
##  Mean   :  2121   Mean   : 3368.95   Mean   :   506.9   Mean   : 483.48
##  3rd Qu.:  1689   3rd Qu.: 4349.42   3rd Qu.:   243.0   3rd Qu.: 605.82
##  Max.   :235132   Max.   :25910.55   Max.   :112464.0   Max.   :4968.59
##      arsons          arsonsPerPop      Viol.Rate        nonViol.Rate
##  Min.   :   0.00   Min.   :  0.00   Min.   :0.00664   Min.   : 0.1168
##  1st Qu.:   1.00   1st Qu.:  7.25   1st Qu.:0.16375   1st Qu.: 2.9132
##  Median :   5.00   Median : 21.02   Median :0.36930   Median : 4.4791
##  Mean   :  30.48   Mean   : 32.04   Mean   :0.58371   Mean   : 4.9410
##  3rd Qu.:  16.00   3rd Qu.: 43.19   3rd Qu.:0.79293   3rd Qu.: 6.2655
##  Max.   :5119.00   Max.   :436.37   Max.   :4.87706   Max.   :27.1198
```

A **five-number summary** of the data consists of the following five sample quantiles: the *minimum*, the *first quartile*, the *median*, the *third quartile*, and the **maximum**. The *mean* is also included in the summary.

Several variables, such as *pctUrban* and *LemasPctOfficDrugUn* seem to be not very informative. There are some potential outliers on the high side of the data.

Association

It is also important to check the association between the two response variables and the predictors. Considering the classification part of the analysis, I also check the correlation between *PctBSorMore* (the percentage of people 25 and over with a bachelors degree or higher education) and other variables and between *medIncome* (median household income) and other variables.

```
cor(crimedata3$Viol.Rate, crimedata3[ -c(1) ])
##      population householdsize racepctblack racePctWhite racePctAsian
## [1,]  0.2120726   -0.01959399     0.625339   -0.6771956   0.03603103
##      racePctHisp agePct12t21 agePct12t29 agePct16t24 agePct65up numbUrban
## [1,]   0.2645212  0.02202454   0.1099322  0.04840845 0.05399668 0.2131591
##        pctUrban  medIncome  pctWWage pctWFarmSelf pctWInvInc pctWSocSec
## [1,] 0.07395461 -0.3963943 -0.282896   -0.1398243 -0.5567918   0.100952
##      pctWPubAsst pctWRetire medFamInc  perCapInc whitePerCap blackPerCap
## [1,]   0.5584722 -0.1071389 -0.410052 -0.3132774  -0.1859766  -0.2087871
##      indianPerCap AsianPerCap OtherPerCap HispPerCap NumUnderPov PctPopUnderPov
## [1,]  -0.05161095  -0.1331356  -0.1004149 -0.2308198   0.2377821      0.5000251
##      PctLess9thGrade PctNotHSGrad PctBSorMore PctUnemployed  PctEmploy
## [1,]       0.3709112    0.4666094  -0.2993101     0.4757308 -0.3125628
##      PctEmplManu PctEmplProfServ PctOccupManu PctOccupMgmtProf MalePctDivorce
## [1,] -0.05769492     -0.07139422    0.2821378       -0.3248576      0.5166304
##      MalePctNevMarr FemalePctDiv TotalPctDiv PersPerFam PctFam2Par PctKids2Par
## [1,]      0.2742171     0.541555   0.5412168  0.1455434 -0.6985964  -0.7281591
```

```
##      PctYoungKids2Par PctTeen2Par PctWorkMomYoungKids PctWorkMom
## [1,]       -0.6567168  -0.6541668         -0.01764763 -0.1414787
##      NumKidsBornNeverMar PctKidsBornNeverMar  NumImmig PctImmigRecent
## [1,]           0.2383514           0.7396185 0.1468389      0.1448305
##      PctImmigRec5 PctImmigRec8 PctImmigRec10 PctRecentImmig PctRecImmig5
## [1,]    0.1922394    0.2329452     0.2782078      0.2232752    0.2408713
##      PctRecImmig8 PctRecImmig10 PctSpeakEnglOnly PctNotSpeakEnglWell
## [1,]    0.2472721     0.2598336       -0.2269061           0.2797524
##      PctLargHouseFam PctLargHouseOccup PersPerOccupHous PersPerOwnOccHous
## [1,]       0.3411392         0.2567727      -0.01983215        -0.1007713
##      PersPerRentOccHous PctPersOwnOccup PctPersDenseHous PctHousLess3BR
## [1,]          0.2395771       -0.512504        0.4029456      0.4661065
##        MedNumBR HousVacant PctHousOccup PctHousOwnOcc PctVacantBoarded
## [1,] -0.3646806  0.2883536   -0.2556133    -0.4606949         0.475131
##      PctVacMore6Mos MedYrHousBuilt PctHousNoPhone PctWOFullPlumb OwnOccLowQuart
## [1,]     0.01757324     -0.1053673       0.476683       0.307364     -0.1950122
##      OwnOccMedVal OwnOccHiQuart OwnOccQrange   RentLowQ RentMedian  RentHighQ
## [1,]   -0.1777798    -0.1646304  -0.08264225 -0.2406297 -0.2299616 -0.2224984
##      RentQrange    MedRent MedRentPctHousInc MedOwnCostPctInc
## [1,] -0.1157187 -0.2308062         0.3113505       0.06433183
##      MedOwnCostPctIncNoMtg NumInShelters NumStreet PctForeignBorn
## [1,]            0.05246134     0.1944513 0.1400381      0.2016467
##      PctBornSameState PctSameHouse85 PctSameCity85 PctSameState85   LandArea
## [1,]      -0.08666947     -0.1560347    0.07191541    -0.01360647 0.07531676
##       PopDens PctUsePubTrans LemasPctOfficDrugUn   murders murdPerPop     rapes
## [1,] 0.262077      0.1902066           0.3191213 0.2476402  0.6766785 0.3353962
##      rapesPerPop robberies robbbPerPop  assaults assaultPerPop burglaries
## [1,]   0.5778006 0.2093678   0.8363753 0.3000157     0.9453698  0.3163043
##      burglPerPop larcenies larcPerPop autoTheft autoTheftPerPop    arsons
## [1,]   0.6977357 0.2946449  0.5100195 0.2445792       0.6443681 0.2328586
##      arsonsPerPop Viol.Rate nonViol.Rate
## [1,]    0.4149693         1    0.6755352
cor(crimedata3$nonViol.Rate, crimedata3[ -c(1) ])
##      population householdsize racepctblack racePctWhite racePctAsian
## [1,]  0.1191185    -0.1929894     0.474151   -0.4762647  -0.03450099
##      racePctHisp agePct12t21 agePct12t29 agePct16t24 agePct65up numbUrban
## [1,]   0.1749136  0.02377601   0.1115818  0.06666053  0.1262788  0.117136
##          pctUrban  medIncome   pctWWage pctWFarmSelf pctWInvInc pctWSocSec
## [1,] -0.000710343 -0.4652496 -0.3197183  -0.08125384 -0.4856934  0.1528602
##      pctWPubAsst  pctWRetire medFamInc  perCapInc whitePerCap blackPerCap
## [1,]   0.4606217 -0.08520737 -0.455433 -0.3167774  -0.2110615  -0.2473046
##      indianPerCap AsianPerCap OtherPerCap HispPerCap NumUnderPov PctPopUnderPov
## [1,]  -0.07624204  -0.1861745  -0.1202579 -0.2580052   0.1430689      0.5100829
##      PctLess9thGrade PctNotHSGrad PctBSorMore PctUnemployed  PctEmploy
## [1,]       0.2875593    0.3662798  -0.2709209     0.3916568 -0.3042169
##      PctEmplManu PctEmplProfServ PctOccupManu PctOccupMgmtProf MalePctDivorce
## [1,]  -0.1082331     -0.04321736    0.2204097        -0.290396      0.5853514
##      MalePctNevMarr FemalePctDiv TotalPctDiv  PersPerFam PctFam2Par PctKids2Par
## [1,]      0.2021434    0.5992672   0.6063286 -0.05189131 -0.6599978  -0.6668012
##      PctYoungKids2Par PctTeen2Par PctWorkMomYoungKids  PctWorkMom
## [1,]       -0.6138067  -0.6168553          0.06369609 -0.03115168
##      NumKidsBornNeverMar PctKidsBornNeverMar   NumImmig PctImmigRecent
## [1,]           0.1215166           0.5505852 0.05511244      0.1748564
```

```
##      PctImmigRec5 PctImmigRec8 PctImmigRec10 PctRecentImmig PctRecImmig5
## [1,]    0.2170453    0.2603904     0.2976672     0.09369742    0.1055244
##      PctRecImmig8 PctRecImmig10 PctSpeakEnglOnly PctNotSpeakEnglWell
## [1,]    0.1070139     0.1154898       -0.1128615           0.1484235
##      PctLargHouseFam PctLargHouseOccup PersPerOccupHous PersPerOwnOccHous
## [1,]        0.163147        0.08729743       -0.2019042        -0.2752962
##      PersPerRentOccHous PctPersOwnOccup PctPersDenseHous PctHousLess3BR
## [1,]         0.08731438      -0.5035655        0.2435143      0.4737501
##       MedNumBR HousVacant PctHousOccup PctHousOwnOcc PctVacantBoarded
## [1,] -0.398836  0.2067246   -0.3036481    -0.4623156        0.3235909
##      PctVacMore6Mos MedYrHousBuilt PctHousNoPhone PctWOFullPlumb OwnOccLowQuart
## [1,]    -0.04364755    -0.02514981      0.4922451      0.2424365     -0.2958043
##      OwnOccMedVal OwnOccHiQuart OwnOccQrange   RentLowQ RentMedian  RentHighQ
## [1,]   -0.2785992    -0.2633786   -0.1602291 -0.3253267  -0.340185 -0.3412779
##      RentQrange    MedRent MedRentPctHousInc MedOwnCostPctInc
## [1,] -0.2508091 -0.3490262         0.2263756      -0.08551563
##      MedOwnCostPctIncNoMtg NumInShelters  NumStreet PctForeignBorn
## [1,]          -0.006156711     0.1026679 0.05914292     0.05891884
##      PctBornSameState PctSameHouse85 PctSameCity85 PctSameState85   LandArea
## [1,]       -0.1120031     -0.2468237   -0.01839414    -0.08347561 0.05450896
##         PopDens PctUsePubTrans LemasPctOfficDrugUn   murders murdPerPop
## [1,] 0.08812175      0.0321276           0.2836925 0.1260416  0.4936093
##          rapes rapesPerPop  robberies robbbPerPop  assaults assaultPerPop
## [1,] 0.2234177   0.5689651 0.09902956   0.6214643 0.1537347     0.5899493
##      burglaries burglPerPop larcenies larcPerPop autoTheft autoTheftPerPop
## [1,]  0.2219688   0.8106424 0.2371676  0.9425977 0.1376853       0.5861271
##         arsons arsonsPerPop Viol.Rate nonViol.Rate
## [1,] 0.1351609    0.4032325 0.6755352            1
cor(crimedata3$PctBSorMore, crimedata3[ -c(1) ])
##        population householdsize racepctblack racePctWhite racePctAsian
## [1,] -0.004769162   -0.04067869   -0.1764581    0.2170757    0.2560691
##      racePctHisp agePct12t21 agePct12t29 agePct16t24 agePct65up   numbUrban
## [1,]  -0.2638345  0.08669034   0.0470167   0.1435027 -0.1974567 0.002724188
##       pctUrban medIncome  pctWWage pctWFarmSelf pctWInvInc pctWSocSec
## [1,] 0.2095083 0.6849089 0.4124488   0.07953508  0.7350539 -0.3696331
##      pctWPubAsst pctWRetire medFamInc perCapInc whitePerCap blackPerCap
## [1,]  -0.5592415 -0.1769298 0.7684257 0.7733709   0.7661573   0.3550773
##      indianPerCap AsianPerCap OtherPerCap HispPerCap NumUnderPov PctPopUnderPov
## [1,]    0.1547228   0.3167577   0.2496618   0.491582   -0.034101     -0.3865512
##      PctLess9thGrade PctNotHSGrad PctBSorMore PctUnemployed PctEmploy
## [1,]      -0.5807326   -0.7526821           1    -0.5493312 0.3906333
##      PctEmplManu PctEmplProfServ PctOccupManu PctOccupMgmtProf MalePctDivorce
## [1,]  -0.3166671       0.5860674   -0.7705047        0.9511871     -0.4757834
##      MalePctNevMarr FemalePctDiv TotalPctDiv PersPerFam PctFam2Par PctKids2Par
## [1,]      0.1893955   -0.4175368  -0.4518269 -0.2107613  0.4518379   0.4859004
##      PctYoungKids2Par PctTeen2Par PctWorkMomYoungKids  PctWorkMom
## [1,]        0.4927704   0.3350156         -0.07346351 0.002923306
##      NumKidsBornNeverMar PctKidsBornNeverMar      NumImmig PctImmigRecent
## [1,]         -0.03229286          -0.3127148 -0.0006708059       0.182345
##      PctImmigRec5 PctImmigRec8 PctImmigRec10 PctRecentImmig PctRecImmig5
## [1,]    0.1486167    0.1457454    0.09808571      0.1025752   0.06605085
##      PctRecImmig8 PctRecImmig10 PctSpeakEnglOnly PctNotSpeakEnglWell
## [1,]   0.06178017    0.03273316        0.1339388          -0.2127483
```

```
##      PctLargHouseFam PctLargHouseOccup PersPerOccupHous PersPerOwnOccHous
## [1,]      -0.3030243        -0.2777878       -0.1197188      -0.003487859
##      PersPerRentOccHous PctPersOwnOccup PctPersDenseHous PctHousLess3BR
## [1,]         -0.3532359       0.2669648       -0.3164325      -0.335845
##       MedNumBR  HousVacant PctHousOccup PctHousOwnOcc PctVacantBoarded
## [1,] 0.2076201 -0.02475905    0.1858418     0.2042035       -0.2808446
##      PctVacMore6Mos MedYrHousBuilt PctHousNoPhone PctWOFullPlumb OwnOccLowQuart
## [1,]     -0.1941964     0.08890447     -0.5194673     -0.3209379      0.6034699
##      OwnOccMedVal OwnOccHiQuart OwnOccQrange  RentLowQ RentMedian RentHighQ
## [1,]    0.6259818     0.6539191    0.6245808 0.5377265  0.5834731 0.5991477
##      RentQrange   MedRent MedRentPctHousInc MedOwnCostPctInc
## [1,]  0.4963048 0.5689907       -0.05246876        0.1668201
##      MedOwnCostPctIncNoMtg NumInShelters    NumStreet PctForeignBorn
## [1,]             -0.081579  0.0008136829 0.0003770247     0.05729905
##      PctBornSameState PctSameHouse85 PctSameCity85 PctSameState85     LandArea
## [1,]       -0.2892878    -0.06802342    -0.3249831       -0.29258 -0.005117687
##           PopDens PctUsePubTrans LemasPctOfficDrugUn     murders murdPerPop
## [1,] -0.03897818      0.2407879         -0.01626548 -0.02325684 -0.2478463
##            rapes rapesPerPop   robberies robbbPerPop    assaults assaultPerPop
## [1,] -0.04196281  -0.3055538 -0.01453556  -0.1876374 -0.02425177    -0.3071766
##       burglaries burglPerPop   larcenies larcPerPop   autoTheft autoTheftPerPop
## [1,] -0.02834001   -0.263312 -0.01577231 -0.2293883 -0.02136853      -0.1852676
##           arsons arsonsPerPop  Viol.Rate nonViol.Rate
## [1,] -0.02862846   -0.1976603 -0.2993101   -0.2709209
cor(crimedata3$medIncome, crimedata3[ -c(1) ])
##       population householdsize racepctblack racePctWhite racePctAsian
## [1,] -0.04921101       0.17707   -0.3368233    0.2913543    0.3139762
##      racePctHisp agePct12t21 agePct12t29 agePct16t24 agePct65up   numbUrban
## [1,]  -0.1591583  -0.2608544  -0.3293798  -0.2894104 -0.2543035 -0.03597296
##       pctUrban medIncome  pctWWage pctWFarmSelf pctWInvInc pctWSocSec
## [1,] 0.3338739         1 0.5754779  -0.04683274  0.7527483 -0.3935861
##      pctWPubAsst  pctWRetire medFamInc perCapInc whitePerCap blackPerCap
## [1,]  -0.6288321 -0.07265149 0.9793766 0.8877038   0.8401277   0.5231437
##      indianPerCap AsianPerCap OtherPerCap HispPerCap NumUnderPov PctPopUnderPov
## [1,]      0.21716   0.4268124   0.3623535  0.6315905  -0.0950269     -0.7591456
##      PctLess9thGrade PctNotHSGrad PctBSorMore PctUnemployed PctEmploy
## [1,]      -0.5461923   -0.6629189   0.6849089    -0.6204121 0.5976028
##      PctEmplManu PctEmplProfServ PctOccupManu PctOccupMgmtProf MalePctDivorce
## [1,] -0.04715581      0.03951178   -0.6029242        0.7316194     -0.5556692
##      MalePctNevMarr FemalePctDiv TotalPctDiv PersPerFam PctFam2Par PctKids2Par
## [1,]     -0.2033088   -0.5369478  -0.5601784 0.08021892  0.7155003   0.7021297
##      PctYoungKids2Par PctTeen2Par PctWorkMomYoungKids  PctWorkMom
## [1,]        0.6985889   0.6107083          -0.1703451 -0.07542862
##      NumKidsBornNeverMar PctKidsBornNeverMar    NumImmig PctImmigRecent
## [1,]         -0.06918699          -0.4537627 -0.01045389     -0.1773342
##      PctImmigRec5 PctImmigRec8 PctImmigRec10 PctRecentImmig PctRecImmig5
## [1,]   -0.1933557   -0.1796706    -0.1913264      0.0814882    0.0770481
##      PctRecImmig8 PctRecImmig10 PctSpeakEnglOnly PctNotSpeakEnglWell
## [1,]   0.09384748    0.08397189       0.01154302          -0.1001547
##      PctLargHouseFam PctLargHouseOccup PersPerOccupHous PersPerOwnOccHous
## [1,]      -0.1478373         -0.084826        0.2746462         0.3406229
##      PersPerRentOccHous PctPersOwnOccup PctPersDenseHous PctHousLess3BR
## [1,]        -0.08847483       0.6189057       -0.2321801     -0.6176862
```

```
##       MedNumBR HousVacant PctHousOccup PctHousOwnOcc PctVacantBoarded
## [1,] 0.4725104 -0.1069863    0.3151322     0.5850276       -0.3133261
##      PctVacMore6Mos MedYrHousBuilt PctHousNoPhone PctWOFullPlumb OwnOccLowQuart
## [1,]     -0.1890688      0.1395698     -0.6998258     -0.3573589       0.797481
##      OwnOccMedVal OwnOccHiQuart OwnOccQrange  RentLowQ RentMedian RentHighQ
## [1,]    0.7882011     0.7780772    0.6056251 0.8052249  0.8544983 0.8450975
##      RentQrange   MedRent MedRentPctHousInc MedOwnCostPctInc
## [1,]   0.621697 0.8587439        -0.2336604        0.3679737
##      MedOwnCostPctIncNoMtg NumInShelters   NumStreet PctForeignBorn
## [1,]           -0.02220444   -0.04483602 -0.02284211       0.182766
##      PctBornSameState PctSameHouse85 PctSameCity85 PctSameState85    LandArea
## [1,]       -0.2377015      0.2627743    0.01417441    -0.03322483 -0.01563394
##          PopDens PctUsePubTrans LemasPctOfficDrugUn    murders murdPerPop
## [1,] -0.04120434      0.2089406          -0.1147791 -0.0631018 -0.3421391
##           rapes rapesPerPop   robberies robbbPerPop   assaults assaultPerPop
## [1,] -0.1195219  -0.4394778 -0.04450215  -0.2612296 -0.0710251    -0.3961708
##       burglaries burglPerPop  larcenies larcPerPop  autoTheft autoTheftPerPop
## [1,] -0.09694304  -0.4114195 -0.1011279 -0.4521798 -0.0559635      -0.1641204
##           arsons arsonsPerPop  Viol.Rate nonViol.Rate
## [1,] -0.06902773   -0.2295056 -0.3963943   -0.4652496
```

The variable ***Viol.Rate*** has a moderate to strong positive correlation with all the crime types variables in the form of per 100K population and the following predictor variables:

- *racepctblack* (the percentage of population that is african american, *0.63*)
- *pctWPubAsst* (the percentage of households with public assistance income, *0.56*)
- *PctPopUnderPov* (the percentage of people under the poverty level, *0.5*)
- *MalePctDivorce*, *FemalePctDiv*, *TotalPctDiv* (the percentage of males/females/total who are divorced, *0.51*, *0.54*, *0.54*)
- *PctKidsBornNeverMar* (the percentage of kids born to never married, *0.74*);

and a moderate negative correlation with:

- *racePctWhite* (the percentage of population that is caucasian, *-0.68*)
- *pctWInvInc* (the percentage of households with investment / rent income, *-0.56*)
- *PctFam2Par* (the percentage of families (with kids) that are headed by two parents, *-0.7*)
- *PctKids2Par* (the percentage of kids in family housing with two parents, *-0.73*)
- *PctYoungKids2Par* (the percent of kids 4 and under in two parent household, *-0.66*)
- *PctTeen2Par* (the percent of kids age 12-17 in two parent households, *-0.65*)
- *PctPersOwnOccup* (the percent of people in owner occupied households, *-0.51*).

The variable ***nonViol.Rate*** has a moderate to strong positive correlation with most of the crime types variables in the form of per 100K population, especially with *burglPerPop* (the number of burglaries per 100K population, *0.81*) and *larcPerPop* (the number of larcenies per 100K population, *0.94*) and the following predictor variables:

- *PctPopUnderPov* (the percentage of people under the poverty level, *0.51*)
- *MalePctDivorce*, *FemalePctDiv*, *TotalPctDiv* (the percentage of males/females/total who are divorced, *0.59*, *0.6*, *0.6*)
- *PctKidsBornNeverMar* (the percentage of kids born to never married, *0.55*);

and a moderate negative correlation with:

- *PctFam2Par* (the percentage of families (with kids) that are headed by two parents, *-0.66*)
- *PctKids2Par* (the percentage of kids in family housing with two parents, *-0.67*)

- *PctYoungKids2Par* (the percent of kids 4 and under in two parent household, *-0.61*)
- *PctTeen2Par* (the percent of kids age 12-17 in two parent households, *-0.62*)
- *PctPersOwnOccup* (the percent of people in owner occupied households, *-0.5*).

The variable ***PctBSorMore*** has a moderate positive linear association with:

- *medIncome* (the median household income, *0.68*)
- *pctWInvInc* (the percentage of households with investment / rent income, *0.74*)
- *medFamInc* (the median family income - differs from household income for non-family households, *0.77*)
- *perCapInc* (per capita income, *0.77*)
- *whitePerCap* (per capita income for caucasians, *0.77*)
- *PctEmplProfServ* (the percentage of people 16 and over who are employed in professional services, *0.59*)
- *PctOccupMgmtProf* (the percentage of people 16 and over who are employed in management or professional occupations, *0.95*)
- *OwnOccLowQuart*, *OwnOccMedVal*, *OwnOccHiQuart* (owner occupied housing - lower/median/upper quartile value, *0.6*, *0.63*, *0.65*)
- *OwnOccQrange* (owner occupied housing - difference between upper quartile and lower quartile values, *0.62*)
- *RentLowQ*, *RentMedian*, *RentHighQ* (rental housing - lower/median/upper quartile rent, *0.54*, *0.58*, *0.6*)
- *MedRent* (the median gross rent, *0.57*)

and a moderate negative correlation with:

- *pctWPubAsst* (the percentage of households with public assistance income, *-0.56*)
- *PctLess9thGrade* (the percentage of people 25 and over with less than a 9th grade education, *-0.58*)
- *PctNotHSGrad* (the percentage of people 25 and over that are not high school graduates, *-0.75*)
- *PctUnemployed* (the percentage of people 16 and over, in the labor force, and unemployed, *-0.55*)
- *PctOccupManu* (the percentage of people 16 and over who are employed in manufacturing, *-0.77*)
- *PctHousNoPhone* (the percent of occupied housing units without phone, *-0.52*)

All the correlation coefficients between *PctBSorMore* and all the crime variables are relatively small and negative.

The variable ***medIncome*** has a moderate to strong positive correlation with:

- *pctWWage* (the percentage of households with wage or salary income, *0.58*)
- *pctWInvInc* (the percentage of households with investment / rent income, *0.75*)
- *medFamInc* (the median family income, *0.98*)
- *perCapInc* (per capita income, *0.89*)
- *whitePerCap* (per capita income for caucasians, *0.84*)
- *blackPerCap* (per capita income for african americans, *0.52*)
- *HispPerCap* (per capita income for people with hispanic heritage, *0.63*)
- *PctBSorMore* (the percentage of people 25 and over with a bachelors degree or higher education, *0.68*)
- *PctEmploy* (the percentage of people 16 and over who are employed, *0.6*)
- *PctOccupMgmtProf* (the percentage of people 16 and over who are employed in management or professional occupations, *0.73*)
- *PctFam2Par* (the percentage of families (with kids) that are headed by two parents, *0.72*)
- *PctKids2Par* (the percentage of kids in family housing with two parents, *0.7*)
- *PctYoungKids2Par* (the percent of kids 4 and under in two parent household, *0.7*)
- *PctTeen2Par* (the percent of kids age 12-17 in two parent households, *0.61*)
- *PctPersOwnOccup* (the percent of people in owner occupied households, *0.62*)
- *PctHousOwnOcc* (the percent of households owner occupied, *0.59*)
- *OwnOccLowQuart*, *OwnOccMedVal*, *OwnOccHiQuart* (owner occupied housing - lower/median/upper quartile value, *0.8*, *0.79*, *0.78*)

- *OwnOccQrange* (owner occupied housing - difference between upper quartile and lower quartile values, *0.61*)
- *RentLowQ*, *RentMedian*, *RentHighQ* (rental housing - lower/median/upper quartile rent, *0.81*, *0.85*, *0.85*) 261
- *RentQrange* (rental housing - difference between upper quartile and lower quartile rent, *0.62*)
- *MedRent* (the median gross rent, *0.86*)

and a moderate negative linear association with:

- *pctWPubAsst* (the percentage of households with public assistance income, *-0.63*)
- *PctPopUnderPov* (the percentage of people under the poverty level, *-0.76*)
- *PctLess9thGrade* (the percentage of people 25 and over with less than a 9th grade education, *-0.54*)
- *PctNotHSGrad* (the percentage of people 25 and over that are not high school graduates, *-0.66*)
- *PctUnemployed* (the percentage of people 16 and over, in the labor force, and unemployed, *-0.62*)
- *PctOccupManu* (the percentage of people 16 and over who are employed in manufacturing, *-0.6*)
- *MalePctDivorce*, *FemalePctDiv*, *TotalPctDiv* (the percentage of males/females/total who are divorced, *-0.56*, *-0.54*, *-0.56*)
- *PctHousLess3BR* (the percent of housing units with less than 3 bedrooms, *-0.62*)
- *PctHousNoPhone* (the percent of occupied housing units without phone, *-0.7*)

medIncome has a weak to moderate negative linear association with all types of crime variables. The largest values are for *rapesPerPop* (*-0.44*), burglPerPop (*-0.41*), larcPerPop (*-0.45*), and the total *nonViol.Rate* (*-0.47*).

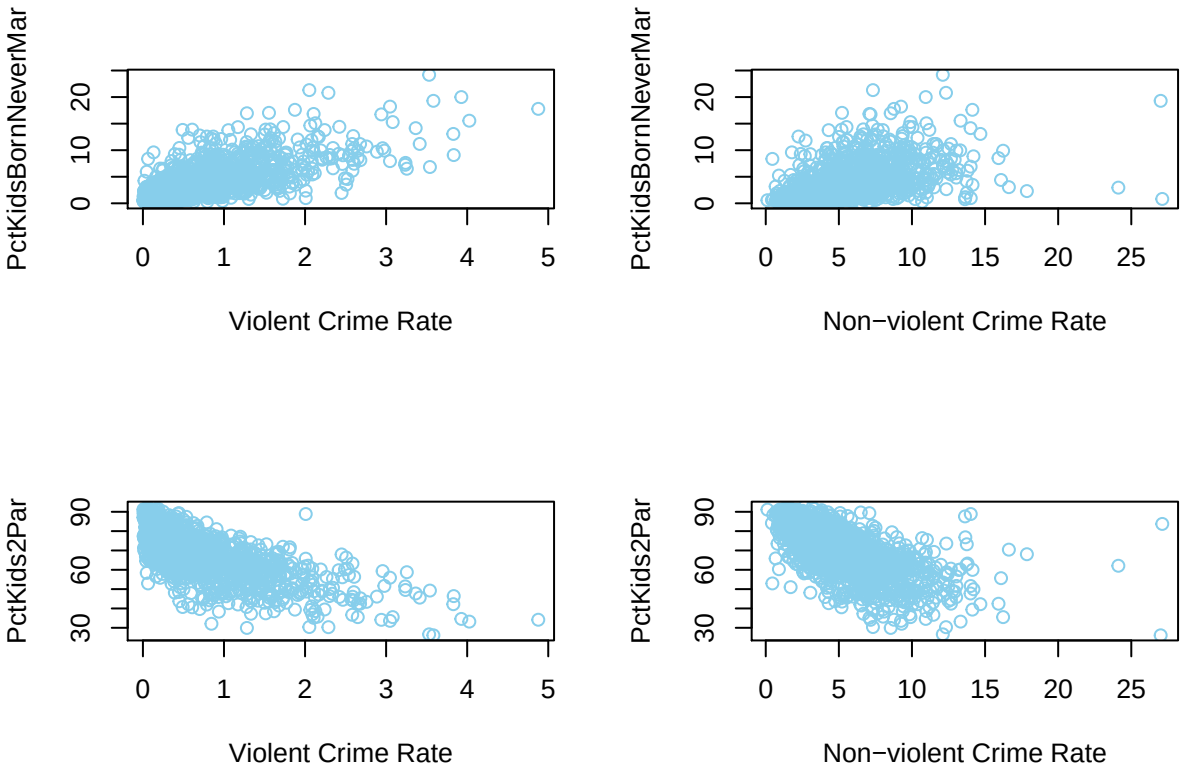
As has been mentioned earlier, the individual crime types will be excluded from the analysis since the information about them is contained in the two response variables.

Also, since *PctBSorMore* and *medIncome* have quite similar pattern in association with other variables, it is probably more interesting to use the education variable *PctBSorMore* for the classification analysis.

Graphics

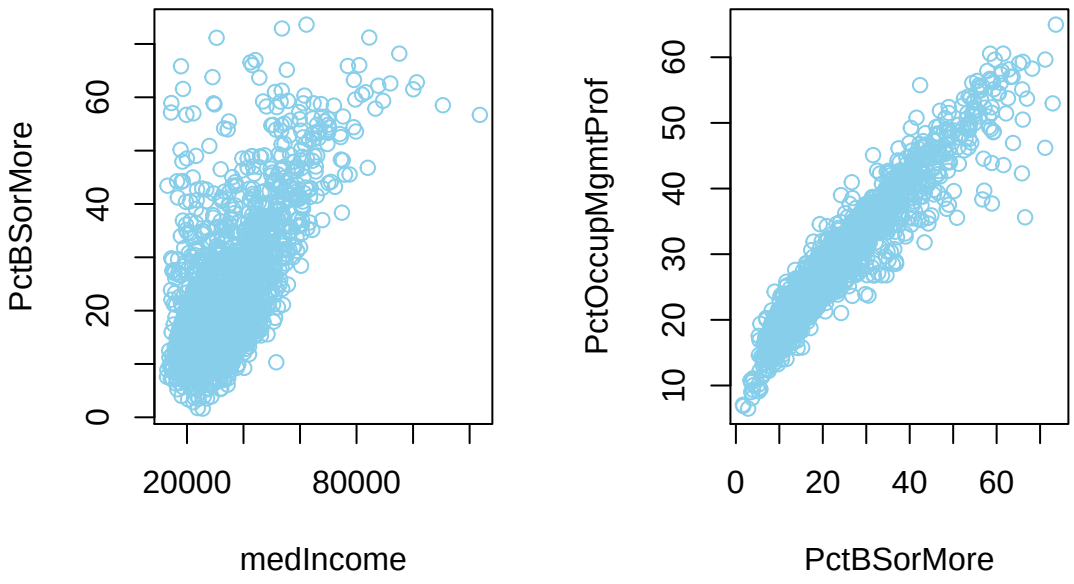
Finally, it may be helpful to visualize the relationships between the response variables and several of the predictors.

