Communities and Violent Crimes study over the US population

Matteo Bergamaschi, Simone De Renzis, Andrea Marchini

Contents

1 Introduction **2**

2 Cleaning and filtering data **2**
2.1 Defining data sets for modelling . . . 4
2.2 Scaling all variables . . . 5

3 Residual analysis **9**
3.1 Original data . . . 9
3.2 Log transform of response variable . . . 11

4 Feature selection **14**
4.1 Multicollinearity . . . 14
4.1.1 VIF . . . 17
4.2 Stepwise selection . . . 18
4.2.1 Backward Selection . . . 18
Mallow's Cp . . . 18
BIC . . . 19
Adjusted R^2 . . . 20
4.2.2 Forward Selection . . . 22
Mallow's Cp . . . 22
BIC . . . 23
Adjusted R^2 . . . 23
4.3 Lasso regression . . . 25
4.4 Summing up the results . . . 27

5 Best regressor **30**
5.1 Ridge regression . . . 30
5.2 Linear regression with selected features . . . 30

6 Does ethnicity play a role? **33**
6.1 Correlations . . . 33
6.2 Partial correlations . . . 35

7 Data over US states **38**
7.1 Maps plot . . . 38
7.2 LASSO regression . . . 42

8 Conclusion **45**

Appendix: Attribute Information **46**

1 Introduction

In this report we perform a study of social aspects of the population of the United States of America, focusing on their influence over the number of violent crimes committed in every community. Various categories of features are considered, from the ethnic composition to the economic conditions of the different communities, from the level of education of people to the family background of children.

Through this paper we will analyze the dataset and its features, in order to find out which of them are more linked together and which ones have more influence over the total number of violent crimes. Several statistical tools will be used to train different regressors, in order to predict how many violent crimes will occur in a given community. Among the most relevant features we discovered that the family background plays a major role, making the rate of children with just one parent one of the most important factors, along with the average income of the population.

The second question aimed to find if there is a connection between the ethnic composition of communities and the number of violent crimes reported in that community: in particular we compared how the situation changes when the population is majority of African-american ethnicity, and Caucasian ethnicity. We discovered that the African-american ethnicity is associated with features that corresponds to a high level of crime, the opposite is true for the Caucasian ethnicity, and that by filtering out this "confounding" factors, the criminality rate is mostly independent from the ethnicity. This proved that the higher criminality rate that is reported in communities with a majority of African-american ethnicity is actually due to the worse socio-economic condition for the African-american population, with respect to people of Caucasian heritage.

At the end the focus will be on specific zones of the US territory, the ones with a peculiar presence of African-americans or people of Caucasian heritage.

2 Cleaning and filtering data

The dataset that we analyzed through this report is called "Communities and Crime" and can be found at the UCI website at the following link: http://archive.ics.uci.edu/ml/machine-learning-databases/00211/CommViolPredUnnormalizedData.txt.

```
base <- "http://archive.ics.uci.edu/ml/machine-learning-databases/"
file <- "00211/CommViolPredUnnormalizedData.txt"
communities.data = read.csv(paste(base, file, sep = ""), header = FALSE,
    na.strings = "?")
```

This dataset contains 2215 records of different communities, located in almost every state of the US, with 147 features describing social aspects of this different places. Note that a state is composed of more communities. Socio-economic data refers to the 1990's US census, law enforcement data from the 1990 US LEMAS survey and crime data from the 1995 FBI UCR. To see the complete list of the variables that will be part of our analysis and their meaning, please refer to Appendix at the end of this document. A brief summary of the features will be provided in the following sections.

To reduce the dimension of our dataset, we deleted several features such as the percentage of immigrants who immigrated within last 3, 5 and 8 years, which have not been considered of much interested for the study. There are also features which are presented with both a value and a percentage over the total number. For these cases, one of the two is dropped.

```
drops <- c("countyCode", "communityCode", "fold", "agePct65up", "numbUrban",
    "medFamInc", "perCapInc", "NumUnderPov", "PersPerFam", "PctImmigRecent",
    "PctImmigRec5", "PctImmigRec8", "PctImmigRec10", "PctRecImmig5", "PctRecImmig8",
    "PctRecImmig10", "PctNotSpeakEnglWell", "MedNumBR", "HousVacant", "PctHousOwnOcc",
    "PctVacantBoarded", "PctVacMore6Mos", "MedYrHousBuilt", "PctForeignBorn",
    "LemasSwornFT", "LemasSwFTFieldOps", "LemasTotalReq", "PolicReqPerOffic",
    "PolicOperBudg", "LemasPctPolicOnPatr", "LemasGangUnitDeploy", "LemasPctOfficDrugUn",
    "NumKindsDrugsSeiz", "LandArea", "OwnOccQrange", "RentQrange", "murders",
```

```
    "murdPerPop", "rapes", "rapesPerPop", "robberies", "robbbPerPop", "assaults",
    "assaultPerPop", "burglaries", "burglPerPop", "larcenies", "larcPerPop",
    "autoTheft", "autoTheftPerPop", "arsons", "arsonsPerPop", "nonViolPerPop",
    "NumKidsBornNeverMar")
communities <- communities.data[, !(names(communities.data) %in% drops)]
```