```
par(mfrow = c(2, 2))
plot(crimedata3$Viol.Rate, crimedata3$PctKidsBornNeverMar,
     xlab = "Violent Crime Rate",
     ylab = "PctKidsBornNeverMar", col="skyblue")
plot(crimedata3$nonViol.Rate, crimedata3$PctKidsBornNeverMar,
     xlab = "Non-violent Crime Rate",
     ylab = "PctKidsBornNeverMar", col="skyblue")

plot(crimedata3$Viol.Rate, crimedata3$PctKids2Par,
     xlab = "Violent Crime Rate",
     ylab = "PctKids2Par", col="skyblue")
plot(crimedata3$nonViol.Rate, crimedata3$PctKids2Par,
     xlab = "Non-violent Crime Rate",
     ylab = "PctKids2Par", col="skyblue")
```



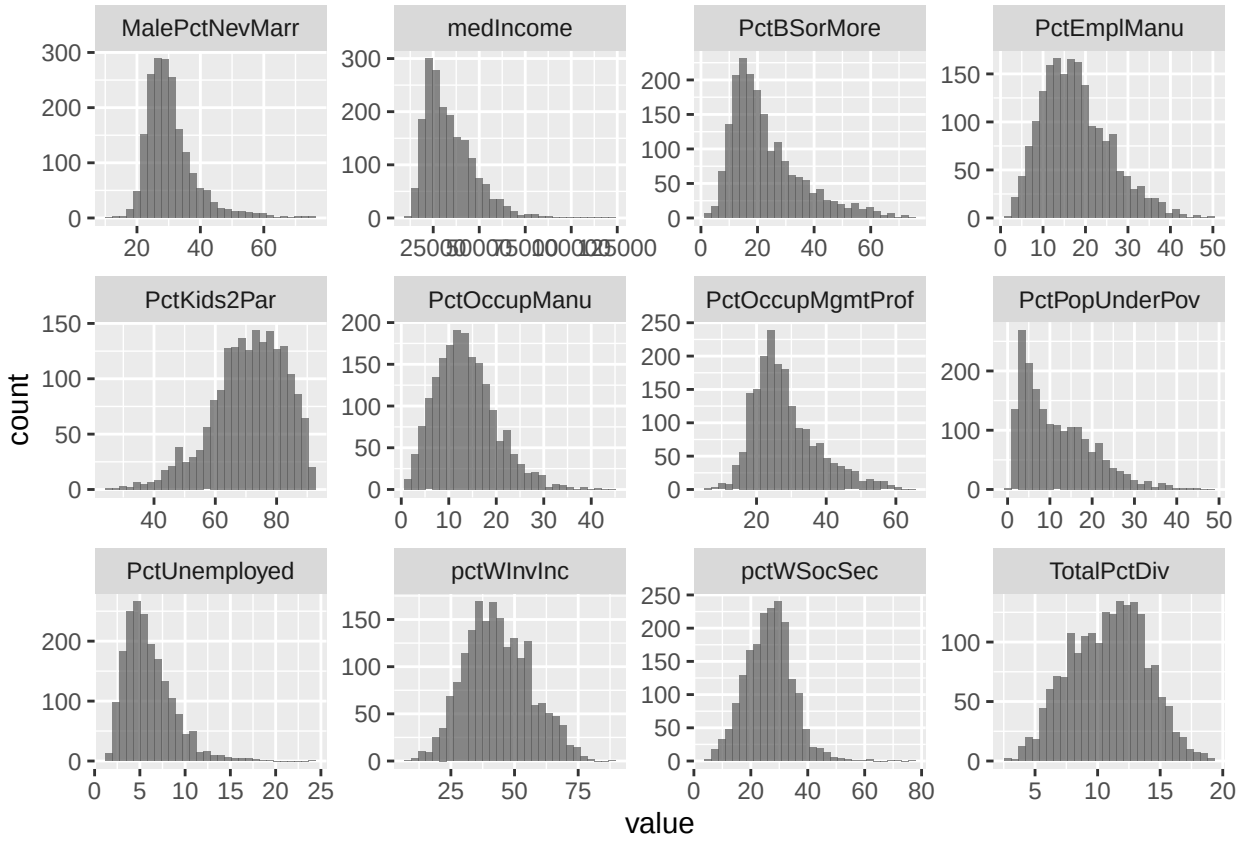
The top two plots display a moderate positive linear relationship between *Violent Crime Rate* and *PctKidsBornNeverMar* as well as between *Non-violent Crime Rate* and *PctKidsBornNeverMar* (the percentage of kids born to never married). The bottom two plots display a moderate negative linear association between *Violent Crime Rate* and *PctKids2Par* as well as between *Non-violent Crime Rate* and *PctKids2Par* (the percentage of kids in family housing with two parents).

```
par(mfrow = c(1, 2))
plot(crimedata3$medIncome, crimedata3$PctBSorMore,
     xlab = "medIncome",
     ylab = "PctBSorMore", col="skyblue")
plot(crimedata3$PctBSorMore, crimedata3$PctOccupMgmtProf,
     xlab = "PctBSorMore",
     ylab = "PctOccupMgmtProf", col="skyblue")
```

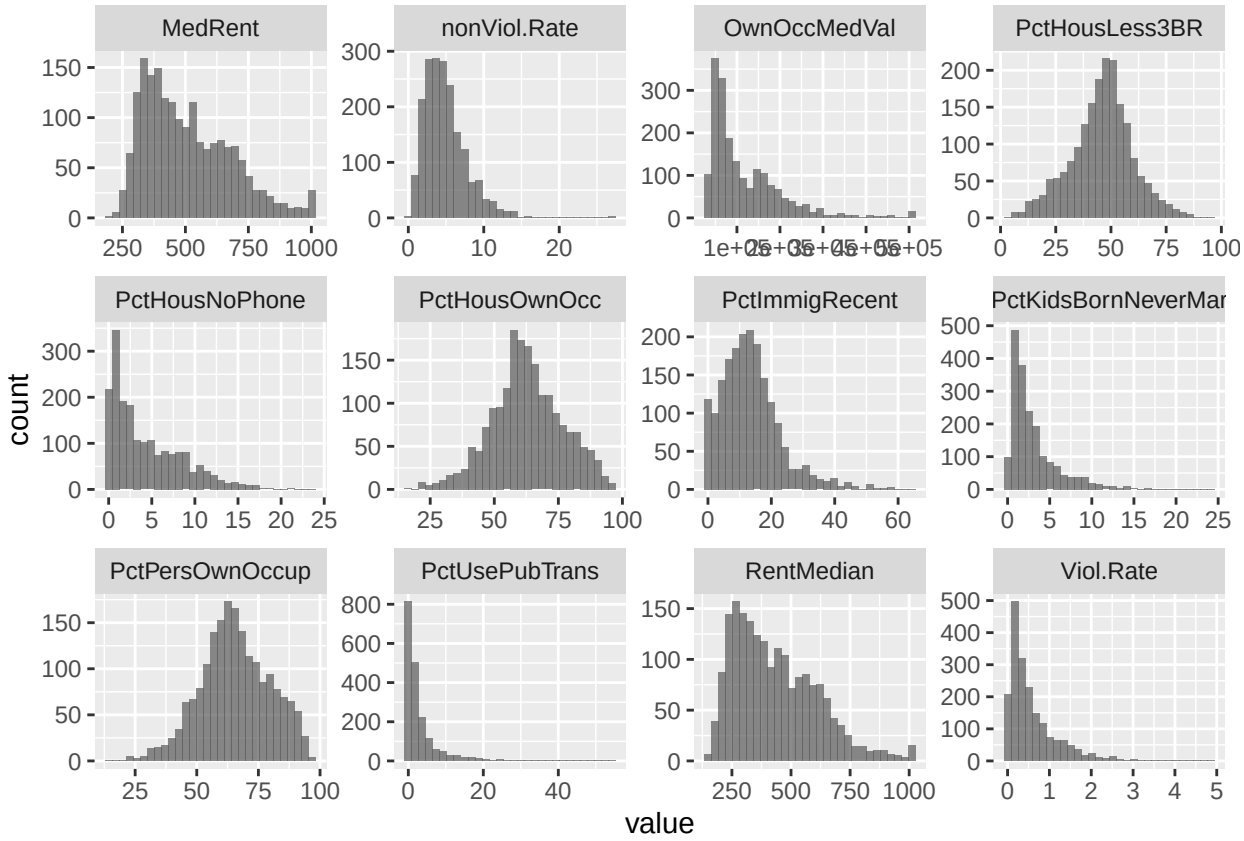


The first plot shows that the two variables *PctBSorMore* and *medIncome* have a moderate to strong linear relationship. The second plot displays a strong linear association between *PctBSorMore* and *PctOccupMgmtProf* (the percentage of people 16 and over who are employed in management or professional occupations).

```
library(ggplot2)
library(tidyr)

# Histograms for some variables
crimedata.h1 <- crimedata3[ c(14, 17, 18,
                              30, 33, 34,
                              36, 38, 39,
                              41, 43, 46) ]
crimedata.h1 %>%
  gather(key, value) %>%
  ggplot(aes(x = value)) +
  facet_wrap(~ key, scales = 'free') +
  geom_histogram(alpha = 0.7)
```

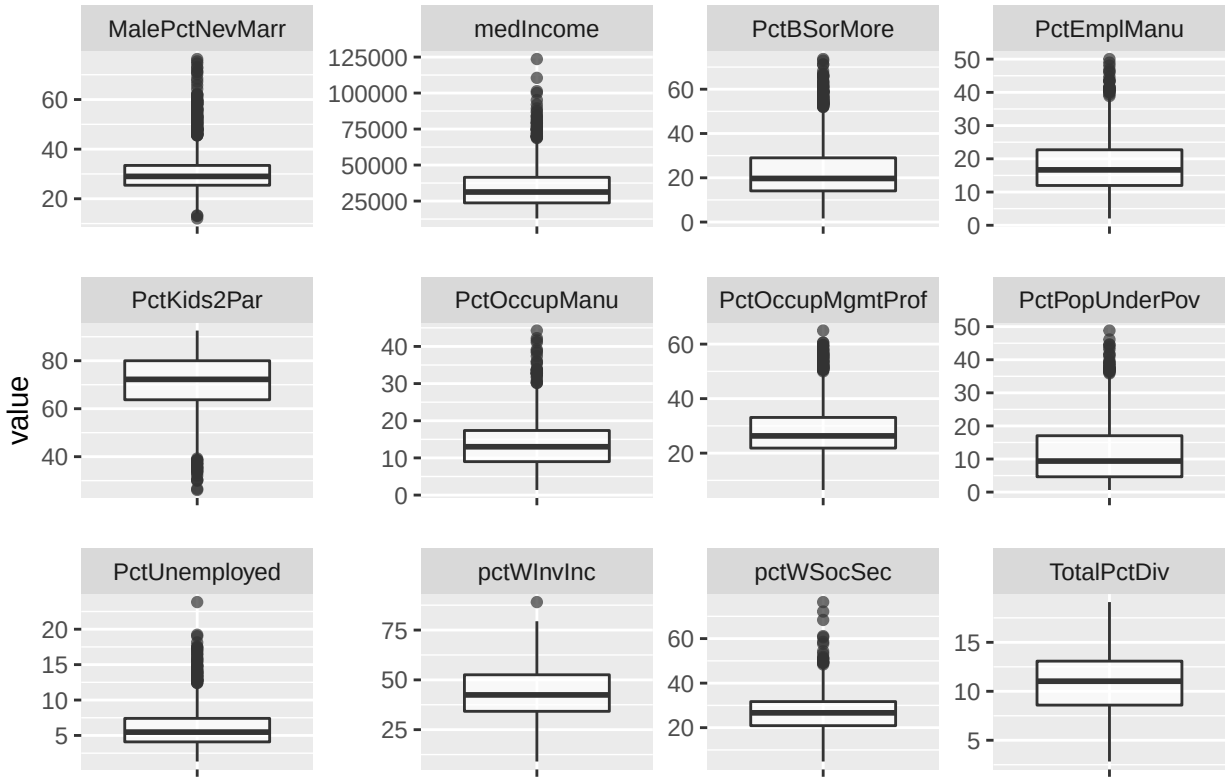



```
crimedata.h2 <- crimedata3[ c(52, 54, 69, 71,
                              75, 79, 82, 86,
                              89, 102, 120, 121)]
crimedata.h2 %>%
  gather(key, value) %>%
  ggplot(aes(x = value)) +
  facet_wrap(~ key, scales = 'free') +
  geom_histogram(alpha = 0.7)
```

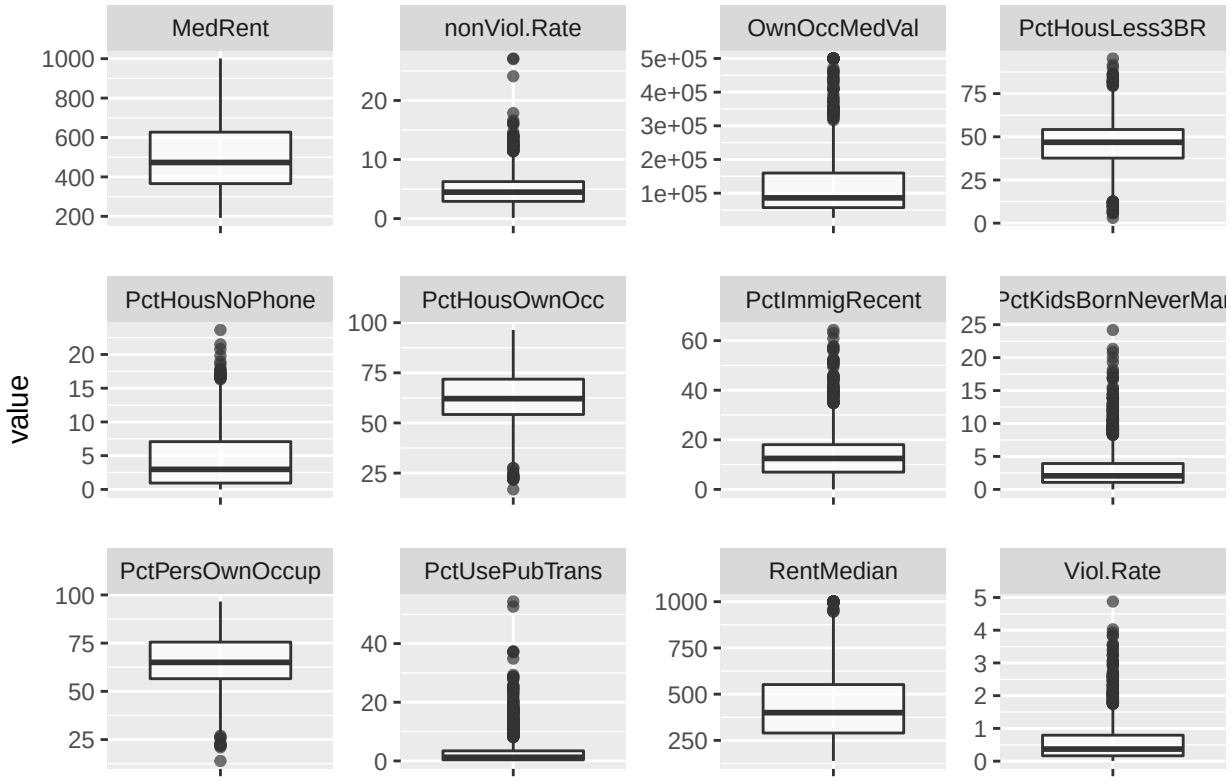


Histograms graphically summarize the distribution of a variable. Most of the displayed variables' distributions are somewhat skewed either to the right or to the left. Several variables appear to be approximately symmetric (for instance, *pctWSocSec* and *PctHousLess3BR*).

```
# Boxplots for some variables
crimedata.h1 %>%
  gather(key, value) %>%
  ggplot(aes(y = value, x = '')) +
  facet_wrap(~ key, scales = 'free') +
  geom_boxplot(alpha = 0.7) +
  labs(x = '')
```




```
crimedata.h2 %>%
  gather(key, value) %>%
  ggplot(aes(y = value, x = '')) +
  facet_wrap(~ key, scales = 'free') +
  geom_boxplot(alpha = 0.7) +
  labs(x = '')
```



Boxplots display mostly skewed data with outliers, including both of the response variables *Violent Crime Rate* and *Non-violent Crime Rate*, *PctKidsBornNeverMar*, *PctUsePubTrans*, and *medIncome* among others.

Regression Analysis

The individual crime variables and the variable *state* are excluded from the analysis. I also exclude *numbUrban*, *NumUnderPov* and keep *pctUrban* and *PctPopUnderPov*.