The second issues that we faced regards missing data. In fact, features as the rate of the different ethnicity in the local police and the number of officers assigned to special drug units have a significantly high number of missing (NA) values. These features mostly come from the LEMAS survey, which considered departments with at least 100 officiers, plus a random sample of other departments. So these features have not been taken into account for this study. In addition to that, 221 records do not present a value for the target (number of violent crimes per population), thus decreasing the number of observations from 2215 to 1994.

```
# detect columns with NA values
na <- colSums(is.na(communities))
na[which(na > 0)]
```

```
##          OtherPerCap      LemasSwFTPerPop LemasSwFTFieldPerPop
##                    1                 1872                 1872
##    LemasTotReqPerPop          PolicPerPop   RacialMatchCommPol
##                 1872                 1872                 1872
##        PctPolicWhite        PctPolicBlack         PctPolicHisp
##                 1872                 1872                 1872
##        PctPolicAsian        PctPolicMinor  OfficAssgnDrugUnits
##                 1872                 1872                 1872
##     PolicAveOTWorked           PolicCars      PolicBudgPerPop
##                 1872                 1872                 1872
##  ViolentCrimesPerPop
##                  221
```

```
communities <- communities[is.na(communities$ViolentCrimesPerPop) <= 0,
    ]
```

```
# drop columns with NA values
drops <- c("OtherPerCap", "LemasSwFTPerPop", "LemasSwFTFieldPerPop", "LemasTotReqPerPop",
    "RacialMatchCommPol", "PolicPerPop", "PctPolicWhite", "PctPolicBlack",
    "PctPolicHisp", "PctPolicAsian", "PctPolicMinor", "OfficAssgnDrugUnits",
    "PolicAveOTWorked", "PolicCars", "PolicBudgPerPop")
communities <- communities[, !(names(communities) %in% drops)]
```

The final dataset is made of 1994 records and 78 features. Only few features are not numerical data, such as the state or the name of the community.

The features of this dataset can be grouped into these categories:

- `Names`: string of characters providing the name of the communities and the abbreviation code of the state;
- `Population`: feature which provides the population of a given community;
- `Ethnicity`: features regarding the composition of the communities as the percentage of different ethnic groups over the total population;
- `Age`: features regarding the distribution of the people in age ranges;
- `Income`: features which details how people earn a living, from the percentage of households who have investments to the rate of people under the poverty level;
- `Education`: features which explain the level of education of the community;

- `Employment`: rates of people in the labor force with a job and their distribution across several sectors;
- `Family`: features regarding rates of divorce;
- `Children`: features regarding the conditions of kids in their families;
- `Immigrants`: features regarding regular immigrants;
- `Households`: attributes regarding quality of life in the houses, like the sanitary conditions or the mean number of people living in the same flat;
- `Homeless`: features describing number of homeless people.

2.1 Defining data sets for modelling

In order to create different models to predict the criminality rate of a given community, the dataset is split between a train set, with the 90% of the observations, and a test set, on which to perform predictions, consisting in 200 records (10%). We decided to perform the splitting at the very beginning of our elaboration, to assure that in the phase of variable selection no information is spilled to the test set which will then be used to assess the ability of our model to predict on new data.

This split is performed by ordering the dataset alphabetically by state, and extracting one record every ten to put it in the test set. This procedure ensures to have an high variety of states and communities. in both sets.

More in detail:

- `communities`: dataset that has been cleaned from NA values
  - `communities.test`: test set for `communities`
- `communities.mod`: dataset with only predictive features
  - `communities.mod.test`: test set for `communities.mod`
- `communities.lm`: dataset with only predictive features (and no response variable)
  - `communities.lm.test`: test set for `communities.lm`

```
communities.tt <- communities  #used later for modelling on zones
tmp.ordered <- communities[order(communities$state), ]  # sort by state
sampler <- seq(1, nrow(tmp.ordered), 10)  # indexes: 1 every 10 rows
communities.test <- tmp.ordered[sampler, ]  # pick 1 every 10 rows
communities <- tmp.ordered[-sampler, ]  # pick the remaining
communities.test <- communities.test[order(as.numeric(rownames(communities.test))),
    ]
communities <- communities[order(as.numeric(rownames(communities))), ]
```

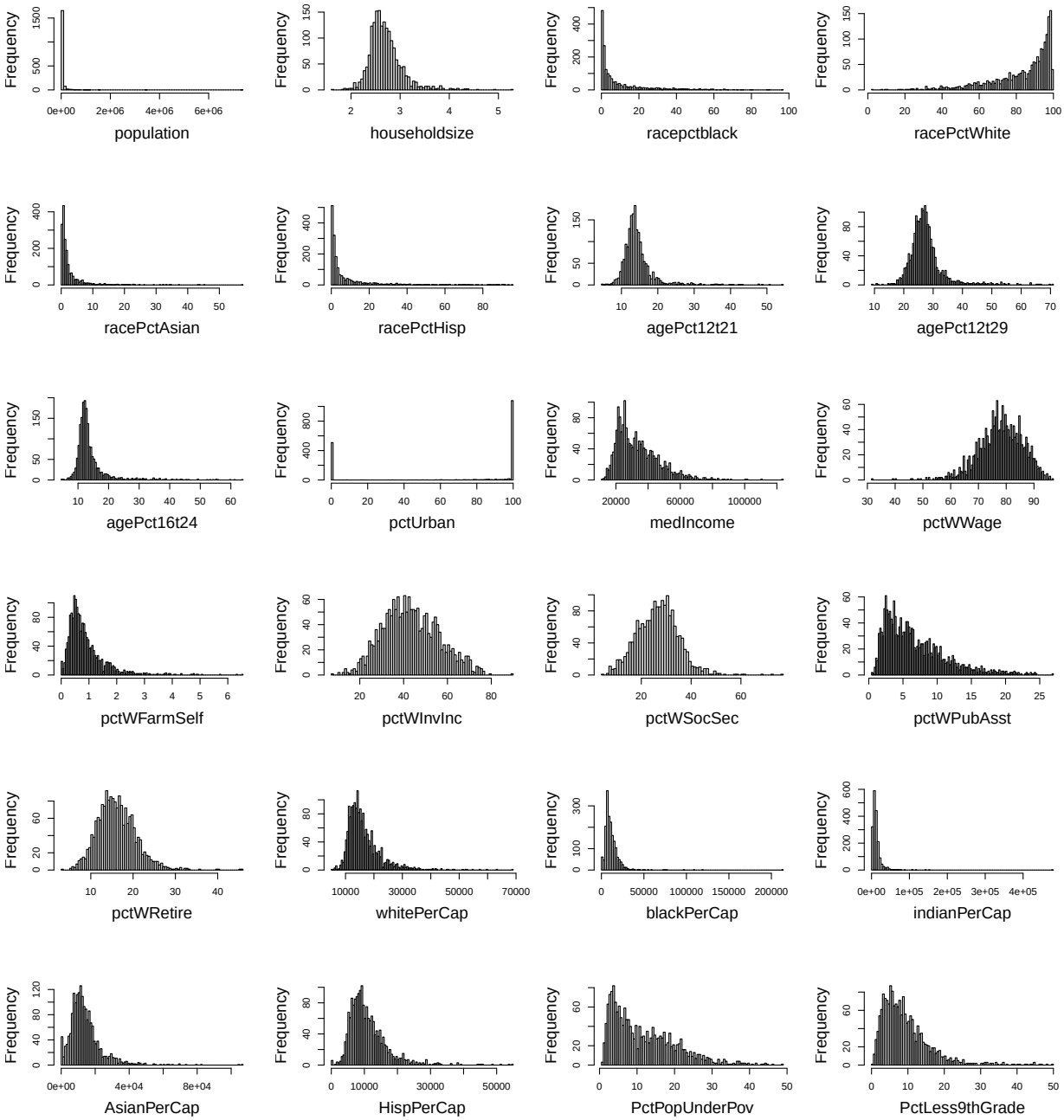
The small number of not numerical features is now dropped, in order to deal exclusively with predictive features.

```
drops <- c("state", "communityname")
communities.mod <- communities[, !(names(communities) %in% drops)]
communities.mod.test <- communities.test[, !(names(communities.test) %in%
    drops)]

drops <- c("state", "communityname", "ViolentCrimesPerPop")
communities.lm <- communities[, !(names(communities) %in% drops)]
communities.lm.test <- communities.test[, !(names(communities.test) %in%
    drops)]
```