```
# When trying OLS two variables: OwnOccQrange (84) and RentQrange (88)
# are not defined because of singularity.
# The variables are not linearly independent.
# I remove the variables that are giving NA
# and obtain the same result for the rest of the variables.
# This is because the information given by those two variables is
# already contained in the other variables and thus redundant.
crimedata4 <- crimedata3[ -c(1, 12, 29, 84, 88, 104:119)]
#names(crimedata4)
dim(crimedata4)
## [1] 1901  100
sum(is.na(crimedata4))
## [1] 0
```

Since there is still a large number of the predictor variables, I will apply the *OLS* to gain a general idea about the model fit and two other methods - stepwise selection, and the lasso regression - as they have the ability of selecting the parameters to include in the final model.

Standardizing the response and the predictors, which are all continuous variables:

```
# Standardize the data
crimedata4.std <- data.frame(scale(crimedata4))
#quick check
summary(crimedata4.std[ c(30, 43, 99, 100)])
```

```
##   PctBSorMore       PctKids2Par        Viol.Rate        nonViol.Rate
##  Min.   :-1.7058   Min.   :-3.7906   Min.   :-0.9485   Min.   :-1.7311
##  1st Qu.:-0.7136   1st Qu.:-0.6162   1st Qu.:-0.6902   1st Qu.:-0.7277
##  Median :-0.2665   Median : 0.1010   Median :-0.3524   Median :-0.1658
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.4754   3rd Qu.: 0.7566   3rd Qu.: 0.3439   3rd Qu.: 0.4753
##  Max.   : 4.0322   Max.   : 1.8181   Max.   : 7.0564   Max.   : 7.9585
```

Let's first consider the non-violent crime rate as the response variable.

The response variable: *nonViol.Rate* (non-violent crime rate)

```
nv.crimedata4.std <- crimedata4.std[ ,-99]
dim(nv.crimedata4.std)
```

```
## [1] 1901   99
```

```
set.seed(7)
n  <- dim(nv.crimedata4.std)[1]
ID <- sample(1:n, size = 600, replace = FALSE)

training <- nv.crimedata4.std[-ID,]
```

```
testing  <- nv.crimedata4.std[ID,]
```

The step above performs a random split of the data into training and testing sets, leaving 600 observations (approximately 30% of the data) in the testing set.

Linear Model using Least Squares (OLS)

Fitting a linear model using the OLS on the training set and reporting the test error (MSE):

```
ols.fit <- lm(nonViol.Rate ~ ., data = training)
summary(ols.fit)
```

```
##
## Call:
## lm(formula = nonViol.Rate ~ ., data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3044 -0.3760 -0.0617  0.2784  8.6816
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.010643   0.019403   0.549 0.583437
## population           -0.333337   0.192634  -1.730 0.083813 .
## householdsize        -0.164708   0.096151  -1.713 0.086968 .
## racepctblack         -0.070773   0.105988  -0.668 0.504421
## racePctWhite         -0.165749   0.119341  -1.389 0.165130
## racePctAsian          0.037024   0.058140   0.637 0.524375
## racePctHisp          -0.079221   0.105389  -0.752 0.452376
## agePct12t21           0.401605   0.143281   2.803 0.005146 **
## agePct12t29           0.059687   0.192792   0.310 0.756923
## agePct16t24          -0.550012   0.274111  -2.007 0.045024 *
## agePct65up            0.189138   0.126722   1.493 0.135820
## pctUrban              0.019042   0.028740   0.663 0.507738
## medIncome            -0.334273   0.224966  -1.486 0.137573
## pctWWage              0.219737   0.113284   1.940 0.052649 .
## pctWFarmSelf         -0.039511   0.026837  -1.472 0.141220
## pctWInvInc            0.098158   0.078203   1.255 0.209663
## pctWSocSec            0.405015   0.119888   3.378 0.000753 ***
## pctWPubAsst           0.045926   0.066043   0.695 0.486947
## pctWRetire           -0.121913   0.041846  -2.913 0.003642 **
## medFamInc             0.061619   0.233104   0.264 0.791561
## perCapInc             0.061582   0.249607   0.247 0.805171
## whitePerCap           0.111590   0.203267   0.549 0.583118
## blackPerCap          -0.033508   0.022055  -1.519 0.128951
## indianPerCap         -0.001987   0.018547  -0.107 0.914721
## AsianPerCap          -0.035045   0.022953  -1.527 0.127066
## OtherPerCap           0.019986   0.023644   0.845 0.398112
## HispPerCap            0.029760   0.029418   1.012 0.311928
## PctPopUnderPov        0.283132   0.100900   2.806 0.005096 **
## PctLess9thGrade      -0.187547   0.103940  -1.804 0.071423 .
## PctNotHSGrad         -0.042920   0.131755  -0.326 0.744664
## PctBSorMore          -0.147326   0.105869  -1.392 0.164305
```

```
## PctUnemployed          0.019321   0.053084   0.364 0.715942
## PctEmploy              0.252764   0.096923   2.608 0.009223 **
## PctEmplManu           -0.067345   0.042382  -1.589 0.112321
## PctEmplProfServ       -0.085340   0.048271  -1.768 0.077324 .
## PctOccupManu           0.059706   0.071461   0.836 0.403602
## PctOccupMgmtProf       0.228746   0.104452   2.190 0.028718 *
## MalePctDivorce         0.501385   0.433508   1.157 0.247675
## MalePctNevMarr         0.302446   0.095469   3.168 0.001574 **
## FemalePctDiv           0.507108   0.522814   0.970 0.332262
## TotalPctDiv           -0.687552   0.934206  -0.736 0.461890
## PersPerFam            -0.238468   0.219841  -1.085 0.278258
## PctFam2Par             0.255046   0.223855   1.139 0.254789
## PctKids2Par           -0.459195   0.215067  -2.135 0.032952 *
## PctYoungKids2Par       0.060746   0.070232   0.865 0.387246
## PctTeen2Par            0.017042   0.054362   0.313 0.753966
## PctWorkMomYoungKids    0.011413   0.052595   0.217 0.828251
## PctWorkMom            -0.054227   0.064050  -0.847 0.397368
## NumKidsBornNeverMar   -0.093234   0.099780  -0.934 0.350284
## PctKidsBornNeverMar    0.094590   0.071803   1.317 0.187976
## NumImmig               0.153141   0.089506   1.711 0.087347 .
## PctImmigRecent         0.029227   0.062121   0.470 0.638086
## PctImmigRec5          -0.141804   0.096831  -1.464 0.143335
## PctImmigRec8           0.170737   0.112482   1.518 0.129301
## PctImmigRec10          0.011618   0.085417   0.136 0.891834
## PctRecentImmig        -0.080463   0.201276  -0.400 0.689402
## PctRecImmig5           0.282289   0.385348   0.733 0.463972
## PctRecImmig8          -0.750071   0.481693  -1.557 0.119697
## PctRecImmig10          0.340157   0.367007   0.927 0.354195
## PctSpeakEnglOnly      -0.120506   0.122829  -0.981 0.326747
## PctNotSpeakEnglWell    0.087882   0.109377   0.803 0.421857
## PctLargHouseFam       -0.098719   0.299865  -0.329 0.742054
## PctLargHouseOccup      0.126799   0.273773   0.463 0.643336
## PersPerOccupHous       0.481431   0.323868   1.487 0.137408
## PersPerOwnOccHous      0.172384   0.202762   0.850 0.395396
## PersPerRentOccHous    -0.397479   0.109615  -3.626 0.000300 ***
## PctPersOwnOccup       -1.654903   0.559220  -2.959 0.003144 **
## PctPersDenseHous      -0.014355   0.110813  -0.130 0.896953
## PctHousLess3BR        -0.095588   0.071821  -1.331 0.183467
## MedNumBR              -0.040715   0.032618  -1.248 0.212189
## HousVacant             0.199950   0.089903   2.224 0.026330 *
## PctHousOccup          -0.041847   0.032488  -1.288 0.197977
## PctHousOwnOcc          1.426574   0.552647   2.581 0.009959 **
## PctVacantBoarded       0.010461   0.032063   0.326 0.744280
## PctVacMore6Mos        -0.076862   0.031005  -2.479 0.013311 *
## MedYrHousBuilt         0.068333   0.043485   1.571 0.116353
## PctHousNoPhone        -0.031809   0.056861  -0.559 0.575984
## PctWOFullPlumb        -0.015010   0.026418  -0.568 0.570016
## OwnOccLowQuart        -0.077317   0.218203  -0.354 0.723150
## OwnOccMedVal           0.288830   0.320887   0.900 0.368249
## OwnOccHiQuart         -0.301282   0.172463  -1.747 0.080904 .
## RentLowQ              -0.231435   0.098678  -2.345 0.019171 *
## RentMedian             0.222389   0.219505   1.013 0.311198
## RentHighQ             -0.194978   0.140469  -1.388 0.165380
## MedRent                0.114576   0.180035   0.636 0.524631
```

```
## MedRentPctHousInc     -0.027125   0.036766  -0.738 0.460791
## MedOwnCostPctInc      -0.075846   0.043159  -1.757 0.079107 .
## MedOwnCostPctIncNoMtg  0.011825   0.031374   0.377 0.706300
## NumInShelters          0.088122   0.083151   1.060 0.289458
## NumStreet              0.013141   0.090748   0.145 0.884888
## PctForeignBorn         0.195914   0.150074   1.305 0.191989
## PctBornSameState      -0.096934   0.056324  -1.721 0.085505 .
## PctSameHouse85        -0.063519   0.070033  -0.907 0.364591
## PctSameCity85          0.031139   0.052079   0.598 0.550016
## PctSameState85         0.073009   0.058231   1.254 0.210169
## LandArea              -0.032922   0.019138  -1.720 0.085645 .
## PopDens               -0.188986   0.037824  -4.997  6.7e-07 ***
## PctUsePubTrans         0.033735   0.037017   0.911 0.362294
## LemasPctOfficDrugUn    0.045932   0.021707   2.116 0.034549 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6927 on 1202 degrees of freedom
## Multiple R-squared:  0.5853, Adjusted R-squared:  0.5515
## F-statistic: 17.31 on 98 and 1202 DF,  p-value: < 2.2e-16
```

```
ols.test <- testing[, 'nonViol.Rate'] - predict(ols.fit,
                      newdata = testing, type = 'response')
# test MSE
ols.test.MSE <- mean(ols.test**2)
ols.test.MSE
```

```
## [1] 0.362435
```

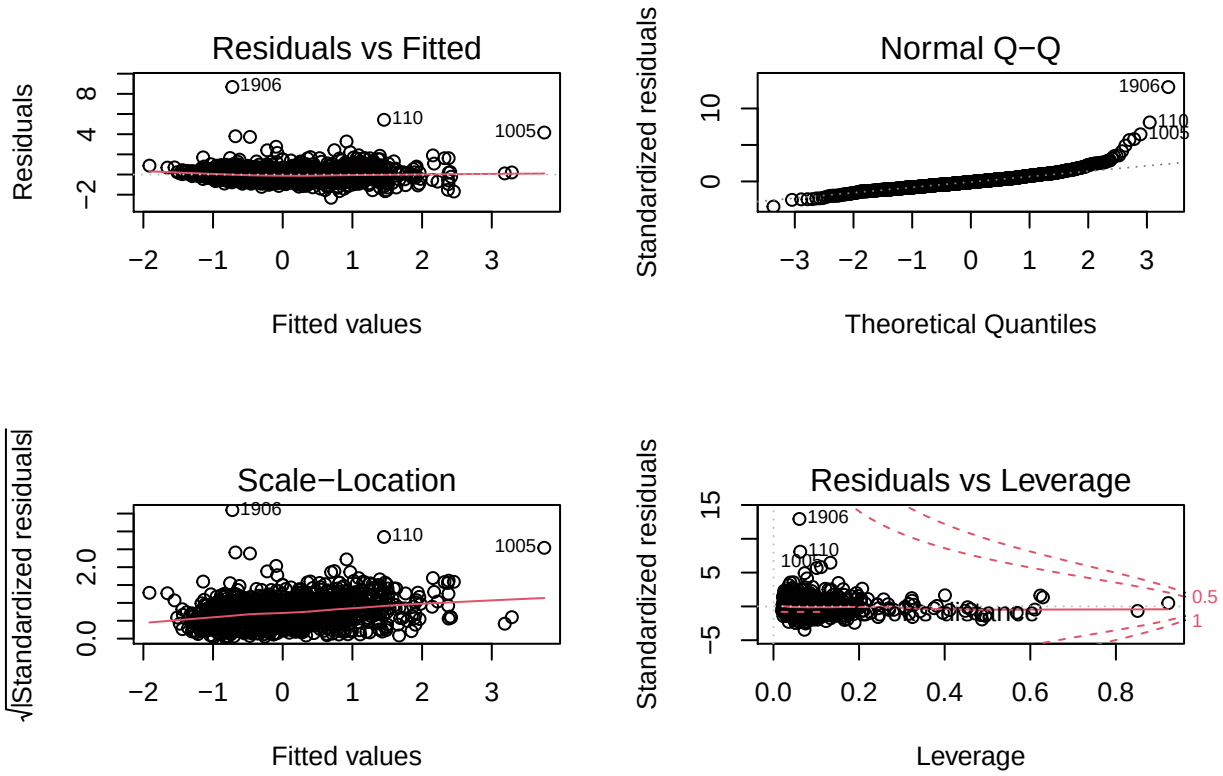
The *OLS* model determines a total of 17 predictors to be statistically significant at 0.05 level of significance:

- *pctWSocSec* the percentage of households with social security income;
- *PersPerRentOccHous* - mean persons per rental household;
- *PopDens* - population density in persons per square mile;
- *agePct12t21* - the percentage of population that is 12-21 in age;
- *pctWRetire* - the percentage of households with retirement income;
- *PctPopUnderPov* - the percentage of people under the poverty level;
- *PctEmploy* - the percentage of people 16 and over who are employed;
- *MalePctNevMarr* - the percentage of males who have never married;
- *PctPersOwnOccup* - the percent of people in owner occupied households;
- *PctHousOwnOcc* - the percent of households owner occupied;
- *agePct16t24* - the percentage of population that is 16-24 in age;
- *PctOccupMgmtProf* - the percentage of people 16 and over who are employed in management or professional occupations;
- *PctKids2Par* - the percentage of kids in family housing with two parents;
- *HousVacant* - the number of vacant households;
- *PctVacMore6Mos* - the percent of vacant housing that has been vacant more than 6 months;
- *RentLowQ* - rental housing - lower quartile rent;
- *LemasPctOfficDrugUn* - the percent of officers assigned to drug units.

For this model, *test MSE* value is 0.362. The $R^2_{adj} = 0.552$ of the *OLS* suggests that a more flexible model than a linear one may work better for these data.

Assumptions Check

```
par(mfrow = c(2,2))
plot(ols.fit)
```



The residuals vs. fitted values plot (as well as the third plot) should look more or less random, however it shows some outliers. The normal probability plot is not ideal either as several points deviate from the straight line. The last plot (Cook's distance) reveals which points have the greatest influence on the regression (leverage points).

Removing three leverage points improves the model performance:

```
r <- abs(ols.fit$residuals)
order((r), decreasing = TRUE)[1:3]
```

```
## [1] 1099   65  587
```

```
training <- training[ -c(1099, 65, 587),]

ols.fit <- lm(nonViol.Rate ~ ., data = training)
summary(ols.fit)
```

```
##
## Call:
```

```
## lm(formula = nonViol.Rate ~ ., data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3379 -0.3769 -0.0506  0.2934  3.7830
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -0.005629   0.017097  -0.329 0.742011
## population            -0.240818   0.169629  -1.420 0.155962
## householdsize         -0.181296   0.084671  -2.141 0.032460 *
## racepctblack          -0.044995   0.093318  -0.482 0.629775
## racePctWhite          -0.138641   0.105039  -1.320 0.187120
## racePctAsian           0.016102   0.051195   0.315 0.753174
## racePctHisp           -0.042503   0.092781  -0.458 0.646965
## agePct12t21            0.461362   0.126178   3.656 0.000267 ***
## agePct12t29            0.142600   0.169750   0.840 0.401043
## agePct16t24           -0.664419   0.241303  -2.753 0.005986 **
## agePct65up             0.135542   0.111554   1.215 0.224593
## pctUrban               0.022373   0.025292   0.885 0.376548
## medIncome             -0.271279   0.199147  -1.362 0.173388
## pctWWage               0.258670   0.099723   2.594 0.009606 **
## pctWFarmSelf          -0.037313   0.023624  -1.579 0.114496
## pctWInvInc             0.114732   0.068830   1.667 0.095798 .
## pctWSocSec             0.434074   0.105566   4.112 4.19e-05 ***
## pctWPubAsst            0.069226   0.058165   1.190 0.234218
## pctWRetire            -0.122497   0.036863  -3.323 0.000917 ***
## medFamInc              0.029429   0.206422   0.143 0.886656
## perCapInc             -0.032973   0.219740  -0.150 0.880747
## whitePerCap            0.176268   0.178923   0.985 0.324743
## blackPerCap           -0.033993   0.019409  -1.751 0.080134 .
## indianPerCap          -0.003100   0.016326  -0.190 0.849453
## AsianPerCap           -0.019843   0.020246  -0.980 0.327230
## OtherPerCap            0.031105   0.020817   1.494 0.135381
## HispPerCap             0.043530   0.025900   1.681 0.093080 .
## PctPopUnderPov         0.289981   0.088798   3.266 0.001123 **
## PctLess9thGrade       -0.174863   0.091559  -1.910 0.056392 .
## PctNotHSGrad          -0.019949   0.116178  -0.172 0.863694
## PctBSorMore           -0.168304   0.093203  -1.806 0.071203 .
## PctUnemployed          0.016581   0.046719   0.355 0.722718
## PctEmploy              0.195544   0.085348   2.291 0.022128 *
## PctEmplManu           -0.086056   0.037364  -2.303 0.021439 *
## PctEmplProfServ       -0.055338   0.042517  -1.302 0.193322
## PctOccupManu           0.122335   0.063203   1.936 0.053152 .
## PctOccupMgmtProf       0.279328   0.092002   3.036 0.002448 **
## MalePctDivorce         0.593691   0.382470   1.552 0.120865
## MalePctNevMarr         0.242436   0.084191   2.880 0.004053 **
## FemalePctDiv           0.632328   0.460978   1.372 0.170410
## TotalPctDiv           -0.930297   0.823920  -1.129 0.259077
## PersPerFam            -0.271454   0.193522  -1.403 0.160965
## PctFam2Par             0.233286   0.197621   1.180 0.238047
## PctKids2Par           -0.364990   0.189797  -1.923 0.054709 .
## PctYoungKids2Par       0.031527   0.061835   0.510 0.610250
## PctTeen2Par           -0.014947   0.047964  -0.312 0.755381
```

```
## PctWorkMomYoungKids    0.023019   0.046312   0.497 0.619249
## PctWorkMom            -0.048008   0.056369  -0.852 0.394560
## NumKidsBornNeverMar   -0.099882   0.088074  -1.134 0.256991
## PctKidsBornNeverMar    0.052176   0.063481   0.822 0.411293
## NumImmig               0.103064   0.078884   1.307 0.191625
## PctImmigRecent         0.042944   0.054687   0.785 0.432448
## PctImmigRec5          -0.156622   0.085227  -1.838 0.066355 .
## PctImmigRec8           0.135701   0.099010   1.371 0.170763
## PctImmigRec10          0.053945   0.075204   0.717 0.473316
## PctRecentImmig        -0.039619   0.177400  -0.223 0.823316
## PctRecImmig5           0.262390   0.339691   0.772 0.440008
## PctRecImmig8          -0.830907   0.424009  -1.960 0.050269 .
## PctRecImmig10          0.429246   0.323051   1.329 0.184191
## PctSpeakEnglOnly      -0.052654   0.108161  -0.487 0.626483
## PctNotSpeakEnglWell    0.056294   0.096327   0.584 0.559058
## PctLargHouseFam       -0.012776   0.264072  -0.048 0.961422
## PctLargHouseOccup      0.064591   0.241092   0.268 0.788814
## PersPerOccupHous       0.447620   0.285393   1.568 0.117044
## PersPerOwnOccHous      0.109037   0.178597   0.611 0.541633
## PersPerRentOccHous    -0.247743   0.096952  -2.555 0.010732 *
## PctPersOwnOccup       -1.143569   0.493021  -2.320 0.020535 *
## PctPersDenseHous      -0.044648   0.097531  -0.458 0.647194
## PctHousLess3BR        -0.062342   0.063342  -0.984 0.325212
## MedNumBR              -0.033896   0.028720  -1.180 0.238150
## HousVacant             0.187188   0.079120   2.366 0.018146 *
## PctHousOccup          -0.049015   0.028619  -1.713 0.087031 .
## PctHousOwnOcc          0.915758   0.487350   1.879 0.060479 .
## PctVacantBoarded       0.017118   0.028250   0.606 0.544648
## PctVacMore6Mos        -0.091845   0.027315  -3.362 0.000797 ***
## MedYrHousBuilt         0.055222   0.038335   1.440 0.149990
## PctHousNoPhone        -0.019210   0.050053  -0.384 0.701194
## PctWOFullPlumb        -0.008779   0.023253  -0.378 0.705828
## OwnOccLowQuart        -0.201479   0.192161  -1.048 0.294625
## OwnOccMedVal           0.405923   0.282477   1.437 0.150975
## OwnOccHiQuart         -0.309849   0.151828  -2.041 0.041490 *
## RentLowQ              -0.164996   0.086916  -1.898 0.057891 .
## RentMedian             0.131405   0.193378   0.680 0.496935
## RentHighQ             -0.213327   0.123713  -1.724 0.084899 .
## MedRent                0.235161   0.158769   1.481 0.138827
## MedRentPctHousInc     -0.007044   0.032397  -0.217 0.827912
## MedOwnCostPctInc      -0.072509   0.038063  -1.905 0.057022 .
## MedOwnCostPctIncNoMtg  0.009180   0.027613   0.332 0.739621
## NumInShelters          0.078505   0.073251   1.072 0.284060
## NumStreet              0.010189   0.079873   0.128 0.898514
## PctForeignBorn         0.227312   0.132084   1.721 0.085515 .
## PctBornSameState      -0.101688   0.049571  -2.051 0.040449 *
## PctSameHouse85        -0.008997   0.061741  -0.146 0.884169
## PctSameCity85         -0.001431   0.045880  -0.031 0.975119
## PctSameState85         0.070780   0.051256   1.381 0.167558
## LandArea              -0.032361   0.016842  -1.921 0.054918 .
## PopDens               -0.157529   0.033415  -4.714 2.71e-06 ***
## PctUsePubTrans        -0.010498   0.032882  -0.319 0.749593
## LemasPctOfficDrugUn    0.056603   0.019123   2.960 0.003138 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6096 on 1199 degrees of freedom
## Multiple R-squared:  0.6341, Adjusted R-squared:  0.6042
## F-statistic: 21.21 on 98 and 1199 DF,  p-value: < 2.2e-16
```

```
ols.test <- testing[, 'nonViol.Rate'] - predict(ols.fit,
                      newdata = testing, type = 'response')
# test MSE
ols.test.MSE <- mean(ols.test**2)
ols.test.MSE
```

```
## [1] 0.3544323
```

The *test MSE* value has decreased from 0.362 to 0.354. The $R^2_{adj} = 0.604$.

Stepwise Selection

Applying the *Stepwise* selection to the above model with *AIC* and reporting the final model and the obtained test error:

```
step.fit <- step(ols.fit, direction = 'both', trace = 0, k = 2)
# k = 2 gives the genuine AIC
step.test <- testing[, 'nonViol.Rate'] - predict(step.fit,
                       newdata = testing, type = 'response')

# Final model and test error
summary(step.fit)
```

```
##
## Call:
## lm(formula = nonViol.Rate ~ population + householdsize + racePctWhite +
##     agePct12t21 + agePct16t24 + pctUrban + medIncome + pctWWage +
##     pctWFarmSelf + pctWInvInc + pctWSocSec + pctWPubAsst + pctWRetire +
##     whitePerCap + blackPerCap + OtherPerCap + HispPerCap + PctPopUnderPov +
##     PctLess9thGrade + PctBSorMore + PctEmploy + PctEmplManu +
##     PctEmplProfServ + PctOccupManu + PctOccupMgmtProf + MalePctDivorce +
##     MalePctNevMarr + FemalePctDiv + PersPerFam + PctFam2Par +
##     PctKids2Par + PctImmigRec5 + PctImmigRec8 + PctRecImmig8 +
##     PctRecImmig10 + PersPerOccupHous + PersPerRentOccHous + PctPersOwnOccup +
##     HousVacant + PctHousOccup + PctHousOwnOcc + PctVacMore6Mos +
##     MedYrHousBuilt + OwnOccMedVal + OwnOccHiQuart + RentLowQ +
##     RentHighQ + MedRent + MedOwnCostPctInc + PctForeignBorn +
##     PctBornSameState + PctSameState85 + LandArea + PopDens +
##     LemasPctOfficDrugUn, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3297 -0.3625 -0.0539  0.2900  3.9430
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -0.004566   0.016837  -0.271 0.786298
## population          -0.101306   0.050942  -1.989 0.046960 *
```

```
## householdsize       -0.185344   0.069791  -2.656 0.008016 **
## racePctWhite        -0.089939   0.044045  -2.042 0.041363 *
## agePct12t21          0.371356   0.099414   3.735 0.000196 ***
## agePct16t24         -0.506291   0.131128  -3.861 0.000119 ***
## pctUrban             0.037524   0.023197   1.618 0.105990
## medIncome           -0.282140   0.109239  -2.583 0.009914 **
## pctWWage             0.219726   0.089865   2.445 0.014621 *
## pctWFarmSelf        -0.032827   0.022681  -1.447 0.148047
## pctWInvInc           0.130098   0.062390   2.085 0.037253 *
## pctWSocSec           0.453584   0.078166   5.803 8.27e-09 ***
## pctWPubAsst          0.074164   0.048307   1.535 0.124974
## pctWRetire          -0.115457   0.034010  -3.395 0.000708 ***
## whitePerCap          0.174232   0.071421   2.440 0.014846 *
## blackPerCap         -0.031949   0.018401  -1.736 0.082770 .
## OtherPerCap          0.033123   0.020233   1.637 0.101875
## HispPerCap           0.040154   0.024909   1.612 0.107213
## PctPopUnderPov       0.266848   0.070357   3.793 0.000156 ***
## PctLess9thGrade     -0.187822   0.041967  -4.475 8.32e-06 ***
## PctBSorMore         -0.139774   0.083405  -1.676 0.094019 .
## PctEmploy            0.156072   0.062188   2.510 0.012211 *
## PctEmplManu         -0.090589   0.034742  -2.607 0.009231 **
## PctEmplProfServ     -0.070490   0.039631  -1.779 0.075540 .
## PctOccupManu         0.121950   0.057841   2.108 0.035201 *
## PctOccupMgmtProf     0.264106   0.086701   3.046 0.002367 **
## MalePctDivorce       0.164696   0.059248   2.780 0.005522 **
## MalePctNevMarr       0.288161   0.068052   4.234 2.46e-05 ***
## FemalePctDiv         0.086337   0.061051   1.414 0.157560
## PersPerFam          -0.202059   0.110417  -1.830 0.067496 .
## PctFam2Par           0.253117   0.166416   1.521 0.128517
## PctKids2Par         -0.370944   0.165126  -2.246 0.024852 *
## PctImmigRec5        -0.077843   0.051426  -1.514 0.130363
## PctImmigRec8         0.147824   0.055791   2.650 0.008161 **
## PctRecImmig8        -0.622098   0.211362  -2.943 0.003308 **
## PctRecImmig10        0.470250   0.235605   1.996 0.046160 *
## PersPerOccupHous     0.533380   0.152154   3.506 0.000472 ***
## PersPerRentOccHous  -0.197626   0.081473  -2.426 0.015423 *
## PctPersOwnOccup     -0.919394   0.244979  -3.753 0.000183 ***
## HousVacant           0.129022   0.056045   2.302 0.021494 *
## PctHousOccup        -0.054423   0.025537  -2.131 0.033272 *
## PctHousOwnOcc        0.675599   0.231577   2.917 0.003593 **
## PctVacMore6Mos      -0.087493   0.024626  -3.553 0.000395 ***
## MedYrHousBuilt       0.057385   0.032208   1.782 0.075044 .
## OwnOccMedVal         0.169853   0.117655   1.444 0.149088
## OwnOccHiQuart       -0.263634   0.118711  -2.221 0.026543 *
## RentLowQ            -0.140799   0.068199  -2.065 0.039175 *
## RentHighQ           -0.194060   0.108118  -1.795 0.072913 .
## MedRent              0.283524   0.117066   2.422 0.015581 *
## MedOwnCostPctInc    -0.065696   0.031185  -2.107 0.035349 *
## PctForeignBorn       0.270070   0.095638   2.824 0.004821 **
## PctBornSameState    -0.107948   0.045008  -2.398 0.016612 *
## PctSameState85       0.060841   0.042244   1.440 0.150053
## LandArea            -0.033781   0.015722  -2.149 0.031855 *
## PopDens             -0.166891   0.029381  -5.680 1.67e-08 ***
## LemasPctOfficDrugUn  0.054544   0.018086   3.016 0.002615 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.604 on 1242 degrees of freedom
## Multiple R-squared:  0.628,  Adjusted R-squared:  0.6115
## F-statistic: 38.12 on 55 and 1242 DF,  p-value: < 2.2e-16
```

```
step.test.MSE <- mean(step.test**2)
step.test.MSE
```

```
## [1] 0.3514059
```

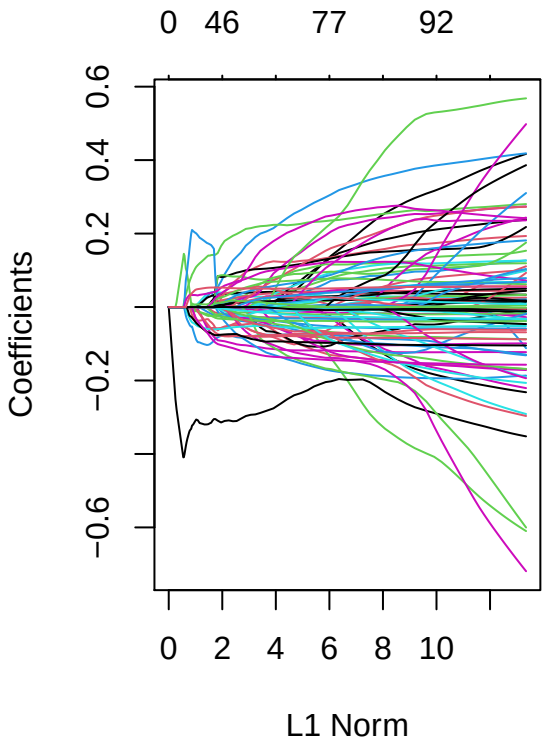
The *Stepwise* selection procedure includes 55 variables in the final model. In addition to the variables listed in the previous section (the *OLS* modeling), there are several variables related to ethnicity, income, employment (i.e., the percentage of people employed in manufacturing/in professional services), education, immigration, marital status (i.e., the percentage of males who are divorced), housing (i.e., the median year housing units built), etc.

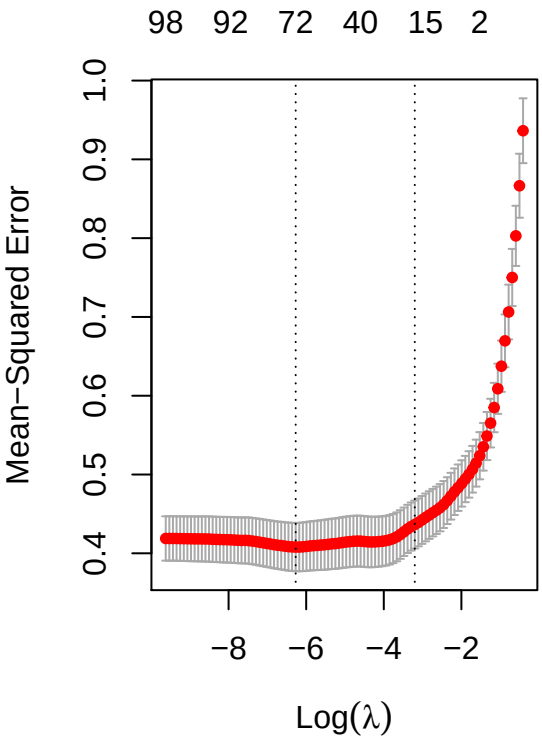
The *test MSE* value is 0.351. The $R^2_{adj} = 0.612$, which is a slight improvement from the *OLS* model.

LASSO Regression Analysis

The LASSO regression analysis on the training data includes plotting the solution path, plotting the cross-validation errors, and selecting the best tuning parameter that minimizes the cross-validation error. The test error is also reported.

```
library(glmnet)

x.train <- model.matrix(nonViol.Rate ~ 0+., data = training)
x.new   <- model.matrix(nonViol.Rate ~ 0+., data = testing)

lasso <- glmnet(x.train, training[,'nonViol.Rate'], alpha = 1,
                standardize = FALSE)
set.seed(7)
# cross-validation to tune the hyper-parameters
cv.lasso <- cv.glmnet(x.train, training[,'nonViol.Rate'], alpha = 1,
                      standardize = FALSE)

par(mfrow = c(1,2))
plot(lasso, 'norm', label = T)
plot(cv.lasso)
```






```
bst_lmd <- cv.lasso$lambda.min

# observe coefficients
coef.lasso <- predict(lasso, x.new,
                      s=bst_lmd, type="coefficient", mode="fraction")
round(coef.lasso, 5)
```

```
## 99 x 1 sparse Matrix of class "dgCMatrix"
##                               1
## (Intercept)           -0.00497
## population            -0.00012
## householdsize         -0.11695
## racepctblack           .
## racePctWhite          -0.07234
## racePctAsian           .
## racePctHisp            .
## agePct12t21            0.09501
## agePct12t29           -0.07966
## agePct16t24           -0.07554
## agePct65up             0.05047
## pctUrban               0.02897
## medIncome             -0.00640
## pctWWage               0.10112
## pctWFarmSelf          -0.02890
## pctWInvInc             0.06432
## pctWSocSec             0.28391
```

```
## pctWPubAsst            0.03273
## pctWRetire            -0.11822
## medFamInc             -0.10779
## perCapInc              .
## whitePerCap            0.10949
## blackPerCap           -0.03170
## indianPerCap          -0.00055
## AsianPerCap           -0.01382
## OtherPerCap            0.04119
## HispPerCap             0.04173
## PctPopUnderPov         0.23173
## PctLess9thGrade       -0.16282
## PctNotHSGrad           .
## PctBSorMore           -0.10215
## PctUnemployed          .
## PctEmploy              0.12777
## PctEmplManu           -0.06216
## PctEmplProfServ       -0.04972
## PctOccupManu           0.07852
## PctOccupMgmtProf       0.17565
## MalePctDivorce         0.12389
## MalePctNevMarr         0.13537
## FemalePctDiv           0.09346
## TotalPctDiv            .
## PersPerFam             .
## PctFam2Par             .
## PctKids2Par           -0.21623
## PctYoungKids2Par       .
## PctTeen2Par           -0.02012
## PctWorkMomYoungKids    .
## PctWorkMom            -0.00006
## NumKidsBornNeverMar   -0.09190
## PctKidsBornNeverMar    0.02761
## NumImmig               .
## PctImmigRecent         0.01201
## PctImmigRec5          -0.07272
## PctImmigRec8           .
## PctImmigRec10          0.11736
## PctRecentImmig         .
## PctRecImmig5           .
## PctRecImmig8          -0.15834
## PctRecImmig10          .
## PctSpeakEnglOnly      -0.00622
## PctNotSpeakEnglWell    0.03863
## PctLargHouseFam        .
## PctLargHouseOccup      0.01828
## PersPerOccupHous       0.18756
## PersPerOwnOccHous     -0.06136
## PersPerRentOccHous     .
## PctPersOwnOccup       -0.13349
## PctPersDenseHous       .
## PctHousLess3BR        -0.07091
## MedNumBR              -0.03780
## HousVacant             0.07266
```

```
## PctHousOccup          -0.05704
## PctHousOwnOcc          .
## PctVacantBoarded       0.01883
## PctVacMore6Mos        -0.08126
## MedYrHousBuilt         0.07525
## PctHousNoPhone         0.00442
## PctWOFullPlumb         .
## OwnOccLowQuart         .
## OwnOccMedVal           .
## OwnOccHiQuart         -0.06506
## RentLowQ              -0.07374
## RentMedian             .
## RentHighQ             -0.05859
## MedRent                0.07691
## MedRentPctHousInc      .
## MedOwnCostPctInc      -0.07581
## MedOwnCostPctIncNoMtg  .
## NumInShelters          0.02011
## NumStreet              0.01520
## PctForeignBorn         0.23366
## PctBornSameState      -0.09658
## PctSameHouse85         0.02664
## PctSameCity85          0.00249
## PctSameState85         0.03623
## LandArea              -0.02406
## PopDens               -0.14161
## PctUsePubTrans        -0.00695
## LemasPctOfficDrugUn    0.05007
```

```
lasso.test <- predict(lasso, newx = x.new,
              s = bst_lmd, type = 'response') - testing[,'nonViol.Rate']

lasso.test.MSE <- mean((lasso.test)**2)
lasso.test.MSE
```

```
## [1] 0.3386904
```

The $LASSO$ model has the effect of forcing some of the coefficient estimates to be exactly 0, yielding a sparse model, which includes only a subset of variables, similar to the stepwise selection methods. The solution path for the $LASSO$ model determines 73 predictors to be included in the model. The *test* MSE value is 0.339.

Summary of Results: *nonViol.Rate* (non-violent crime rate)

```
#### Calculating testing R^2

test.avg <- mean(testing[, 'nonViol.Rate'])
ols.pred <- predict(ols.fit, newdata = testing)
ols.test.R2 <- 1-mean((testing[,'nonViol.Rate']-
          ols.pred)^2)/mean((testing[, 'nonViol.Rate']-test.avg)^2)


step.pred <- predict(step.fit, newdata = testing)
step.test.R2 <- 1-mean((testing[,'nonViol.Rate']-
```

```
        step.pred)^2)/mean((testing[, 'nonViol.Rate']-test.avg)^2)


lasso.pred <- predict(lasso, x.new, s=bst_lmd)
lasso.test.R2 <- 1-mean((testing[,'nonViol.Rate']-
        lasso.pred)^2)/mean((testing[, 'nonViol.Rate']-test.avg)^2)


nv.test.R2 <- rbind(c("OLS", "Stepwise", "LASSO"),
          round(c(ols.test.R2, step.test.R2, lasso.test.R2), digits=3))


nv.adj.R2 <- rbind(c("OLS", "Stepwise"), c(0.604, 0.612))

nv.test.MSE <- rbind(c("OLS", "Stepwise", "LASSO"),
        round(c(ols.test.MSE, step.test.MSE, lasso.test.MSE), digits=3))


nv.test.MSE
```

```
##      [,1]    [,2]       [,3]
## [1,] "OLS"   "Stepwise" "LASSO"
## [2,] "0.354" "0.351"    "0.339"
```

```
nv.adj.R2
```

```
##      [,1]    [,2]
## [1,] "OLS"   "Stepwise"
## [2,] "0.604" "0.612"
```

```
nv.test.R2
```

```
##      [,1]    [,2]       [,3]
## [1,] "OLS"   "Stepwise" "LASSO"
## [2,] "0.581" "0.585"    "0.6"
```

The *test MSE* improves for the *Stepwise* selection approach compared to the OLS, and it further improves for the $LASSO$ method. However, the R^2_{adj} of the OLS and *Stepwise* models suggest that linear models do not approximate very well the true function regardless of the type of a model. Perhaps, a more flexible model or addition of some other predictor variables could lead to better estimation and lower *test* MSE.

The R^2 is the proportion of variation in y, explained by the regression. It measures the goodness of fit of a model. The higher R^2 is, the greater is the explanatory power of the regression model. The R^2_{adj} takes into account the relative simplicity of a model. A decrease in R^2_{adj} from the addition of one or more predictors signals that the added variable(s) are of little importance in the regression equation.

In addition to computing the generalization error (or test error), it may be interesting to compute the R^2 on the testing data set to gain some idea about the predictive quality of the model. If the model generalizes well, the value of $R^2_{testing}$ should not be much different from $R^2_{training}$. And this is what we observe here.

The table below summarizes the results of the analysis.

Model	MSE	R^2_{adj}
OLS	0.354	0.604
Stepwise	0.351	0.612
LASSO	0.339	

The response variable: *Viol.Rate* (violent crime rate)

```
v.crimedata4.std <- crimedata4.std[ ,-100]
dim(v.crimedata4.std)
```

```
## [1] 1901   99
```

```
set.seed(7)
n  <- dim(v.crimedata4.std)[1]
ID <- sample(1:n, size = 600, replace = FALSE)

training <- v.crimedata4.std[-ID,]
testing  <- v.crimedata4.std[ID,]
```

The step above performs a random split of the data into training and testing sets, leaving 600 observations (approximately 30% of the data) in the testing set.

Linear Model using Least Squares (OLS)

Fitting a linear model using the OLS on the training set and reporting the test error (MSE):

```
ols.fit <- lm(Viol.Rate ~ ., data = training)
summary(ols.fit)
```

```
##
## Call:
## lm(formula = Viol.Rate ~ ., data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3323 -0.3040 -0.0690  0.2241  3.5127
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.0220326  0.0168194   1.310 0.190462
## population           -0.4078768  0.1669878  -2.443 0.014727 *
## householdsize        -0.0611641  0.0833500  -0.734 0.463200
## racepctblack          0.2389826  0.0918772   2.601 0.009406 **
## racePctWhite          0.0461774  0.1034526   0.446 0.655416
## racePctAsian         -0.0204932  0.0503993  -0.407 0.684363
## racePctHisp          -0.0640732  0.0913580  -0.701 0.483226
## agePct12t21           0.1837325  0.1242052   1.479 0.139331
## agePct12t29          -0.2695779  0.1671248  -1.613 0.107000
## agePct16t24          -0.0346654  0.2376176  -0.146 0.884035
## agePct65up            0.0763035  0.1098513   0.695 0.487436
## pctUrban              0.0814826  0.0249136   3.271 0.001104 **
## medIncome            -0.2183022  0.1950155  -1.119 0.263189
## pctWWage             -0.1647129  0.0982020  -1.677 0.093746 .
## pctWFarmSelf          0.0237822  0.0232644   1.022 0.306865
## pctWInvInc           -0.0539627  0.0677916  -0.796 0.426184
## pctWSocSec           -0.0653950  0.1039265  -0.629 0.529310
## pctWPubAsst           0.1114353  0.0572507   1.946 0.051834 .
## pctWRetire           -0.0783193  0.0362749  -2.159 0.031044 *
```

```
## medFamInc              0.1562253  0.2020696   0.773 0.439600
## perCapInc             -0.0913065  0.2163759  -0.422 0.673114
## whitePerCap           -0.0057590  0.1762049  -0.033 0.973932
## blackPerCap           -0.0103994  0.0191185  -0.544 0.586579
## indianPerCap           0.0007298  0.0160782   0.045 0.963805
## AsianPerCap            0.0313305  0.0198968   1.575 0.115601
## OtherPerCap            0.0322010  0.0204963   1.571 0.116431
## HispPerCap             0.0250154  0.0255016   0.981 0.326823
## PctPopUnderPov        -0.1448724  0.0874666  -1.656 0.097919 .
## PctLess9thGrade       -0.2209314  0.0901020  -2.452 0.014347 *
## PctNotHSGrad           0.1609520  0.1142135   1.409 0.159029
## PctBSorMore            0.0170376  0.0917744   0.186 0.852753
## PctUnemployed          0.0085913  0.0460167   0.187 0.851927
## PctEmploy              0.1470709  0.0840192   1.750 0.080297 .
## PctEmplManu           -0.0612772  0.0367393  -1.668 0.095598 .
## PctEmplProfServ       -0.0040285  0.0418444  -0.096 0.923319
## PctOccupManu           0.0396823  0.0619471   0.641 0.521915
## PctOccupMgmtProf       0.1310977  0.0905456   1.448 0.147916
## MalePctDivorce         0.6931420  0.3757929   1.844 0.065359 .
## MalePctNevMarr         0.1562623  0.0827590   1.888 0.059245 .
## FemalePctDiv           0.4899045  0.4532094   1.081 0.279929
## TotalPctDiv           -1.0855631  0.8098306  -1.340 0.180342
## PersPerFam            -0.1226599  0.1905724  -0.644 0.519932
## PctFam2Par             0.2632700  0.1940518   1.357 0.175131
## PctKids2Par           -0.4182081  0.1864340  -2.243 0.025066 *
## PctYoungKids2Par      -0.0102672  0.0608818  -0.169 0.866107
## PctTeen2Par           -0.0225062  0.0471244  -0.478 0.633029
## PctWorkMomYoungKids    0.0572254  0.0455927   1.255 0.209671
## PctWorkMom            -0.1312697  0.0555230  -2.364 0.018225 *
## NumKidsBornNeverMar   -0.0269798  0.0864955  -0.312 0.755154
## PctKidsBornNeverMar    0.1371982  0.0622439   2.204 0.027699 *
## NumImmig               0.1656855  0.0775898   2.135 0.032930 *
## PctImmigRecent         0.0532334  0.0538502   0.989 0.323084
## PctImmigRec5          -0.0247965  0.0839397  -0.295 0.767733
## PctImmigRec8           0.0153139  0.0975068   0.157 0.875228
## PctImmigRec10          0.0102556  0.0740452   0.139 0.889865
## PctRecentImmig         0.1130824  0.1744791   0.648 0.517035
## PctRecImmig5          -0.1959067  0.3340452  -0.586 0.557671
## PctRecImmig8          -0.0767799  0.4175633  -0.184 0.854142
## PctRecImmig10          0.1514822  0.3181456   0.476 0.634060
## PctSpeakEnglOnly      -0.1075091  0.1064763  -1.010 0.312843
## PctNotSpeakEnglWell   -0.1003405  0.0948154  -1.058 0.290144
## PctLargHouseFam        0.1238792  0.2599424   0.477 0.633759
## PctLargHouseOccup     -0.2585774  0.2373239  -1.090 0.276128
## PersPerOccupHous       0.3731745  0.2807499   1.329 0.184032
## PersPerOwnOccHous      0.1361689  0.1757675   0.775 0.438663
## PersPerRentOccHous    -0.2337182  0.0950214  -2.460 0.014048 *
## PctPersOwnOccup       -1.1281643  0.4847681  -2.327 0.020119 *
## PctPersDenseHous       0.2568444  0.0960601   2.674 0.007602 **
## PctHousLess3BR         0.1050625  0.0622592   1.688 0.091766 .
## MedNumBR              -0.0131439  0.0282758  -0.465 0.642126
## HousVacant             0.2744423  0.0779342   3.521 0.000445 ***
## PctHousOccup          -0.0137014  0.0281630  -0.487 0.626699
## PctHousOwnOcc          0.9654995  0.4790707   2.015 0.044089 *
```

```
## PctVacantBoarded       0.0737520  0.0277940   2.654 0.008071 **
## PctVacMore6Mos        -0.0449817  0.0268773  -1.674 0.094470 .
## MedYrHousBuilt        -0.0123700  0.0376958  -0.328 0.742853
## PctHousNoPhone         0.0307635  0.0492912   0.624 0.532669
## PctWOFullPlumb        -0.0249426  0.0229010  -1.089 0.276307
## OwnOccLowQuart         0.1082102  0.1891529   0.572 0.567376
## OwnOccMedVal          -0.0074960  0.2781661  -0.027 0.978506
## OwnOccHiQuart         -0.1562373  0.1495022  -1.045 0.296210
## RentLowQ              -0.1750224  0.0855406  -2.046 0.040966 *
## RentMedian             0.0063551  0.1902811   0.033 0.973362
## RentHighQ             -0.1343072  0.1217676  -1.103 0.270257
## MedRent                0.3126000  0.1560658   2.003 0.045401 *
## MedRentPctHousInc     -0.0048618  0.0318711  -0.153 0.878782
## MedOwnCostPctInc       0.0171690  0.0374128   0.459 0.646384
## MedOwnCostPctIncNoMtg -0.0771226  0.0271969  -2.836 0.004649 **
## NumInShelters          0.0140775  0.0720805   0.195 0.845189
## NumStreet              0.0589711  0.0786664   0.750 0.453621
## PctForeignBorn         0.0563330  0.1300937   0.433 0.665079
## PctBornSameState      -0.0334136  0.0488250  -0.684 0.493884
## PctSameHouse85         0.0030418  0.0607091   0.050 0.960048
## PctSameCity85          0.0275060  0.0451459   0.609 0.542460
## PctSameState85         0.0621744  0.0504786   1.232 0.218302
## LandArea              -0.0124886  0.0165900  -0.753 0.451730
## PopDens               -0.0968381  0.0327880  -2.953 0.003203 **
## PctUsePubTrans         0.0268281  0.0320887   0.836 0.403286
## LemasPctOfficDrugUn    0.0536777  0.0188168   2.853 0.004410 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6005 on 1202 degrees of freedom
## Multiple R-squared:  0.6849, Adjusted R-squared:  0.6592
## F-statistic: 26.65 on 98 and 1202 DF,  p-value: < 2.2e-16
```

```
ols.test <- testing[, 'Viol.Rate'] - predict(ols.fit, newdata = testing,
                                             type = 'response')
# test MSE
ols.test.MSE <- mean(ols.test**2)
ols.test.MSE
```

```
## [1] 0.3114086
```

The *OLS* model determines a total of 20 predictors to be statistically significant at 0.05 level of significance:

- *HousVacant* - the number of vacant households;
- *racepctblack* - the percentage of population that is african american;
- *pctUrban* - percentage of people living in areas classified as urban;
- *PctPersDenseHous* - percent of persons in dense housing;
- *PctVacantBoarded* - the percent of vacant housing that is boarded up;
- *MedOwnCostPctIncNoMtg* - the median owners cost as a percentage of household income - for owners without a mortgage;
- *PopDens* - population density in persons per square mile;
- *LemasPctOfficDrugUn* - the percent of officers assigned to drug units;
- *population* - population for community;
- *pctWRetire* - the percentage of households with retirement income;
- *PctLess9thGrade* - the percentage of people 25 and over with less than a 9th grade education;

- *PctKids2Par* - the percentage of kids in family housing with two parents;
- *PctWorkMom* - the percentage of moms of kids under 18 in labor force;

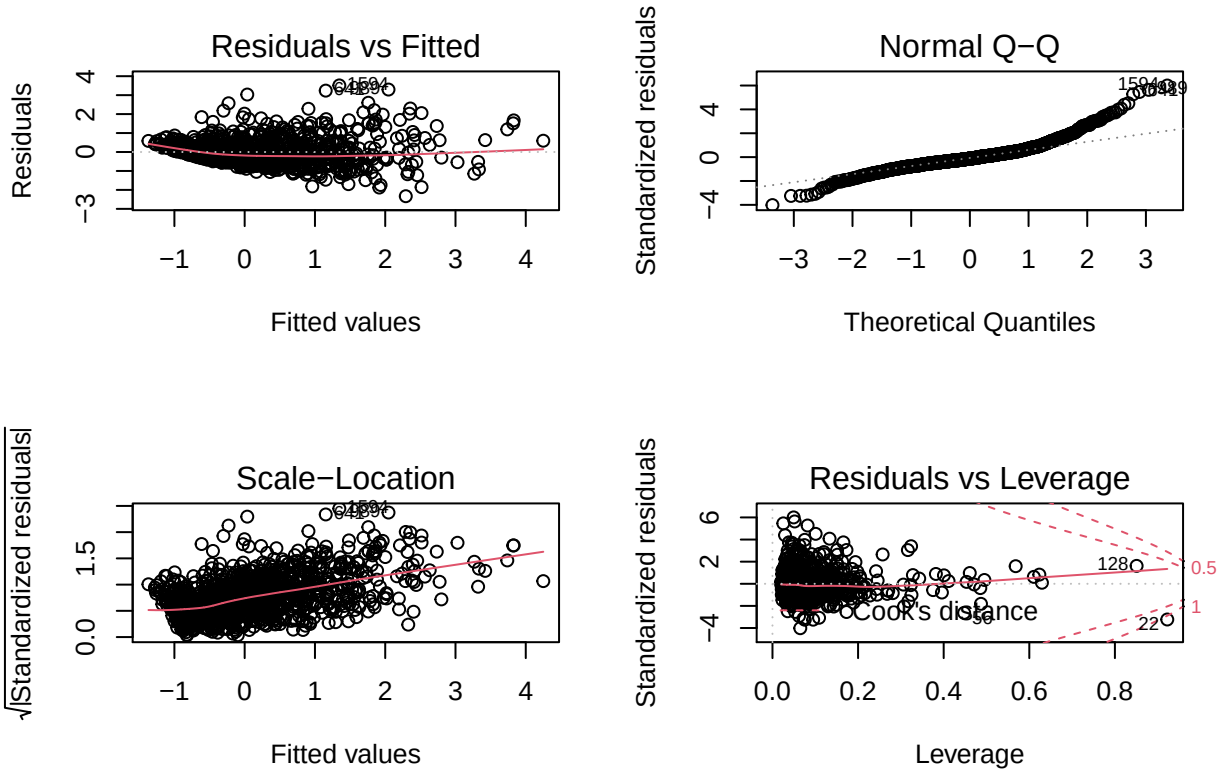
- *PctKidsBornNeverMar* - the percentage of kids born to never married;
- *NumImmig* - the total number of people known to be foreign born;

- *PersPerRentOccHous* - the mean persons per rental household;
- *PctPersOwnOccup* - the percent of people in owner occupied households;
- *PctHousOwnOcc* - the percent of households owner occupied;
- *RentLowQ* - rental housing - lower quartile rent;
- *MedRent* - the median gross rent.

For this model, the *test MSE* value is 0.311. The $R^2_{adj} = 0.659$ of the *OLS* suggests that the linear model works somewhat better when using violent crime rate as the response variable compared to the model with the non-violent crime rate as the response.

Assumptions Check

```
par(mfrow = c(2,2))
plot(ols.fit)
```



The residuals vs. fitted values plot does not show a completely random pattern suggesting some violation of the constant variance assumption. The Normal Q-Q plot reveals even more deviation from the normal distribution of error than in the non-violent crime rate modeling scenario.

Removing three leverage points improves the model performance:

```
r <- abs(ols.fit$residuals)
order((r), decreasing = TRUE)[1:3]
```

```
## [1] 925 576 381
```

```
training <- training[ -c(925, 576, 381),]

ols.fit <- lm(Viol.Rate ~ ., data = training)
summary(ols.fit)
```

```
##
## Call:
## lm(formula = Viol.Rate ~ ., data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.13299 -0.30746 -0.06362  0.21043  3.06317
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.0135845  0.0161436   0.841 0.400245
## population             -0.4019472  0.1601465  -2.510 0.012208 *
## householdsize          -0.0623362  0.0799285  -0.780 0.435604
## racepctblack            0.2457474  0.0880768   2.790 0.005352 **
## racePctWhite            0.0420961  0.0991924   0.424 0.671359
## racePctAsian           -0.0142587  0.0483180  -0.295 0.767967
## racePctHisp            -0.0460094  0.0876035  -0.525 0.599541
## agePct12t21             0.1669067  0.1190852   1.402 0.161301
## agePct12t29            -0.3140754  0.1602711  -1.960 0.050268 .
## agePct16t24             0.0191169  0.2278832   0.084 0.933159
## agePct65up              0.0705895  0.1053180   0.670 0.502826
## pctUrban                0.0690893  0.0239279   2.887 0.003954 **
## medIncome              -0.1652142  0.1871442  -0.883 0.377512
## pctWWage               -0.1661810  0.0941499  -1.765 0.077807 .
## pctWFarmSelf            0.0253618  0.0223158   1.136 0.255975
## pctWInvInc             -0.0605793  0.0650123  -0.932 0.351621
## pctWSocSec             -0.0595624  0.0996955  -0.597 0.550324
## pctWPubAsst             0.1334669  0.0549284   2.430 0.015252 *
## pctWRetire             -0.0751008  0.0347768  -2.160 0.031008 *
## medFamInc               0.1236329  0.1938428   0.638 0.523726
## perCapInc               0.0410943  0.2078068   0.198 0.843272
## whitePerCap            -0.1322434  0.1693596  -0.781 0.435048
## blackPerCap            -0.0109691  0.0183289  -0.598 0.549647
## indianPerCap           -0.0003289  0.0154154  -0.021 0.982982
## AsianPerCap             0.0267858  0.0191023   1.402 0.161104
## OtherPerCap             0.0155485  0.0197285   0.788 0.430781
## HispPerCap              0.0248053  0.0244547   1.014 0.310627
## PctPopUnderPov         -0.1589716  0.0839015  -1.895 0.058367 .
## PctLess9thGrade        -0.2291260  0.0863825  -2.652 0.008096 **
## PctNotHSGrad            0.1802165  0.1095091   1.646 0.100093
## PctBSorMore             0.0253788  0.0879987   0.288 0.773091
## PctUnemployed          -0.0142490  0.0442122  -0.322 0.747292
```

```
## PctEmploy              0.1615736  0.0805587   2.006 0.045117 *
## PctEmplManu           -0.0554295  0.0352262  -1.574 0.115860
## PctEmplProfServ       -0.0103953  0.0401567  -0.259 0.795782
## PctOccupManu           0.0296122  0.0594212   0.498 0.618333
## PctOccupMgmtProf       0.1100273  0.0868625   1.267 0.205514
## MalePctDivorce         0.6304776  0.3605877   1.748 0.080638 .
## MalePctNevMarr         0.1791202  0.0793663   2.257 0.024195 *
## FemalePctDiv           0.4157741  0.4346933   0.956 0.339024
## TotalPctDiv           -0.9555669  0.7768630  -1.230 0.218926
## PersPerFam            -0.1502762  0.1827024  -0.823 0.410945
## PctFam2Par             0.2025417  0.1863286   1.087 0.277249
## PctKids2Par           -0.3104043  0.1790914  -1.733 0.083314 .
## PctYoungKids2Par      -0.0147363  0.0584064  -0.252 0.800848
## PctTeen2Par           -0.0294946  0.0451883  -0.653 0.514072
## PctWorkMomYoungKids    0.0687797  0.0437262   1.573 0.115991
## PctWorkMom            -0.1494756  0.0532677  -2.806 0.005095 **
## NumKidsBornNeverMar   -0.0248947  0.0829209  -0.300 0.764059
## PctKidsBornNeverMar    0.1554197  0.0597080   2.603 0.009355 **
## NumImmig               0.1633177  0.0744074   2.195 0.028361 *
## PctImmigRecent         0.0482033  0.0516275   0.934 0.350659
## PctImmigRec5          -0.0107575  0.0805337  -0.134 0.893759
## PctImmigRec8          -0.0080666  0.0936834  -0.086 0.931397
## PctImmigRec10          0.0252666  0.0710573   0.356 0.722217
## PctRecentImmig         0.1211260  0.1672656   0.724 0.469113
## PctRecImmig5          -0.2128506  0.3202710  -0.665 0.506437
## PctRecImmig8          -0.0392020  0.4004869  -0.098 0.922039
## PctRecImmig10          0.1196943  0.3051508   0.392 0.694946
## PctSpeakEnglOnly      -0.1101501  0.1020703  -1.079 0.280734
## PctNotSpeakEnglWell   -0.1161320  0.0909156  -1.277 0.201722
## PctLargHouseFam        0.1668307  0.2492209   0.669 0.503364
## PctLargHouseOccup     -0.2874366  0.2275294  -1.263 0.206729
## PersPerOccupHous       0.3748114  0.2691410   1.393 0.163993
## PersPerOwnOccHous      0.1454453  0.1684970   0.863 0.388204
## PersPerRentOccHous    -0.2182527  0.0911054  -2.396 0.016746 *
## PctPersOwnOccup       -1.1199127  0.4647311  -2.410 0.016111 *
## PctPersDenseHous       0.2521527  0.0920933   2.738 0.006273 **
## PctHousLess3BR         0.1034988  0.0596905   1.734 0.083189 .
## MedNumBR              -0.0283463  0.0271460  -1.044 0.296595
## HousVacant             0.2773429  0.0747086   3.712 0.000215 ***
## PctHousOccup          -0.0131885  0.0269998  -0.488 0.625309
## PctHousOwnOcc          0.9541808  0.4592866   2.078 0.037965 *
## PctVacantBoarded       0.0547164  0.0268022   2.041 0.041421 *
## PctVacMore6Mos        -0.0407228  0.0257681  -1.580 0.114289
## MedYrHousBuilt        -0.0130413  0.0361510  -0.361 0.718354
## PctHousNoPhone         0.0408725  0.0472793   0.864 0.387492
## PctWOFullPlumb        -0.0203707  0.0219668  -0.927 0.353936
## OwnOccLowQuart         0.1179882  0.1813245   0.651 0.515364
## OwnOccMedVal          -0.0443991  0.2666736  -0.166 0.867798
## OwnOccHiQuart         -0.1308098  0.1433342  -0.913 0.361625
## RentLowQ              -0.1768207  0.0820050  -2.156 0.031265 *
## RentMedian             0.0217892  0.1824106   0.119 0.904938
## RentHighQ             -0.1540525  0.1167764  -1.319 0.187351
## MedRent                0.3041376  0.1496481   2.032 0.042338 *
## MedRentPctHousInc     -0.0021630  0.0305983  -0.071 0.943656
```

```
## MedOwnCostPctInc       0.0119970  0.0358889   0.334 0.738226
## MedOwnCostPctIncNoMtg -0.0774425  0.0260719  -2.970 0.003034 **
## NumInShelters          0.0269944  0.0691124   0.391 0.696172
## NumStreet              0.0408524  0.0754368   0.542 0.588233
## PctForeignBorn         0.0490555  0.1247252   0.393 0.694162
## PctBornSameState      -0.0577412  0.0468639  -1.232 0.218151
## PctSameHouse85         0.0109120  0.0582099   0.187 0.851332
## PctSameCity85          0.0207991  0.0432843   0.481 0.630944
## PctSameState85         0.0762000  0.0484190   1.574 0.115806
## LandArea              -0.0140893  0.0159040  -0.886 0.375850
## PopDens               -0.0975513  0.0314323  -3.104 0.001957 **
## PctUsePubTrans         0.0295340  0.0307652   0.960 0.337258
## LemasPctOfficDrugUn    0.0579243  0.0180488   3.209 0.001366 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5756 on 1199 degrees of freedom
## Multiple R-squared:  0.6954, Adjusted R-squared:  0.6705
## F-statistic: 27.94 on 98 and 1199 DF,  p-value: < 2.2e-16
```

```
ols.test <- testing[, 'Viol.Rate'] - predict(ols.fit, newdata = testing,
                                             type = 'response')
# test MSE
ols.test.MSE <- mean(ols.test**2)
ols.test.MSE
```

```
## [1] 0.3081019
```

The *test MSE* value has decreased from 0.311 to 0.308. The $R^2_{adj} = 0.671$.

Stepwise Selection

Applying the *stepwiseselection* to the above model with *AIC* and reporting the final model and the obtained test error:

```
step.fit <- step(ols.fit, direction = 'both', trace = 0, k = 2)
step.test <- testing[, 'Viol.Rate'] - predict(step.fit, newdata = testing,
                                              type = 'response')

# Final model and test error
summary(step.fit)
```

```
##
## Call:
## lm(formula = Viol.Rate ~ population + racepctblack + agePct12t21 +
##     agePct12t29 + pctUrban + pctWWage + pctWInvInc + pctWPubAsst +
##     pctWRetire + AsianPerCap + PctPopUnderPov + PctLess9thGrade +
##     PctNotHSGrad + PctEmploy + PctEmplManu + PctOccupMgmtProf +
##     MalePctDivorce + MalePctNevMarr + TotalPctDiv + PctKids2Par +
##     PctWorkMomYoungKids + PctWorkMom + PctKidsBornNeverMar +
##     NumImmig + PctImmigRecent + PctSpeakEnglOnly + PctNotSpeakEnglWell +
##     PctLargHouseOccup + PersPerOwnOccHous + PersPerRentOccHous +
##     PctPersOwnOccup + PctPersDenseHous + PctHousLess3BR + HousVacant +
##     PctHousOwnOcc + PctVacantBoarded + PctVacMore6Mos + OwnOccHiQuart +
```

```
##     RentLowQ + MedRent + MedOwnCostPctIncNoMtg + NumStreet +
##     PctBornSameState + PctSameState85 + PopDens + LemasPctOfficDrugUn,
##     data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.12889 -0.31040 -0.06767  0.20722  3.04239
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.01380    0.01586   0.870 0.384455
## population              -0.45294    0.12315  -3.678 0.000245 ***
## racepctblack             0.19526    0.03980   4.906 1.05e-06 ***
## agePct12t21              0.14867    0.05736   2.592 0.009658 **
## agePct12t29             -0.31462    0.08552  -3.679 0.000244 ***
## pctUrban                 0.07250    0.02163   3.351 0.000829 ***
## pctWWage                -0.17940    0.06756  -2.656 0.008019 **
## pctWInvInc              -0.12479    0.05402  -2.310 0.021039 *
## pctWPubAsst              0.08161    0.04267   1.913 0.056009 .
## pctWRetire              -0.05989    0.02900  -2.065 0.039147 *
## AsianPerCap              0.02712    0.01822   1.488 0.136922
## PctPopUnderPov          -0.08593    0.05805  -1.480 0.139062
## PctLess9thGrade         -0.24424    0.07757  -3.149 0.001680 **
## PctNotHSGrad             0.23475    0.09466   2.480 0.013269 *
## PctEmploy                0.19000    0.05475   3.470 0.000538 ***
## PctEmplManu             -0.03817    0.02086  -1.830 0.067491 .
## PctOccupMgmtProf         0.08780    0.04617   1.902 0.057419 .
## MalePctDivorce           0.32545    0.08817   3.691 0.000233 ***
## MalePctNevMarr           0.16525    0.05610   2.945 0.003284 **
## TotalPctDiv             -0.30424    0.09956  -3.056 0.002292 **
## PctKids2Par             -0.15696    0.08282  -1.895 0.058312 .
## PctWorkMomYoungKids      0.06209    0.04144   1.498 0.134311
## PctWorkMom              -0.13257    0.04716  -2.811 0.005017 **
## PctKidsBornNeverMar      0.14282    0.05020   2.845 0.004514 **
## NumImmig                 0.16985    0.06489   2.617 0.008969 **
## PctImmigRecent           0.04677    0.02052   2.279 0.022849 *
## PctSpeakEnglOnly        -0.07828    0.05426  -1.443 0.149359
## PctNotSpeakEnglWell     -0.12505    0.06597  -1.895 0.058259 .
## PctLargHouseOccup       -0.12749    0.06051  -2.107 0.035305 *
## PersPerOwnOccHous        0.26935    0.08429   3.195 0.001431 **
## PersPerRentOccHous      -0.23667    0.07766  -3.047 0.002357 **
## PctPersOwnOccup         -1.46486    0.31476  -4.654 3.60e-06 ***
## PctPersDenseHous         0.25714    0.06686   3.846 0.000126 ***
## PctHousLess3BR           0.10992    0.04655   2.361 0.018370 *
## HousVacant               0.29190    0.06403   4.559 5.65e-06 ***
## PctHousOwnOcc            1.28680    0.29691   4.334 1.58e-05 ***
## PctVacantBoarded         0.05622    0.02501   2.248 0.024743 *
## PctVacMore6Mos          -0.03569    0.02239  -1.594 0.111169
## OwnOccHiQuart           -0.10448    0.04333  -2.411 0.016052 *
## RentLowQ                -0.16023    0.06199  -2.585 0.009861 **
## MedRent                  0.21277    0.07259   2.931 0.003437 **
## MedOwnCostPctIncNoMtg -0.08817    0.02197  -4.012 6.37e-05 ***
## NumStreet                0.07023    0.04968   1.414 0.157728
## PctBornSameState        -0.07368    0.04045  -1.821 0.068778 .
```

```
## PctSameState85         0.09220    0.03827   2.409 0.016126 *
## PopDens               -0.06815    0.02539  -2.684 0.007369 **
## LemasPctOfficDrugUn    0.06043    0.01726   3.502 0.000478 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.569 on 1251 degrees of freedom
## Multiple R-squared:  0.6895, Adjusted R-squared:  0.6781
## F-statistic:  60.4 on 46 and 1251 DF,  p-value: < 2.2e-16
```

```
step.test.MSE <- mean(step.test**2)
step.test.MSE
```

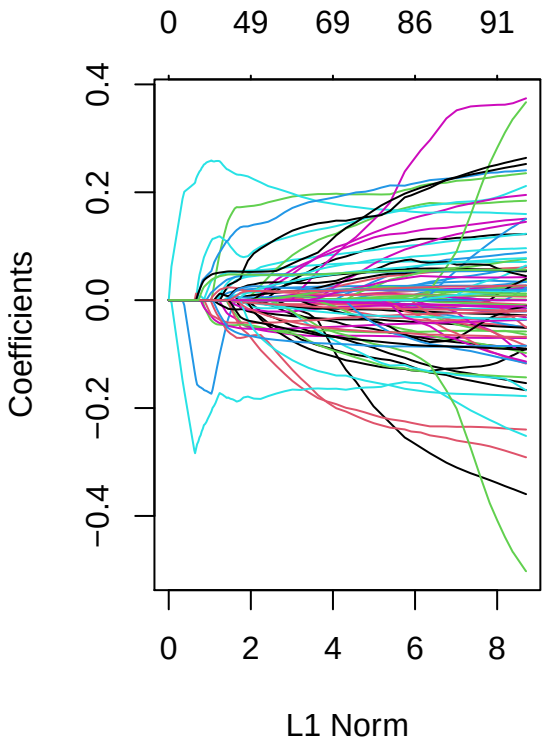
```
## [1] 0.303987
```

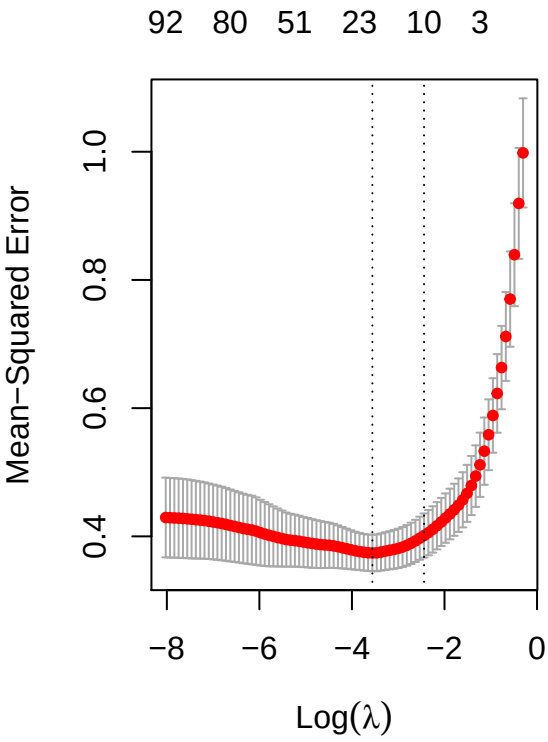
The *stepwiseselection* procedure includes 46 variables in the final model. There are several variables related to ethnicity, age, income, education (i.e., the percentage of people 25 and over with less than a 9th grade education/that are not high school graduates), employment, family and marital status (i.e., the percentage of moms of kids under 18 in labor force, the percentage of males who are divorced/who have never married), immigration (i.e., total number of people known to be foreign born, the percentage of immigrants who immigrated within last 3 years), housing and rent, population density in persons per square mile, community characteristics (i.e., percent of people living in the same house 5 years before, number of homeless people counted in the street), etc.

The *test MSE* value is 0.304. The $R^2_{adj} = 0.678$.

LASSO Regression Analysis

```
x.train <- model.matrix(Viol.Rate ~ 0+., data = training)
x.new   <- model.matrix(Viol.Rate ~ 0+., data = testing)

lasso <- glmnet(x.train, training[,'Viol.Rate'], alpha = 1,
                standardize = FALSE)
set.seed(7)
# cross-validation to tune the hyper-parameters
cv.lasso <- cv.glmnet(x.train, training[,'Viol.Rate'], alpha = 1,
                      standardize = FALSE)

par(mfrow = c(1,2))
plot(lasso, 'norm', label = T)
plot(cv.lasso)
```






```
bst_lmd <- cv.lasso$lambda.min

# observe coefficients
coef.lasso <- predict(lasso, x.new,
                      s=bst_lmd, type="coefficient", mode="fraction")
round(coef.lasso, 5)
```

```
## 99 x 1 sparse Matrix of class "dgCMatrix"
##                               1
## (Intercept)             0.00974
## population              .
## householdsize           .
## racepctblack            0.05900
## racePctWhite           -0.12509
## racePctAsian            .
## racePctHisp             .
## agePct12t21             .
## agePct12t29            -0.02605
## agePct16t24             .
## agePct65up              .
## pctUrban                0.04230
## medIncome               .
## pctWWage                .
## pctWFarmSelf            .
## pctWInvInc              .
## pctWSocSec              .
```

```
## pctWPubAsst            0.00289
## pctWRetire             .
## medFamInc              .
## perCapInc              .
## whitePerCap            .
## blackPerCap            .
## indianPerCap           .
## AsianPerCap            0.00023
## OtherPerCap            .
## HispPerCap             .
## PctPopUnderPov         .
## PctLess9thGrade        .
## PctNotHSGrad           .
## PctBSorMore            .
## PctUnemployed          .
## PctEmploy              .
## PctEmplManu           -0.00495
## PctEmplProfServ        .
## PctOccupManu           .
## PctOccupMgmtProf       .
## MalePctDivorce         0.11647
## MalePctNevMarr         .
## FemalePctDiv           .
## TotalPctDiv            .
## PersPerFam             .
## PctFam2Par             .
## PctKids2Par           -0.18379
## PctYoungKids2Par       .
## PctTeen2Par            .
## PctWorkMomYoungKids    .
## PctWorkMom            -0.04474
## NumKidsBornNeverMar    .
## PctKidsBornNeverMar    0.25801
## NumImmig               .
## PctImmigRecent         .
## PctImmigRec5           .
## PctImmigRec8           .
## PctImmigRec10          .
## PctRecentImmig         .
## PctRecImmig5           .
## PctRecImmig8           .
## PctRecImmig10          .
## PctSpeakEnglOnly       .
## PctNotSpeakEnglWell    .
## PctLargHouseFam        .
## PctLargHouseOccup      .
## PersPerOccupHous       .
## PersPerOwnOccHous      .
## PersPerRentOccHous     .
## PctPersOwnOccup        .
## PctPersDenseHous       0.08008
## PctHousLess3BR         0.00287
## MedNumBR              -0.03952
## HousVacant             0.05014
```

```
## PctHousOccup          -0.03341
## PctHousOwnOcc          .
## PctVacantBoarded       0.02333
## PctVacMore6Mos         .
## MedYrHousBuilt         .
## PctHousNoPhone         .
## PctWOFullPlumb         .
## OwnOccLowQuart         .
## OwnOccMedVal           .
## OwnOccHiQuart          .
## RentLowQ               .
## RentMedian             .
## RentHighQ              .
## MedRent                .
## MedRentPctHousInc      0.01512
## MedOwnCostPctInc       .
## MedOwnCostPctIncNoMtg -0.01688
## NumInShelters          .
## NumStreet              .
## PctForeignBorn         0.00696
## PctBornSameState       .
## PctSameHouse85         .
## PctSameCity85          .
## PctSameState85         .
## LandArea               .
## PopDens                .
## PctUsePubTrans         .
## LemasPctOfficDrugUn    0.04268
```

```
lasso.test <- predict(lasso, newx = x.new,
              s = bst_lmd, type = 'response') - testing[,'Viol.Rate']

lasso.test.MSE <- mean((lasso.test)**2)
lasso.test.MSE
```

```
## [1] 0.2994793
```

The solution path for the *LASSO* model determines 21 predictors to be included in the model. The *test MSE* value is 0.299.

Summary of Results: *Viol.Rate* (violent crime rate)

```
#### Calculating testing R^2

test.avg <- mean(testing[, 'Viol.Rate'])
ols.pred <- predict(ols.fit, newdata = testing)
ols.test.R2 <- 1-mean((testing[,'Viol.Rate']-
            ols.pred)^2)/mean((testing[, 'Viol.Rate']-test.avg)^2)


step.pred <- predict(step.fit, newdata = testing)
step.test.R2 <- 1-mean((testing[,'Viol.Rate']-
        step.pred)^2)/mean((testing[, 'Viol.Rate']-test.avg)^2)
```

```
lasso.pred <- predict(lasso, x.new, s=bst_lmd)
lasso.test.R2 <- 1-mean((testing[,'Viol.Rate']-
             lasso.pred)^2)/mean((testing[, 'Viol.Rate']-test.avg)^2)


v.test.R2 <- rbind(c("OLS", "Stepwise", "LASSO"),
          round(c(ols.test.R2, step.test.R2, lasso.test.R2), digits=3))


v.test.MSE <- rbind(c("OLS", "Stepwise", "LASSO"),
        round(c(ols.test.MSE, step.test.MSE, lasso.test.MSE), digits=3))


v.adj.R2 <- rbind(c("OLS", "Stepwise"), c(0.671, 0.678))

v.test.MSE
```

```
##      [,1]    [,2]       [,3]
## [1,] "OLS"   "Stepwise" "LASSO"
## [2,] "0.308" "0.304"    "0.299"
```

```
v.adj.R2
```

```
##      [,1]    [,2]
## [1,] "OLS"   "Stepwise"
## [2,] "0.671" "0.678"
```

```
v.test.R2
```

```
##      [,1]    [,2]       [,3]
## [1,] "OLS"   "Stepwise" "LASSO"
## [2,] "0.646" "0.651"    "0.656"
```

The *test* MSE marginally improves for the *Stepwise* selection approach compared to the OLS, and there is an additional small improvement for the $LASSO$ regression. The $LASSO$ model is the most parsimonious as it includes only 21 predictors and thus it seems to be the best model out of the ones presented in the current analysis.

Both the *test* MSE and R^2_{adj} suggest that linear models work a bit better for predicting the violent crime rate than for predicting non-violent crime rate. However, a more flexible model or addition of other explanatory variables could possibly lead to even better estimation and lower *test* MSE.

Model	MSE	R^2_{adj}
OLS	0.308	0.671
Stepwise	0.304	0.678
LASSO	0.299	

Classification Analysis

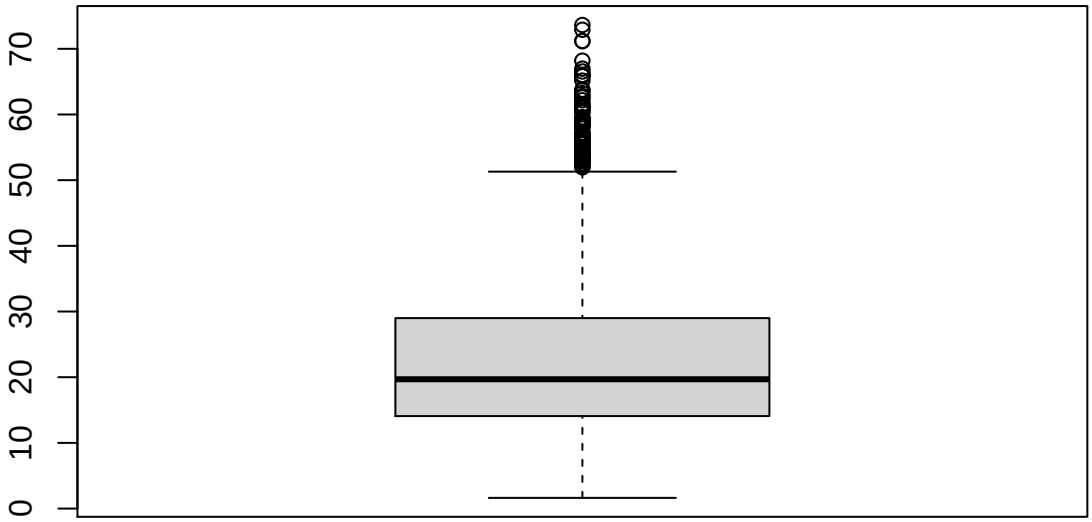
The second part of the analysis will focus on determining the factors that possibly lead to a larger % of people with bachelor's degree or higher education in a community.

Since the data set does not explicitly include a label for each community with a large % of people with bachelor's degree or higher, such a variable will be generated using the *PctBSorMore* (the percentage of people 25 and over with a bachelors degree or higher education). For the purpose of the analysis, communities with the % of educated people in the 66.7 percentile or above will be considered "more educated".

```
summary(crimedata4$PctBSorMore)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.63   14.08   19.69   23.03   29.00   73.63
```

```
# there are outliers: several communities have a large % of educated people
boxplot(crimedata4$PctBSorMore)
```




```
quantile(crimedata4$PctBSorMore,probs=2/3)
```

```
## 66.66667%
##  25.14667
```

```
Edu <- ifelse(crimedata4$PctBSorMore < 25, 0, 1)
# exclude all education related variables
class.crimedata <- data.frame(crimedata4[, -c(28:30)], Edu)
summary(class.crimedata$Edu)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3361  1.0000  1.0000
```

```
sum(class.crimedata$Edu)
```

```
## [1] 639
```

```
dim(class.crimedata)
```

```
## [1] 1901   98
```

We can once again have a look at the correlation coefficients for the *Edu* (which reflexes the associations for *PctBSorMore*) and the other variables:

```
cor(class.crimedata$Edu, class.crimedata)
##       population householdsize racepctblack racePctWhite racePctAsian
## [1,] -0.01565964  -0.007100512   -0.1686658    0.1835883     0.198261
##      racePctHisp agePct12t21 agePct12t29 agePct16t24 agePct65up  pctUrban
## [1,]  -0.1838172  0.07877017  0.06859988   0.1356151 -0.2128581 0.1952533
##      medIncome  pctWWage pctWFarmSelf pctWInvInc pctWSocSec pctWPubAsst
## [1,] 0.5597842 0.3981497   0.05936404  0.6051996 -0.3616263  -0.4863604
##      pctWRetire medFamInc perCapInc whitePerCap blackPerCap indianPerCap
## [1,] -0.1730851  0.617546 0.5997761   0.5886419   0.2875057    0.1291675
##      AsianPerCap OtherPerCap HispPerCap PctPopUnderPov PctUnemployed PctEmploy
## [1,]    0.257704   0.1998207  0.3978791     -0.3394138    -0.4615588 0.3836285
##      PctEmplManu PctEmplProfServ PctOccupManu PctOccupMgmtProf MalePctDivorce
## [1,]  -0.2658774       0.4364407    -0.650991        0.7789236     -0.4112975
##      MalePctNevMarr FemalePctDiv TotalPctDiv PersPerFam PctFam2Par PctKids2Par
## [1,]      0.1839551    -0.357914  -0.3905635 -0.1321576  0.3988373    0.424652
##      PctYoungKids2Par PctTeen2Par PctWorkMomYoungKids PctWorkMom
## [1,]        0.4302576   0.2979933         -0.05173591 0.01690667
##      NumKidsBornNeverMar PctKidsBornNeverMar    NumImmig PctImmigRecent
## [1,]         -0.04466836          -0.2643309 -0.01485215      0.1299224
##      PctImmigRec5 PctImmigRec8 PctImmigRec10 PctRecentImmig PctRecImmig5
## [1,]    0.1093638    0.1117204     0.0701585     0.09237883     0.068495
##      PctRecImmig8 PctRecImmig10 PctSpeakEnglOnly PctNotSpeakEnglWell
## [1,]    0.0661194    0.04406359       0.08708229            -0.14159
##      PctLargHouseFam PctLargHouseOccup PersPerOccupHous PersPerOwnOccHous
## [1,]      -0.2129761        -0.1930367        -0.065666        0.04294871
##      PersPerRentOccHous PctPersOwnOccup PctPersDenseHous PctHousLess3BR
## [1,]         -0.2795622       0.2230675       -0.2297589     -0.2635734
##       MedNumBR  HousVacant PctHousOccup PctHousOwnOcc PctVacantBoarded
## [1,] 0.1433457 -0.02945102    0.1521452     0.1657934       -0.2221285
##      PctVacMore6Mos MedYrHousBuilt PctHousNoPhone PctWOFullPlumb OwnOccLowQuart
## [1,]     -0.1851003      0.1317804     -0.4541527     -0.2618037      0.4875842
##      OwnOccMedVal OwnOccHiQuart  RentLowQ RentMedian RentHighQ   MedRent
## [1,]    0.5046947     0.5342215 0.4733693  0.5013017 0.5252031 0.4933443
##      MedRentPctHousInc MedOwnCostPctInc MedOwnCostPctIncNoMtg NumInShelters
## [1,]       -0.01592546        0.1837037           -0.06787694  -0.005360789
##        NumStreet PctForeignBorn PctBornSameState PctSameHouse85 PctSameCity85
## [1,] -0.01377752     0.06594092       -0.2522889    -0.08756951    -0.2930182
##      PctSameState85   LandArea     PopDens PctUsePubTrans LemasPctOfficDrugUn
## [1,]     -0.2499067 0.01540112 -0.02408809      0.1898818        -0.003816981
##       Viol.Rate nonViol.Rate Edu
```

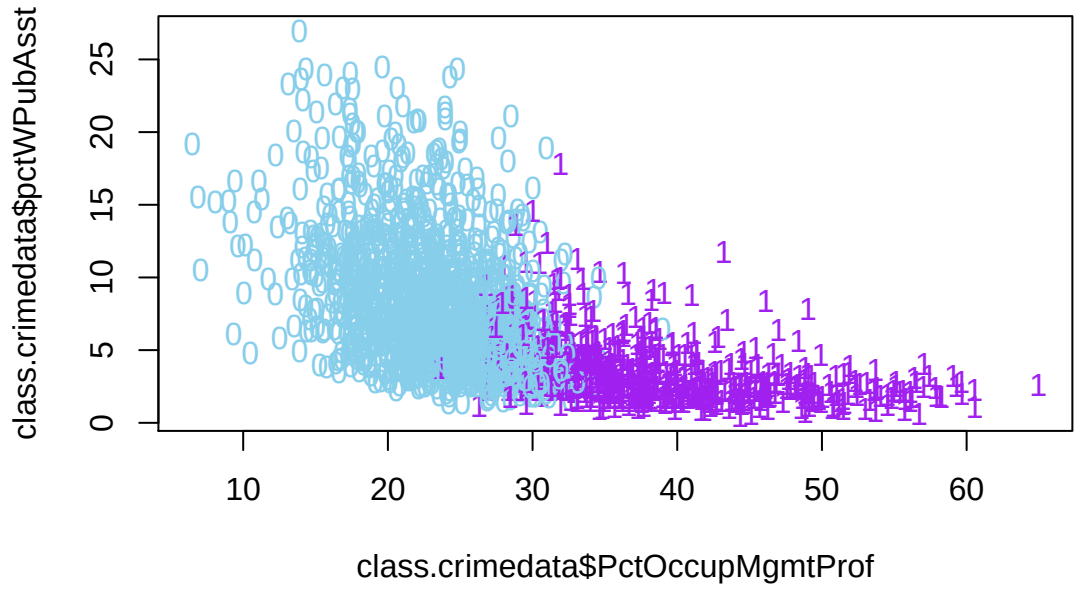
```
## [1,] -0.2650687    -0.259392   1
```

Some of the moderate to strong positive linear associations with *Edu* include *PctOccupMgmtProf* (the percentage of people 16 and over who are employed in management or professional occupations), *medFamInc* (the median family income - differs from household income for non-family households), *pctWInvInc* (the percentage of households with investment / rent income), *perCapInc* (per capita income), *whitePerCap* (per capita income for caucasians, *medIncome* (the median household income), and *OwnOccLowQuart*, *OwnOccMedVal*, *OwnOccHiQuart* (owner occupied housing - lower/median/upper quartile value).

Some of the moderate negative linear associations with *Edu* include *PctOccupManu* (the percentage of people 16 and over who are employed in manufacturing), *pctWPubAsst* (the percentage of households with public assistance income), and *PctHousNoPhone* (the percent of occupied housing units without phone)

We consider two inputs at a time and use them to assess the possibilities for the observations to belong to the "more educated" class or "less educated" class.

```
fact.class <- as.factor(class.crimedata$Edu)

#par(mfrow = c(1, 2))
plot(x = class.crimedata$PctOccupMgmtProf,
     y = class.crimedata$pctWPubAsst,
     pch = as.character(class.crimedata$Edu),
     col = c("skyblue", "purple")[fact.class])
```




```
plot(x = class.crimedata$pctWInvInc,
     y = class.crimedata$PctOccupManu,
     pch = as.character(class.crimedata$Edu),
     col = c("skyblue", "purple")[fact.class])
```

By checking graphically the data set using just two variables at a time, we notice a clear possibility for a linear separation between communities with a higher % of educated people and those with lower % of educated people.

Again, randomly splitting the data into training and testing sets, leaving 600 observations (approximately 30% of data) in the testing set:

```
set.seed(7)
n  <- dim(class.crimedata)[1]
ID <- sample(1:n, size = 600, replace = FALSE)

training <- class.crimedata[-ID,]
testing  <- class.crimedata[ID,]
```

Parametric (model-based) Methods

LDA (Linear Discriminant Analysis)

LDA is a probabilistic learning method, which produces the posterior probabilities for each observation to belong to one of the two classes. *LDA* assign the class label to the observation based on the largest posterior probability from the two groups. This method requires the equality of variance-covariance matrix assumption for the two groups which should be tested in the low-dimensional case. However, when the number of the predictors p is large, the *LDA* is always preferred over the *QDA* (Quadratic Discriminant Analysis) as *QDA* highly relies on the normality assumption which is difficult to verify when p is large. The *LDA* is robust to the violation of the multivariate normal assumption.

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.0.4
```

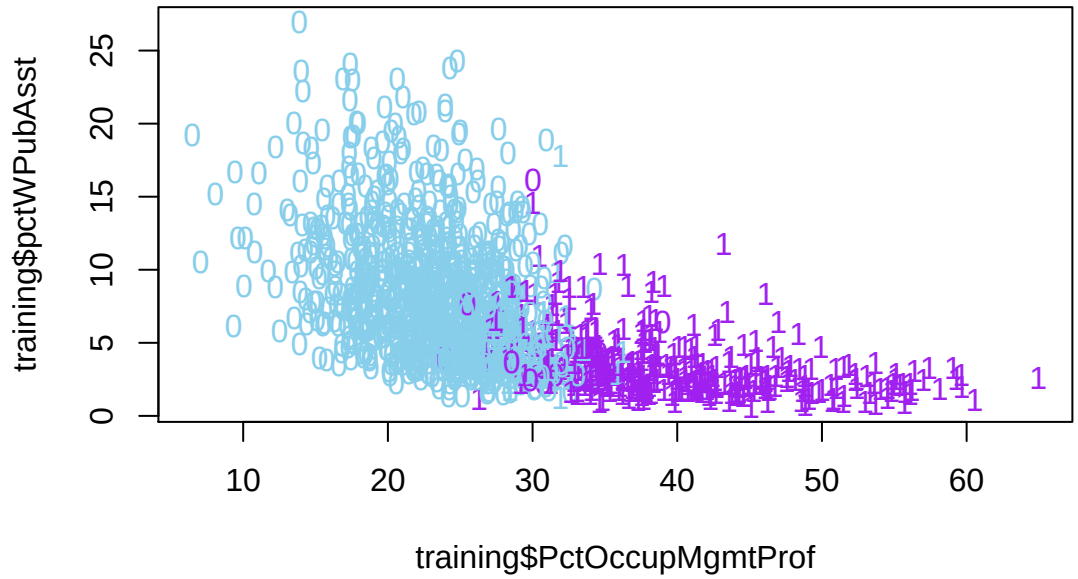
```
# By default the prior is the sample proportion
train.crimedata.lda <- lda(Edu ~ ., data = training)
train.crimedata.predict <- predict(train.crimedata.lda)$posterior

# The confusion matrix: the true class label vs. the predicted class label.
table(training$Edu, predict(train.crimedata.lda )$class)
```

```
##
##       0   1
##   0 862  11
##   1  46 382
```

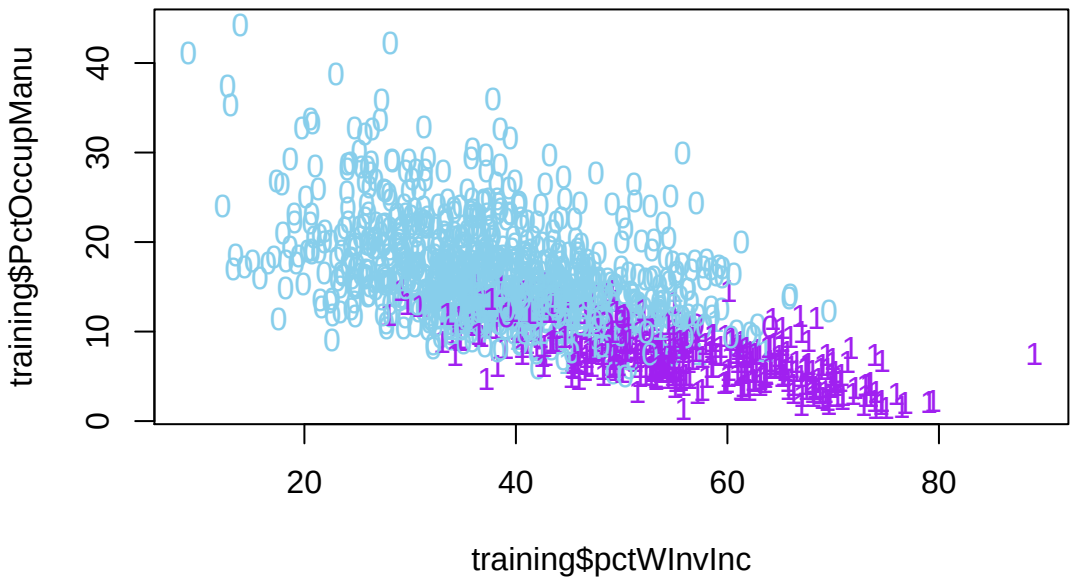
```
# The training error/the misclassification error rate
mean(training$Edu!=predict(train.crimedata.lda)$class)
```

```
## [1] 0.04381245
```

```
#par(mfrow = c(1, 2))
plot(x=training$PctOccupMgmtProf, y = training$pctWPubAsst,
     pch = as.character(training$Edu),
     col = c("skyblue", "purple")[predict(train.crimedata.lda)$class])
```




```
plot(x=training$pctWInvInc, y = training$PctOccupManu,
     pch = as.character(training$Edu),
     col = c("skyblue", "purple")[predict(train.crimedata.lda)$class])
```



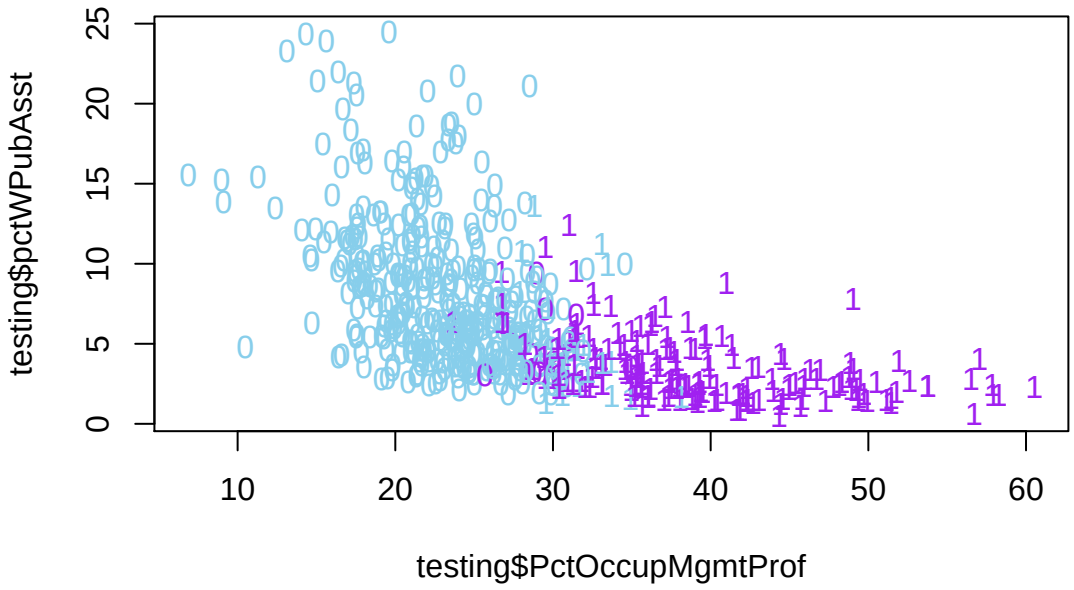

```
# Test Data
test.crimedata.predict <- predict(train.crimedata.lda, newdata = testing)

# The confusion matrix
table(testing$Edu, test.crimedata.predict$class)
```

```
##
##       0   1
##   0 379  10
##   1  27 184
```

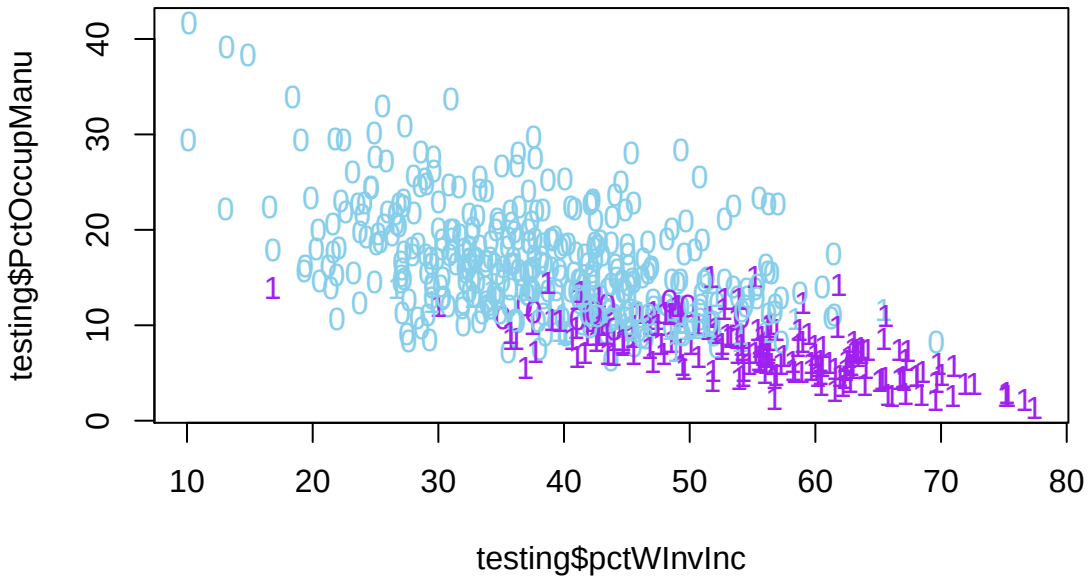
```
# The test error
mean(testing$Edu!=test.crimedata.predict$class)
```

```
## [1] 0.06166667
```

```
#par(mfrow = c(1, 2))
plot(x=testing$PctOccupMgmtProf, y = testing$pctWPubAsst,
     pch = as.character(testing$Edu),
     col = c("skyblue", "purple")[test.crimedata.predict$class])
```




```
plot(x=testing$pctWInvInc, y = testing$PctOccupManu,
     pch = as.character(testing$Edu),
     col = c("skyblue", "purple")[test.crimedata.predict$class])
```



The training error is approximately *0.0438*. The test error is about *0.0617*, which is higher than the training error. The misclassification error for the test set is usually sufficiently higher than the training error when a linear model is capable of finding a reasonable discriminant line to separate the two group.

The plots show that most of the misclassified observations are close to the border.

Overall, the LDA method seems to be a good fit for the data.

Logistic Regression

The *Logistic Regression* can be characterized by so-called structural defect, i.e., a binary response and continuous predictors. Like the LDA, this method assumes the $logit(\pi)$. However, while in the LDA the $logit(\pi)$ is derived by assuming the posterior probability function, i.e. that X follows the multivariate normal distribution, the *Logistic Regression* the $logit(\pi)$ is derived from the Bernoulli distribution. Both assume the $logit(\pi)$ is a linear function of the predictors. Also, while the LDA imposes the assumption on X (marginal distribution), which is difficult to verify in the high dimensional scenario, the *Logistic Regression* assumes the conditional distribution of the response $Y|X \sim Bernoulli$, which is in general a more reasonable assumption. Thus, while it is not important for the current project, the *Logistic Regression* is more flexible as it can intake both continuous and discrete predictors, while the LDA is applicable for continuous predictors only. The *Logistic Regression* is more practical when p is not large and it allows the interpretation of the coefficients.

Both methods are relatively simple and should be used first for a classification problem as they allow to combine the effect of all the factors when identifying the difference between groups.

The major limitation of the *Logistic Regression*, however, is the separation issue, i.e., the model set up requires some overlap between the groups and does not work for well-separated groups. The existence of the partial separation is very difficult to identify when the number of predictors is large and can be known from the error messages based on the *Logistic Regression* fit. This limitation makes the *Logistic Regression* impractical for applications with high dimensions and requires a remedy in a form of a reduced model.

```
fit.logit <- glm(Edu~., family=binomial(link=logit), data=training)
```

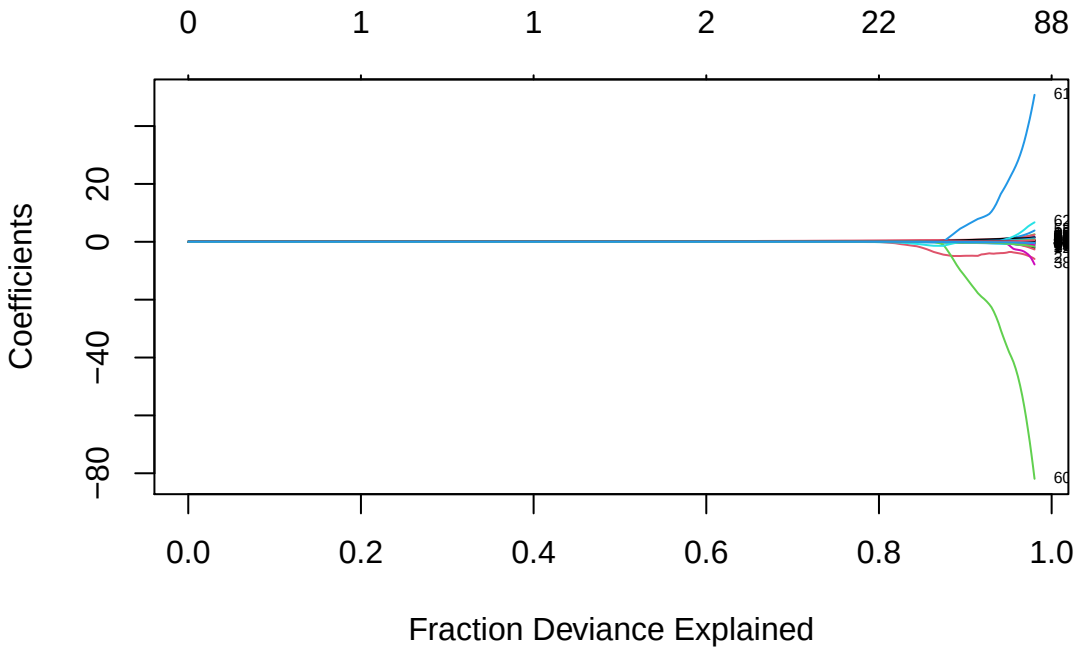
```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

As expected, the *Logistic Regression* does not converge due to the partial separation in the data.

Regularized Logistic Regression Using the L_1 penalty for variables selection and shrinkage to fit a reduced *Logistic Regression* resulted in the same error message:

```
## L1 Logistic regression
x.train <- model.matrix(Edu ~ 0+., data = training)
x.new   <- model.matrix(Edu ~ 0+., data = testing)

class.lasso <- glmnet(x.train, training[,'Edu'], family = "binomial")

# solution path
plot(class.lasso, xvar = "dev", label = TRUE)
```




```
pred.lasso <- predict(class.lasso,type="response",newx=x.new, s=c(0.01,0.05))


cvfit <- cv.glmnet(x.train, training[,'Edu'], family = "binomial", type.measure = "class")
#plot(cvfit)

#cvfit$lambda.min
#cvfit$lambda.1se
coef(cvfit, s = "lambda.min")
```

```
## 98 x 1 sparse Matrix of class "dgCMatrix"
##                                    1
## (Intercept)           -1.855403e+02
## population             7.728503e-06
## householdsize         -4.273831e+00
## racepctblack          -2.187998e-02
## racePctWhite           .
## racePctAsian           8.677565e-02
## racePctHisp            .
## agePct12t21            .
## agePct12t29            .
## agePct16t24            3.855599e-01
## agePct65up             3.762024e-01
## pctUrban               5.768713e-03
## medIncome             -1.099186e-04
## pctWWage              -9.037138e-03
## pctWFarmSelf          -1.365051e-01
## pctWInvInc             1.519717e-01
## pctWSocSec            -2.219257e-01
```

```
## pctWPubAsst           -4.601435e-01
## pctWRetire            -1.284938e-01
## medFamInc              1.105207e-04
## perCapInc             -2.792814e-04
## whitePerCap            .
## blackPerCap           -1.336860e-04
## indianPerCap          -4.937289e-06
## AsianPerCap           -4.888624e-05
## OtherPerCap           -2.395886e-05
## HispPerCap            -3.043349e-05
## PctPopUnderPov        -6.494134e-03
## PctUnemployed          1.148595e-01
## PctEmploy              4.247108e-01
## PctEmplManu           -3.242227e-02
## PctEmplProfServ        2.099314e-01
## PctOccupManu           1.698082e-02
## PctOccupMgmtProf       8.103898e-01
## MalePctDivorce        -3.125307e-01
## MalePctNevMarr        -8.266502e-02
## FemalePctDiv           .
## TotalPctDiv            .
## PersPerFam             .
## PctFam2Par             .
## PctKids2Par            .
## PctYoungKids2Par      -7.006674e-02
## PctTeen2Par           -9.824353e-02
## PctWorkMomYoungKids   -8.728669e-02
## PctWorkMom             .
## NumKidsBornNeverMar   -2.803780e-05
## PctKidsBornNeverMar    6.082886e-02
## NumImmig              -1.242811e-05
## PctImmigRecent         .
## PctImmigRec5          -2.473745e-02
## PctImmigRec8           6.111796e-03
## PctImmigRec10          .
## PctRecentImmig         .
## PctRecImmig5           .
## PctRecImmig8           .
## PctRecImmig10          4.791557e-01
## PctSpeakEnglOnly       5.728057e-02
## PctNotSpeakEnglWell    1.388996e-01
## PctLargHouseFam        7.763609e-01
## PctLargHouseOccup      .
## PersPerOccupHous      -1.976641e+01
## PersPerOwnOccHous      8.699954e+00
## PersPerRentOccHous     .
## PctPersOwnOccup        1.018968e-01
## PctPersDenseHous       .
## PctHousLess3BR        -1.583085e-01
## MedNumBR               .
## HousVacant            -8.534166e-05
## PctHousOccup          -1.034397e-01
## PctHousOwnOcc          .
## PctVacantBoarded       1.088712e-01
```

```
## PctVacMore6Mos          4.124119e-02
## MedYrHousBuilt          9.508205e-02
## PctHousNoPhone          3.377789e-01
## PctWOFullPlumb         -3.847501e-01
## OwnOccLowQuart          .
## OwnOccMedVal            .
## OwnOccHiQuart           3.355566e-05
## RentLowQ               -7.685124e-03
## RentMedian              .
## RentHighQ               4.995373e-03
## MedRent                 .
## MedRentPctHousInc       1.008328e-01
## MedOwnCostPctInc       -6.677055e-02
## MedOwnCostPctIncNoMtg   1.082095e-01
## NumInShelters           6.186705e-04
## NumStreet              -1.853199e-03
## PctForeignBorn         -1.804144e-01
## PctBornSameState        .
## PctSameHouse85         -6.309133e-02
## PctSameCity85           .
## PctSameState85         -2.192275e-02
## LandArea                .
## PopDens                -1.204326e-04
## PctUsePubTrans          1.037296e-01
## LemasPctOfficDrugUn     1.326662e-01
## Viol.Rate              -2.551986e-01
## nonViol.Rate           -1.017084e-01
```

```
fit.logit <- glm(Edu~population+householdsize+racepctblack+racePctAsian+
                   agePct16t24+agePct65up+pctUrban+medIncome+pctWWage+
                   pctWFarmSelf+pctWInvInc+pctWSocSec+
                   pctWPubAsst+pctWRetire+medFamInc+perCapInc+
                   blackPerCap+indianPerCap+AsianPerCap+
                   OtherPerCap+HispPerCap+PctPopUnderPov+PctUnemployed+
                   PctEmploy+PctEmploy+PctEmplManu+PctEmplProfServ+PctOccupManu+
                   PctOccupMgmtProf+MalePctDivorce+MalePctNevMarr+
                   PctYoungKids2Par+PctTeen2Par+
                   PctWorkMomYoungKids+NumKidsBornNeverMar+PctKidsBornNeverMar+
                   NumImmig+PctImmigRec5+PctImmigRec8+PctRecImmig10+
                   PctSpeakEnglOnly+PctNotSpeakEnglWell+PctLargHouseFam+
                   PersPerOccupHous+PersPerOwnOccHous+PctPersOwnOccup+
                   PctPersOwnOccup+PctHousLess3BR+HousVacant+
                   PctHousOccup+PctVacantBoarded+PctVacMore6Mos+MedYrHousBuilt+
                   PctHousNoPhone+PctWOFullPlumb+OwnOccHiQuart+RentLowQ+
                   RentHighQ+MedRentPctHousInc+MedOwnCostPctInc+
                   MedOwnCostPctIncNoMtg+NumInShelters+
                   NumStreet+PctForeignBorn+PctSameHouse85+PctSameState85+
                   PopDens+PctUsePubTrans+LemasPctOfficDrugUn+
                   Viol.Rate+nonViol.Rate,
                   family=binomial(link=logit), data=training)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Classification Tree (CART)

While the *LDA* and the *Logistic Regression* are model-based, interpretable methods that allow the estimation of classification probability, they are not as flexible as, for instance, a local classifier kNN. However, non model-based classifiers cannot estimate probabilities and are not interpretable (i.e., they are useful for prediction but cannot be used to study the relative importance among the inputs).

Some approaches in between the two mentioned above are the tree-based methods. Such methods are non-model based, however they are interpretable and can provide the estimation of the probabilities. They can also handle both continuous and categorical features in a simple and natural way, enjoy automatic stepwise selection and impurity reduction; they are invariant under monotonic transformation and, which is important for the current project, robust to outliers due to the feature truncation which reduces the effect of the extreme values. Some of the limitations include high variance due to the hierarchical nature of the splitting process; results may differ if there are even small changes in data, i.e. tree models suffer from instability. It is important to avoid overfitting by "pruning a tree" and apply the cost-complexity measure to achieve an efficient algorithm when using such methods.

Considering the importance of the pruning and controlling the complexity of the parameters I build the *Classification Tree* model using two different packages.

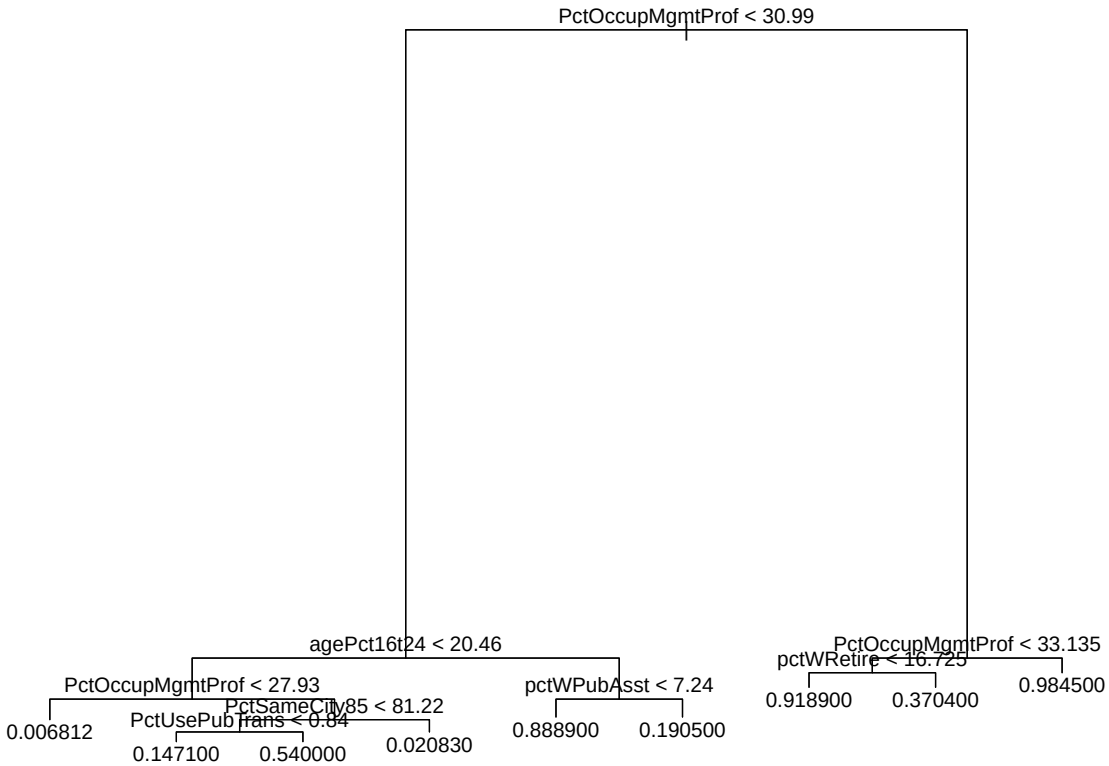
```
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.0.4
```

```
crimedata.tree <- tree(Edu ~ ., data = training)
crimedata.tree
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 1301 287.2000 0.329000
##    2) PctOccupMgmtProf < 30.99 914  61.2300 0.072210
##      4) agePct16t24 < 20.46 866  36.3300 0.043880
##        8) PctOccupMgmtProf < 27.93 734   4.9660 0.006812 *
##        9) PctOccupMgmtProf > 27.93 132  24.7500 0.250000
##         18) PctSameCity85 < 81.22 84  19.8100 0.381000
##           36) PctUsePubTrans < 0.84 34   4.2650 0.147100 *
##           37) PctUsePubTrans > 0.84 50  12.4200 0.540000 *
##         19) PctSameCity85 > 81.22 48   0.9792 0.020830 *
##      5) agePct16t24 > 20.46 48  11.6700 0.583300
##       10) pctWPubAsst < 7.24 27   2.6670 0.888900 *
##       11) pctWPubAsst > 7.24 21   3.2380 0.190500 *
##    3) PctOccupMgmtProf > 30.99 387  23.3900 0.935400
##      6) PctOccupMgmtProf < 33.135 64  13.7500 0.687500
##       12) pctWRetire < 16.725 37   2.7570 0.918900 *
##       13) pctWRetire > 16.725 27   6.2960 0.370400 *
##      7) PctOccupMgmtProf > 33.135 323   4.9230 0.984500 *
```

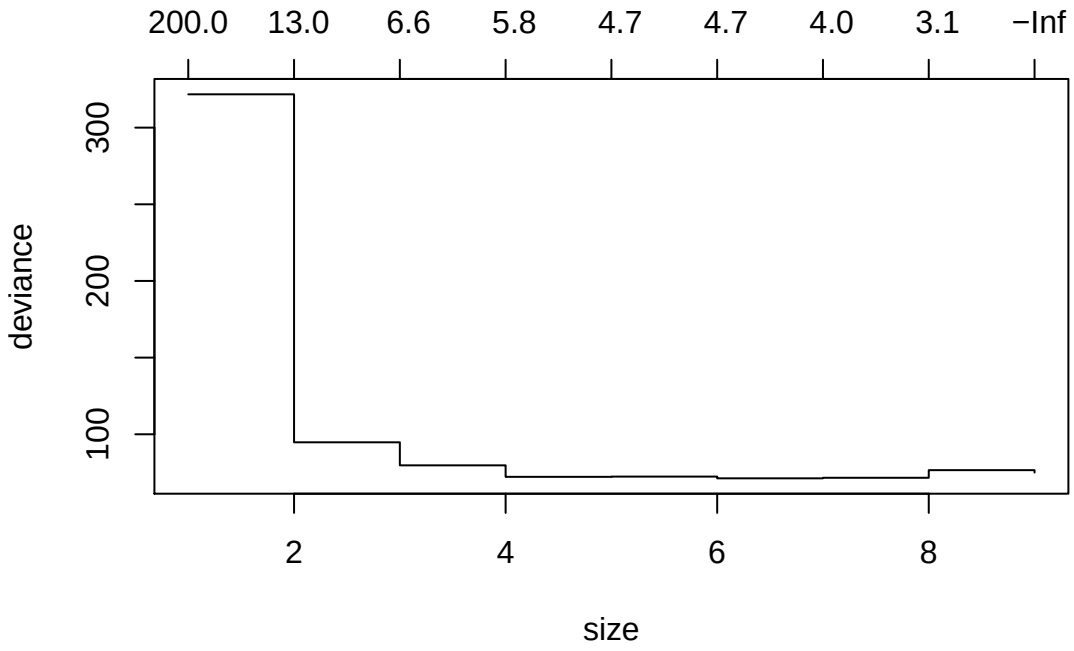
```
plot(crimedata.tree)
text(crimedata.tree)
```




```
crimedata.tree.cv <- cv.tree(crimedata.tree, K = nrow(training))
crimedata.tree.cv
```

```
## $size
## [1] 9 8 7 6 5 4 3 2 1
##
## $dev
## [1]  75.00422  76.60982  71.62204  71.26140  72.39119  72.22983  79.73081
## [8]  94.75861 321.72785
##
## $k
## [1]       -Inf   3.124818   3.961310   4.696947   4.712412   5.761905   6.616623
## [8]  13.234905 202.578392
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(crimedata.tree.cv)
```




```
crimedata.tree <- tree(as.factor(Edu) ~ ., data = training)
crimedata.prune.tree<-prune.tree(crimedata.tree, best = 6)


confusion <- function(a, b){
  tbl <- table(a, b)
  mis <- 1 - sum(diag(tbl))/sum(tbl)
  list(table = tbl, misclass.prob = mis)
}

confusion(predict(crimedata.prune.tree, type="class"), training$Edu)
```

```
## $table
##    b
## a     0   1
##   0 838  38
##   1  35 390
##
## $misclass.prob
## [1] 0.05611068
```

```
confusion(predict(crimedata.prune.tree, testing, type="class"), testing$Edu)
```

```
## $table
##    b
## a     0   1
##   0 374  23
##   1  15 188
```

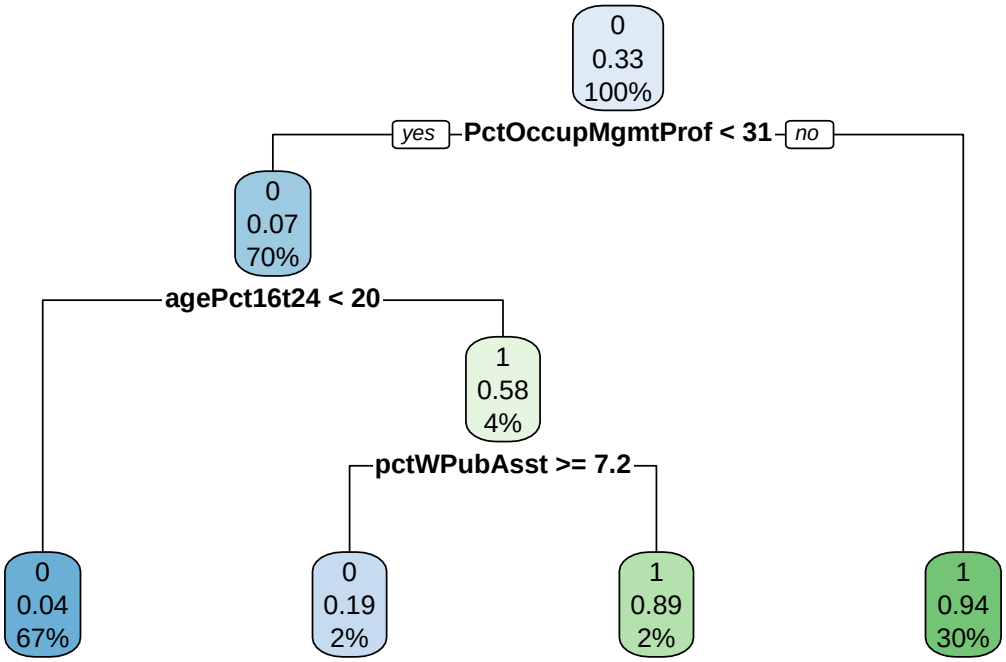
```
##
## $misclass.prob
## [1] 0.06333333
```

Using an alternative package:

```
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.0.5
```

```
crimedata.rp <- rpart(Edu ~ ., data = training, method="class")
rpart.plot(crimedata.rp)
```




```
confusion(predict(crimedata.rp, type="class"), training$Edu)
```

```
## $table
##    b
## a     0   1
##   0 845  42
##   1  28 386
##
## $misclass.prob
## [1] 0.05380477
```

```
confusion(predict(crimedata.rp, testing, type="class"), testing$Edu)
```

```
## $table
##    b
## a     0   1
```

```
##   0 375  26
##   1  14 185
##
## $misclass.prob
## [1] 0.06666667
```

Apparently, only a few features, mainly *PctOccupMgmtProf* (the percentage of people 16 and over who are employed in management or professional occupations), *agePct16t24* (the percentage of the population that is 16-24 in age), and *pctWPubAsst* (the percentage of households with public assistance income), are sufficient to build a decision tree model that fits the data. *PctOccupMgmtProf* is reused in the model. Some other features used by the model include *pctWRetire* (the percentage of households with retirement income), *PctSameCity85* (the percent of people living in the same city for 5 years as in 1985), and *PctUsePubTrans* (the percent of people using public transit for commuting).

The *tree* package model provides a smaller misclassification error on the testing data set. Thus, for the *Classification Tree* method I report the training error of *0.0561* and the test error of *0.0633*. The test error is slightly higher than for the *LDA* method.

Ensemble Classifiers

The main goal of all the ensemble classifiers is to improve the performance of individual "weak learners" that suffer from instability by constructing many classifiers from the same data and combining or averaging the outputs together. This allows to reduce the variability and to improve the prediction accuracy.

Bagging

Each node of a *Classification Tree* provides only a local optimal decision due to the instability of the method. In most cases a single tree model is not sufficiently accurate. Ensemble classifiers allow to construct multiple tree models from the same training sample, using bootstrap resampling.

While the *Bagging* ensemble classifier can reduce the variance and improve the prediction accuracy by aggregating predictions from multiple individual tree models built from the bootstrap samples and classifying observations by consensus voting or by averaging probabilities, it has some important limitation such as loss of interpretability and the fact that it requires the individual classifiers to be independent. As a result, it cannot handle well highly correlated predictor variables, leading to increased bias, and thus may not be the best choice for the *Communities and Crime* data. Nevertheless, I attempt using the *Bagging* method. The most important turning parameter for this method is the number of the bootstrap samples. Most of the time 25 or 50 bootstrap replicates provide the most reasonable misclassification rate. Again, I apply two alternative packages *ipred* and *rpart*.

```
library(ipred)
```

```
## Warning: package 'ipred' was built under R version 4.0.5
```

```
# The number of bootstrap samples of 25 provides the best test error
crimedata.bag1 <- bagging(as.factor(Edu) ~ ., data = training, coob = T)
crimedata.bag1
```

```
##
## Bagging classification trees with 25 bootstrap replications
##
## Call: bagging.data.frame(formula = as.factor(Edu) ~ ., data = training,
##     coob = T)
##
## Out-of-bag estimate of misclassification error:  0.0638
```

```
acc=(testing[,98]==predict(crimedata.bag1, testing))
1-length(acc[acc=="TRUE"])/length(acc)
```

```
## [1] 0.04833333
```

```
crimedata.bag1$mtrees[[1]]$btree
```

```
## n= 1301
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##   1) root 1301 461 0 (0.645657187 0.354342813)
##     2) PctOccupMgmtProf< 30.32 857  62 0 (0.927654609 0.072345391)
##       4) agePct12t21< 19.605 813  35 0 (0.956949569 0.043050431)
##         8) PctOccupMgmtProf< 28.075 697   3 0 (0.995695839 0.004304161)
##          16) PctHousOccup>=57.2 694   0 0 (1.000000000 0.000000000) *
##          17) PctHousOccup< 57.2 3   0 1 (0.000000000 1.000000000) *
##         9) PctOccupMgmtProf>=28.075 116  32 0 (0.724137931 0.275862069)
##          18) PctSameHouse85>=46.31 75   9 0 (0.880000000 0.120000000)
##            36) pctWRetire>=15.32 57   1 0 (0.982456140 0.017543860)
##              72) agePct12t21>=10.83 56   0 0 (1.000000000 0.000000000) *
##              73) agePct12t21< 10.83 1   0 1 (0.000000000 1.000000000) *
##            37) pctWRetire< 15.32 18   8 0 (0.555555556 0.444444444)
##              74) agePct16t24>=12.025 10   1 0 (0.900000000 0.100000000)
##               148) agePct16t24< 16.435 9   0 0 (1.000000000 0.000000000) *
##               149) agePct16t24>=16.435 1   0 1 (0.000000000 1.000000000) *
##              75) agePct16t24< 12.025 8   1 1 (0.125000000 0.875000000)
##               150) householdsize>=2.96 1   0 0 (1.000000000 0.000000000) *
##               151) householdsize< 2.96 7   0 1 (0.000000000 1.000000000) *
##          19) PctSameHouse85< 46.31 41  18 1 (0.439024390 0.560975610)
##            38) PctBornSameState< 43.265 12   1 0 (0.916666667 0.083333333)
##              76) HispPerCap< 11441.5 11   0 0 (1.000000000 0.000000000) *
##              77) HispPerCap>=11441.5 1   0 1 (0.000000000 1.000000000) *
##            39) PctBornSameState>=43.265 29   7 1 (0.241379310 0.758620690)
##              78) PctWorkMomYoungKids< 59.335 8   2 0 (0.750000000 0.250000000)
##               156) population< 103055 6   0 0 (1.000000000 0.000000000) *
##               157) population>=103055 2   0 1 (0.000000000 1.000000000) *
##              79) PctWorkMomYoungKids>=59.335 21   1 1 (0.047619048 0.952380952)
##               158) population< 11701 1   0 0 (1.000000000 0.000000000) *
##               159) population>=11701 20   0 1 (0.000000000 1.000000000) *
##       5) agePct12t21>=19.605 44  17 1 (0.386363636 0.613636364)
##        10) pctWPubAsst>=6.965 17   2 0 (0.882352941 0.117647059)
##          20) PctSameHouse85>=39.95 14   0 0 (1.000000000 0.000000000) *
##          21) PctSameHouse85< 39.95 3   1 1 (0.333333333 0.666666667)
##            42) householdsize>=3.115 1   0 0 (1.000000000 0.000000000) *
##            43) householdsize< 3.115 2   0 1 (0.000000000 1.000000000) *
##        11) pctWPubAsst< 6.965 27   2 1 (0.074074074 0.925925926)
##          22) householdsize< 2.56 1   0 0 (1.000000000 0.000000000) *
##          23) householdsize>=2.56 26   1 1 (0.038461538 0.961538462)
##            46) householdsize>=3.77 1   0 0 (1.000000000 0.000000000) *
##            47) householdsize< 3.77 25   0 1 (0.000000000 1.000000000) *
##     3) PctOccupMgmtProf>=30.32 444  45 1 (0.101351351 0.898648649)
##       6) OwnOccHiQuart< 83350 28  10 0 (0.642857143 0.357142857)
```

```
##         12) PctEmplManu>=12.3 17   0 0 (1.000000000 0.000000000) *
##         13) PctEmplManu< 12.3 11   1 1 (0.090909091 0.909090909)
##           26) householdsize< 2.25 1   0 0 (1.000000000 0.000000000) *
##           27) householdsize>=2.25 10   0 1 (0.000000000 1.000000000) *
##        7) OwnOccHiQuart>=83350 416  27 1 (0.064903846 0.935096154)
##         14) PctOccupMgmtProf< 33.135 77  22 1 (0.285714286 0.714285714)
##           28) agePct65up>=14.735 19   6 0 (0.684210526 0.315789474)
##             56) perCapInc< 20033.5 14   1 0 (0.928571429 0.071428571)
##               112) racepctblack< 11.845 13   0 0 (1.000000000 0.000000000) *
##               113) racepctblack>=11.845 1   0 1 (0.000000000 1.000000000) *
##             57) perCapInc>=20033.5 5   0 1 (0.000000000 1.000000000) *
##           29) agePct65up< 14.735 58   9 1 (0.155172414 0.844827586)
##             58) nonViol.Rate< 3.14485 20   9 1 (0.450000000 0.550000000)
##               116) PctOccupManu>=9.83 8   0 0 (1.000000000 0.000000000) *
##               117) PctOccupManu< 9.83 12   1 1 (0.083333333 0.916666667)
##                 234) population>=30841.5 1   0 0 (1.000000000 0.000000000) *
##                 235) population< 30841.5 11   0 1 (0.000000000 1.000000000) *
##             59) nonViol.Rate>=3.14485 38   0 1 (0.000000000 1.000000000) *
##         15) PctOccupMgmtProf>=33.135 339   5 1 (0.014749263 0.985250737)
##           30) householdsize>=3.36 12   3 1 (0.250000000 0.750000000)
##             60) population>=29453.5 3   0 0 (1.000000000 0.000000000) *
##             61) population< 29453.5 9   0 1 (0.000000000 1.000000000) *
##           31) householdsize< 3.36 327   2 1 (0.006116208 0.993883792)
##             62) PctSameHouse85>=75.015 2   1 0 (0.500000000 0.500000000)
##               124) population>=13729.5 1   0 0 (1.000000000 0.000000000) *
##               125) population< 13729.5 1   0 1 (0.000000000 1.000000000) *
##             63) PctSameHouse85< 75.015 325   1 1 (0.003076923 0.996923077)
##               126) PctVacantBoarded>=7.925 5   1 1 (0.200000000 0.800000000)
##                 252) householdsize>=2.84 1   0 0 (1.000000000 0.000000000) *
##                 253) householdsize< 2.84 4   0 1 (0.000000000 1.000000000) *
##               127) PctVacantBoarded< 7.925 320   0 1 (0.000000000 1.000000000) *
```

```
confusion(predict(crimedata.bag1, type="class"), training$Edu)
```

```
## $table
##    b
## a     0   1
##   0 836  46
##   1  37 382
##
## $misclass.prob
## [1] 0.06379708
```

```
confusion(predict(crimedata.bag1, testing, type="class"), testing$Edu)
```

```
## $table
##    b
## a     0   1
##   0 377  17
##   1  12 194
##
## $misclass.prob
## [1] 0.04833333
```

```
# For 50 bootstrap samples the test error increases
# The out-of-bag estimate of the misclassification error decreases
```

```
crimedata.bag2 <- bagging(as.factor(Edu) ~ ., data = training, nbagg=50, coob = T)
crimedata.bag2
```

```
##
## Bagging classification trees with 50 bootstrap replications
##
## Call: bagging.data.frame(formula = as.factor(Edu) ~ ., data = training,
##     nbagg = 50, coob = T)
##
## Out-of-bag estimate of misclassification error:  0.06
```

```
acc=(testing[,98]==predict(crimedata.bag2, testing))
1-length(acc[acc=="TRUE"])/length(acc)
```

```
## [1] 0.055
```

```
# rpart package
crimedata.bag3 <- bagging(as.factor(Edu) ~ ., data = training, coob = T)
crimedata.bag3
```

```
##
## Bagging classification trees with 25 bootstrap replications
##
## Call: bagging.data.frame(formula = as.factor(Edu) ~ ., data = training,
##     coob = T)
##
## Out-of-bag estimate of misclassification error:  0.0661
```

```
acc=(testing[,98]==predict(crimedata.bag3, testing))
1-length(acc[acc=="TRUE"])/length(acc)
```

```
## [1] 0.06166667
```

This method utilizes a lot more predictor variables compared to a single *Classification Tree*. The *ipred* package model provides a smaller test error of approximately *0.0483*. The out-of-bag estimate of the misclassification error from the training sample is approximately *0.0638*, which is higher than the test error. Since many of the individual features in the *Communities and Crime* are correlated, this might be due to the increased bias issue that the *Bagging* method suffers from. There are other possible explanations, however. For instance, this could mean that the method generalizes well.

Random Forest

The *Random Forest* method extends *Bagging* by decreasing the correlation among individual classifiers. In addition to subsetting individual observations (by bootstrapping), it also samples the features at each step which allows to de-correlate the classifiers. Thus, the classifiers from the bootstrap samples may contain different subsets of the predictors. As a rule of thumb, the number of variables tried at each split is roughly a square root of the total number of the features. This method allows assessing the relative importance of the features in terms of the overall prediction. It provides numerical measures and relative importance plots based on the mean decrease in terms of the different impurity measures, such as prediction error (*Accuracy*) or the *Gini Index*. The goal is still to achieve the largest reduction in terms of the impurity of a tree node.

The *Random Forest* allows to further reduce the misclassification error.

```
crimedata.tr2 <- training
crimedata.tr2$Edu <- as.factor(training$Edu)
crimedata.te2 <- testing
```

```
crimedata.te2$Edu <- as.factor(testing$Edu)


library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.5
```

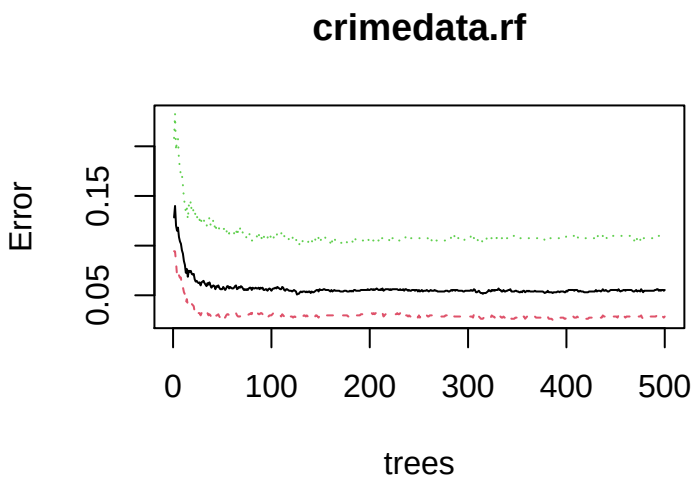
```
# Number of variables tried at each node by default is 9
# Increasing this # to 18 provides the lowest test error
crimedata.rf = randomForest(Edu ~., data=crimedata.tr2, mtry=18, importance=TRUE)
crimedata.rf
```

```
##
## Call:
##  randomForest(formula = Edu ~ ., data = crimedata.tr2, mtry = 18,      importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 18
##
##         OOB estimate of  error rate: 5.53%
## Confusion matrix:
##     0   1 class.error
## 0 848  25  0.02863688
## 1  47 381  0.10981308
```

```
#summary(crimedata.rf)
confusion(predict(crimedata.rf, crimedata.te2), crimedata.te2$Edu)
```

```
## $table
##    b
## a     0   1
##   0 382  19
##   1   7 192
##
## $misclass.prob
## [1] 0.04333333
```

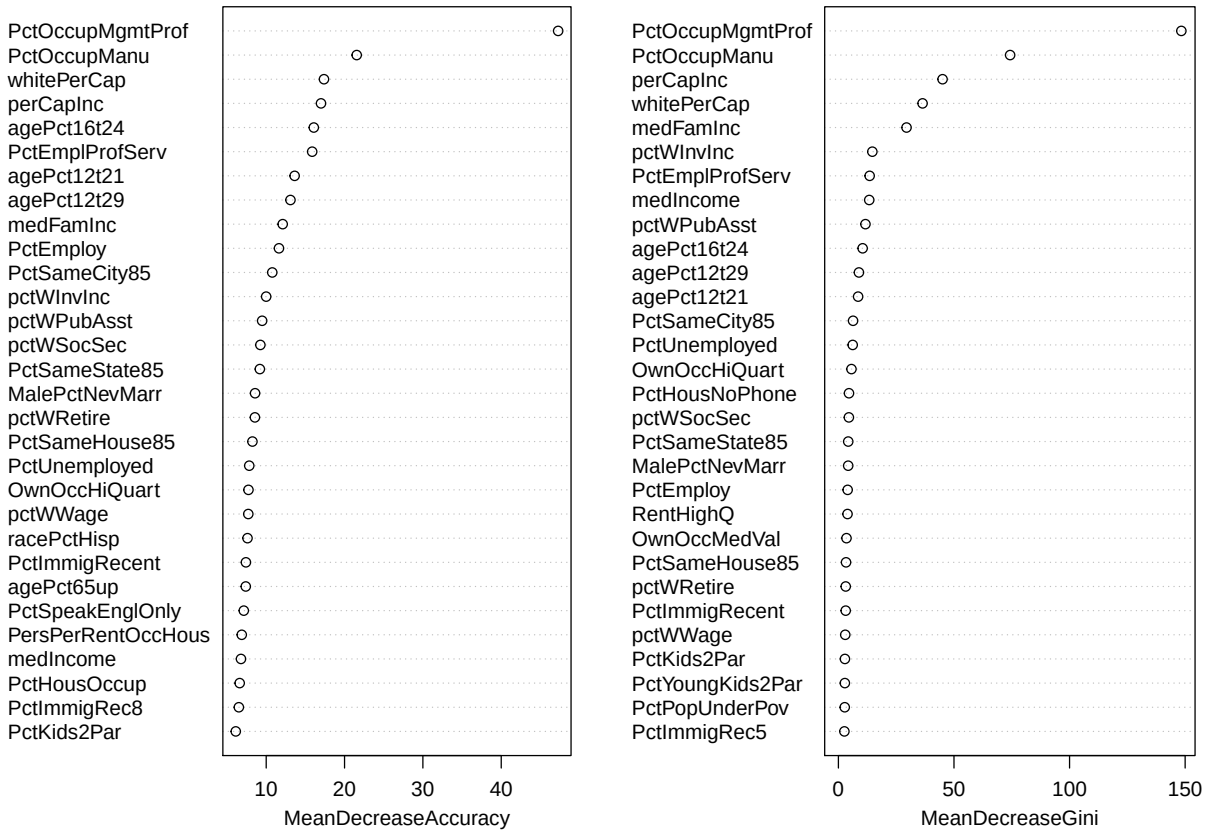
```
plot(crimedata.rf)
```



Numerical measures and relative importance plots based on the mean decrease in terms of the *Accuracy* and *Gini Index*:

```
varImpPlot(crimedata.rf, main = "Relative Importance Plots")
```

Relative Importance Plots




```
# relative importance of each feature
cbind(importance(crimedata.rf, type=1), importance(crimedata.rf, type=2))
```

```
##                     MeanDecreaseAccuracy MeanDecreaseGini
## population                      4.015030       1.67369362
## householdsize                   4.401126       1.00558392
## racepctblack                    4.292726       0.88888896
## racePctWhite                    5.111720       1.18514404
## racePctAsian                    5.291214       2.08900748
## racePctHisp                     7.612488       1.77981023
## agePct12t21                    13.630015       8.52737903
## agePct12t29                    13.096527       8.87935332
## agePct16t24                    16.075864      10.47817879
## agePct65up                      7.391787       2.49950136
## pctUrban                        1.698495       0.22885814
## medIncome                       6.780676      13.33681568
## pctWWage                        7.713565       2.95931771
## pctWFarmSelf                    3.181993       1.58412615
## pctWInvInc                      9.995827      14.67901639
## pctWSocSec                      9.254551       4.52943771
## pctWPubAsst                     9.482455      11.69279914
## pctWRetire                      8.560130       3.16450451
```

```
## medFamInc                 12.104582      29.47507973
## perCapInc                 16.997074      45.05562096
## whitePerCap               17.375704      36.41131245
## blackPerCap                3.196293       1.75221140
## indianPerCap               1.589964       1.07394216
## AsianPerCap                4.892502       1.34655762
## OtherPerCap                1.301616       0.94619830
## HispPerCap                 3.831516       1.08852409
## PctPopUnderPov             4.631853       2.73735663
## PctUnemployed              7.835672       6.18431433
## PctEmploy                 11.631441       4.00136662
## PctEmplManu                5.621030       1.96442067
## PctEmplProfServ           15.868388      13.52039863
## PctOccupManu              21.552559      74.29038055
## PctOccupMgmtProf          47.265270     148.35588906
## MalePctDivorce             5.893620       2.09556312
## MalePctNevMarr             8.581152       4.23500849
## FemalePctDiv               3.937287       1.04987557
## TotalPctDiv                5.452594       1.32878258
## PersPerFam                 3.821302       1.10799114
## PctFam2Par                 4.111412       2.05873210
## PctKids2Par                6.115704       2.81507225
## PctYoungKids2Par           4.701031       2.79008216
## PctTeen2Par                4.931001       0.94685990
## PctWorkMomYoungKids        1.881197       1.17414678
## PctWorkMom                 2.321699       1.28724987
## NumKidsBornNeverMar        4.970518       1.11598355
## PctKidsBornNeverMar        4.862013       1.08745054
## NumImmig                   5.389002       1.27781963
## PctImmigRecent             7.412630       3.15818694
## PctImmigRec5               5.710754       2.58804783
## PctImmigRec8               6.508879       2.11005886
## PctImmigRec10              5.556522       1.68413059
## PctRecentImmig             3.546492       1.10470287
## PctRecImmig5               4.591052       0.98708482
## PctRecImmig8               3.329792       1.00754257
## PctRecImmig10              4.550722       0.87814663
## PctSpeakEnglOnly           7.140261       1.99253667
## PctNotSpeakEnglWell        3.761784       1.10312519
## PctLargHouseFam            4.183783       1.10490510
## PctLargHouseOccup          3.787497       0.99941668
## PersPerOccupHous           4.740925       1.19483655
## PersPerOwnOccHous          4.031616       0.93971315
## PersPerRentOccHous         6.878611       2.34001334
## PctPersOwnOccup            3.315184       0.76152561
## PctPersDenseHous           4.741891       1.17070383
## PctHousLess3BR             4.563405       1.07784640
## MedNumBR                   2.154081       0.09481274
## HousVacant                 4.726327       1.69580205
## PctHousOccup               6.614569       2.30217148
## PctHousOwnOcc              4.370590       1.29743554
## PctVacantBoarded           3.217722       1.20370664
## PctVacMore6Mos             3.247085       1.46668298
## MedYrHousBuilt             4.144662       1.22325438
```

```
## PctHousNoPhone                5.229518       4.57755108
## PctWOFullPlumb                3.535807       1.03773935
## OwnOccLowQuart                5.353084       1.57155575
## OwnOccMedVal                  5.796589       3.49373979
## OwnOccHiQuart                 7.735238       5.64168427
## RentLowQ                      4.050894       1.50228009
## RentMedian                    4.135159       1.83997873
## RentHighQ                     4.718035       3.94131744
## MedRent                       3.449229       1.51653194
## MedRentPctHousInc             4.479201       1.31713081
## MedOwnCostPctInc              2.316636       1.07671540
## MedOwnCostPctIncNoMtg         1.876203       1.53308532
## NumInShelters                 4.571992       1.17659801
## NumStreet                     1.105376       0.32519645
## PctForeignBorn                5.233587       1.13628220
## PctBornSameState              5.802147       1.85645173
## PctSameHouse85                8.242030       3.30767055
## PctSameCity85                10.774718       6.34131786
## PctSameState85                9.189358       4.25215172
## LandArea                      4.025037       2.22213330
## PopDens                       1.613857       1.34209215
## PctUsePubTrans                2.393670       1.08531892
## LemasPctOfficDrugUn           2.717991       0.91104433
## Viol.Rate                     4.150618       1.11224202
## nonViol.Rate                  1.402458       1.03164087
```

The *Random Forest* method with 18 variables tried at each split has the training error of *0.0553* and the test error of *0.0433*. Again, the test error is lower than the training error.

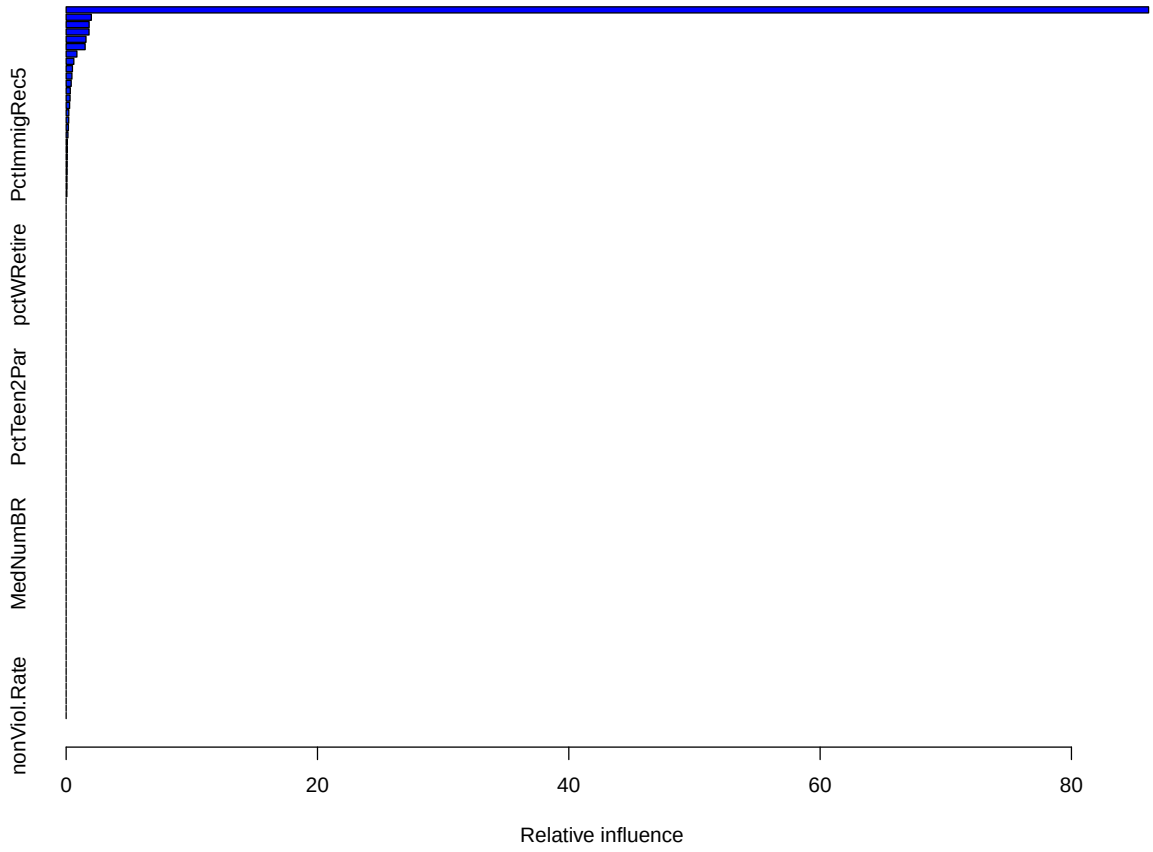
Boosting

While *Bagging* and *Random Forest* both build multiple classification trees in parallel, the *Boosting* uses ensemble learning in a sequential way instead of bootstrapping: it re-weights the original observations, giving the misclassified observations higher weights before constructing the next tree. One of the concerns with the ensemble methods is overfitting. *Boosting* is known to be resistant to overfitting.

```
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 4.0.5
```

```
crimedata.boost <- gbm(Edu ~ ., data = training, distribution = "bernoulli")
summary(crimedata.boost)
```




```
##                                    var     rel.inf
## PctOccupMgmtProf          PctOccupMgmtProf 86.13894330
## agePct16t24                    agePct16t24  2.00506474
## whitePerCap                    whitePerCap  1.82825694
## agePct12t29                    agePct12t29  1.82138377
## PctOccupManu                  PctOccupManu  1.57992706
## agePct12t21                    agePct12t21  1.51035972
## PctEmploy                        PctEmploy  0.85576614
## pctWSocSec                      pctWSocSec  0.61066332
## PctSameState85                PctSameState85  0.50022505
## PctEmplProfServ              PctEmplProfServ  0.45940675
## perCapInc                          perCapInc  0.40084760
## PctSameCity85                  PctSameCity85  0.32942034
## HousVacant                        HousVacant  0.31150087
## pctWPubAsst                      pctWPubAsst  0.27536900
## PctBornSameState          PctBornSameState  0.21242786
## PctVacMore6Mos                PctVacMore6Mos  0.20370994
## population                        population  0.18058417
## PctImmigRec5                    PctImmigRec5  0.14281555
## PctHousOwnOcc                  PctHousOwnOcc  0.09558133
## PctHousOccup                    PctHousOccup  0.09487887
## pctWWage                            pctWWage  0.08841822
```

```
## PctImmigRecent              PctImmigRecent  0.08229854
## LandArea                          LandArea  0.07903266
## PctImmigRec8                PctImmigRec8  0.06761793
## PersPerRentOccHous      PersPerRentOccHous  0.06709862
## MedOwnCostPctInc          MedOwnCostPctInc  0.05840171
## householdsize                householdsize  0.00000000
## racepctblack                  racepctblack  0.00000000
## racePctWhite                  racePctWhite  0.00000000
## racePctAsian                  racePctAsian  0.00000000
## racePctHisp                    racePctHisp  0.00000000
## agePct65up                      agePct65up  0.00000000
## pctUrban                          pctUrban  0.00000000
## medIncome                        medIncome  0.00000000
## pctWFarmSelf                  pctWFarmSelf  0.00000000
## pctWInvInc                      pctWInvInc  0.00000000
## pctWRetire                      pctWRetire  0.00000000
## medFamInc                        medFamInc  0.00000000
## blackPerCap                    blackPerCap  0.00000000
## indianPerCap                  indianPerCap  0.00000000
## AsianPerCap                    AsianPerCap  0.00000000
## OtherPerCap                    OtherPerCap  0.00000000
## HispPerCap                      HispPerCap  0.00000000
## PctPopUnderPov              PctPopUnderPov  0.00000000
## PctUnemployed                PctUnemployed  0.00000000
## PctEmplManu                    PctEmplManu  0.00000000
## MalePctDivorce              MalePctDivorce  0.00000000
## MalePctNevMarr              MalePctNevMarr  0.00000000
## FemalePctDiv                  FemalePctDiv  0.00000000
## TotalPctDiv                    TotalPctDiv  0.00000000
## PersPerFam                      PersPerFam  0.00000000
## PctFam2Par                      PctFam2Par  0.00000000
## PctKids2Par                    PctKids2Par  0.00000000
## PctYoungKids2Par          PctYoungKids2Par  0.00000000
## PctTeen2Par                    PctTeen2Par  0.00000000
## PctWorkMomYoungKids    PctWorkMomYoungKids  0.00000000
## PctWorkMom                      PctWorkMom  0.00000000
## NumKidsBornNeverMar    NumKidsBornNeverMar  0.00000000
## PctKidsBornNeverMar    PctKidsBornNeverMar  0.00000000
## NumImmig                          NumImmig  0.00000000
## PctImmigRec10                PctImmigRec10  0.00000000
## PctRecentImmig              PctRecentImmig  0.00000000
## PctRecImmig5                  PctRecImmig5  0.00000000
## PctRecImmig8                  PctRecImmig8  0.00000000
## PctRecImmig10                PctRecImmig10  0.00000000
## PctSpeakEnglOnly          PctSpeakEnglOnly  0.00000000
## PctNotSpeakEnglWell    PctNotSpeakEnglWell  0.00000000
## PctLargHouseFam              PctLargHouseFam  0.00000000
## PctLargHouseOccup          PctLargHouseOccup  0.00000000
## PersPerOccupHous            PersPerOccupHous  0.00000000
## PersPerOwnOccHous          PersPerOwnOccHous  0.00000000
## PctPersOwnOccup              PctPersOwnOccup  0.00000000
## PctPersDenseHous            PctPersDenseHous  0.00000000
## PctHousLess3BR              PctHousLess3BR  0.00000000
## MedNumBR                          MedNumBR  0.00000000
```

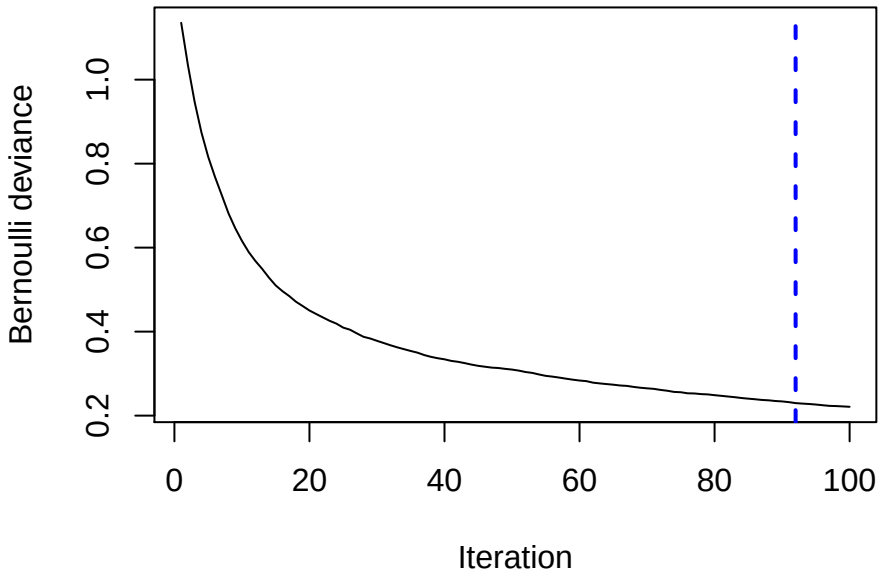
```
## PctVacantBoarded           PctVacantBoarded  0.00000000
## MedYrHousBuilt               MedYrHousBuilt  0.00000000
## PctHousNoPhone               PctHousNoPhone  0.00000000
## PctWOFullPlumb               PctWOFullPlumb  0.00000000
## OwnOccLowQuart               OwnOccLowQuart  0.00000000
## OwnOccMedVal                   OwnOccMedVal  0.00000000
## OwnOccHiQuart                 OwnOccHiQuart  0.00000000
## RentLowQ                           RentLowQ  0.00000000
## RentMedian                       RentMedian  0.00000000
## RentHighQ                         RentHighQ  0.00000000
## MedRent                             MedRent  0.00000000
## MedRentPctHousInc         MedRentPctHousInc  0.00000000
## MedOwnCostPctIncNoMtg MedOwnCostPctIncNoMtg  0.00000000
## NumInShelters                 NumInShelters  0.00000000
## NumStreet                         NumStreet  0.00000000
## PctForeignBorn               PctForeignBorn  0.00000000
## PctSameHouse85               PctSameHouse85  0.00000000
## PopDens                             PopDens  0.00000000
## PctUsePubTrans               PctUsePubTrans  0.00000000
## LemasPctOfficDrugUn     LemasPctOfficDrugUn  0.00000000
## Viol.Rate                         Viol.Rate  0.00000000
## nonViol.Rate                   nonViol.Rate  0.00000000
```

```
crimedata.boost
```

```
## gbm(formula = Edu ~ ., distribution = "bernoulli", data = training)
## A gradient boosted model with bernoulli loss function.
## 100 iterations were performed.
## There were 97 predictors of which 26 had non-zero influence.
```

We can see that *PctOccupMgmtProf* is by far the most important variable.

```
best.iter <- gbm.perf(crimedata.boost, method="OOB")
```




```
print(best.iter)
```

```
## [1] 92
## attr(,"smoother")
## Call:
## loess(formula = object$oobag.improve ~ x, enp.target = min(max(4,
##     length(x)/10), 50))
##
## Number of Observations: 100
## Equivalent Number of Parameters: 8.32
## Residual Standard Error: 0.001897
```

```
#summary(crimedata.boost, n.trees=best.iter)

crimedata.tr <- (predict(crimedata.boost, n.trees=best.iter)>0)*TRUE
confusion(crimedata.tr, training$Edu)
```

```
## $table
##    b
## a     0   1
##   0 850  33
##   1  23 395
##
## $misclass.prob
## [1] 0.04304381
```

```
crimedata.predict <- (predict(crimedata.boost, testing, n.trees=best.iter)>0)*TRUE
confusion(crimedata.predict, testing$Edu)
```

```
## $table
##    b
```

```
## a     0   1
##   0 378  17
##   1  11 194
##
## $misclass.prob
## [1] 0.04666667
```

The *Boosting* method has the training error of *0.0430* and the test error of *0.0467*. Again, the test error is lower than the training error.

Model Comparison

The table below summarizes the results of the classification analysis.

Model	Training error	Test error
LDA	4.38%	6.17%
CART	5.61%	6.33%
Bagging	6.38%	4.83%
RF	5.53%	4.33%
Boosting	4.30%	4.67%

The *Random Forest* model performs the best since it has the smallest misclassification rate on the test data set.