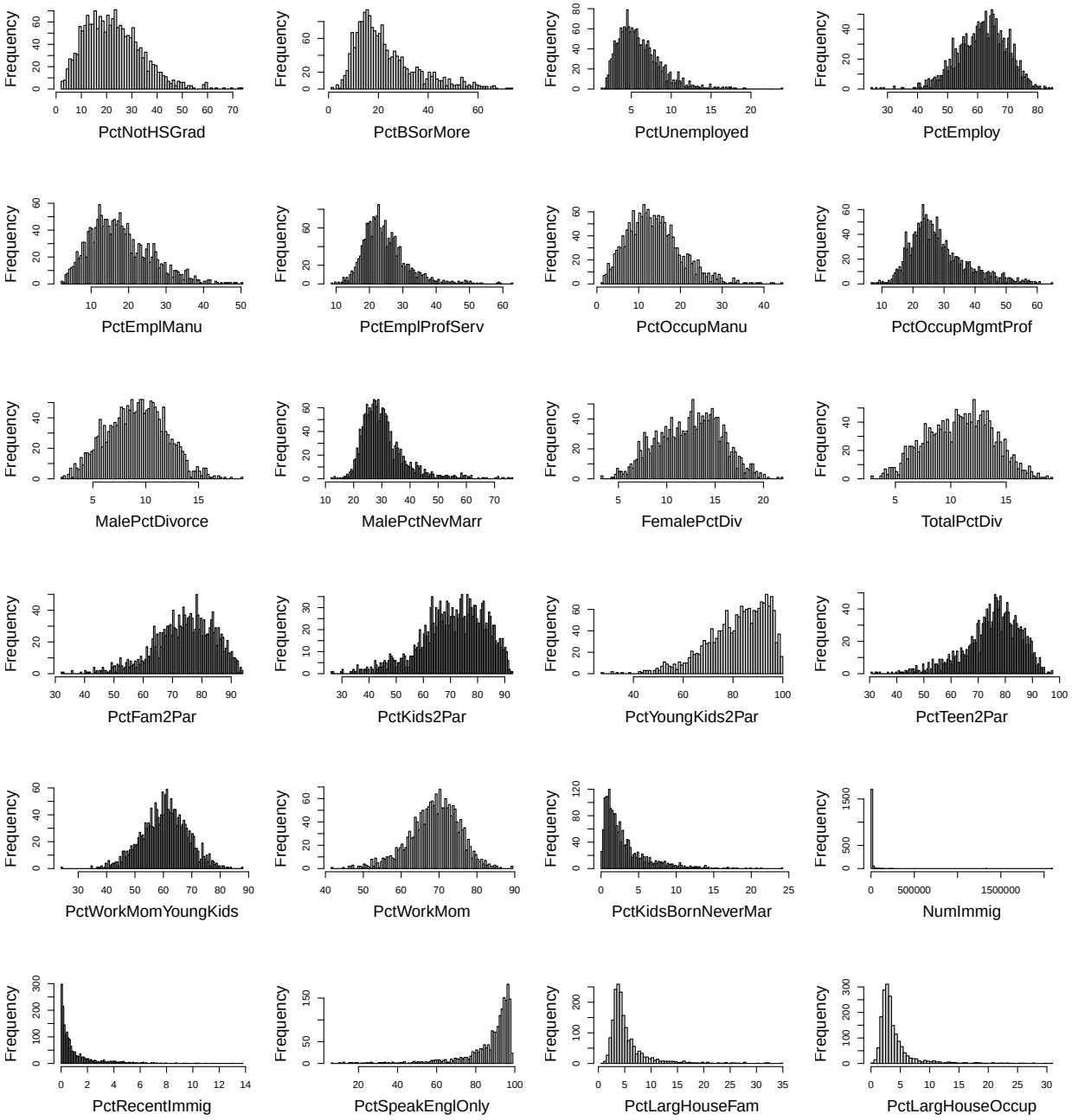
2.2 Scaling all variables

In order to visualize the distribution of data in every feature, histograms are displayed.

```
# plot histograms of first 24 features
par(mfrow = c(6, 4))
for (i in c(1:24)) {
    hist(data.matrix(communities.mod[i]), xlab = names(communities.mod[i]),
        main = NULL, cex.lab = 1.6, breaks = 100)
}
```

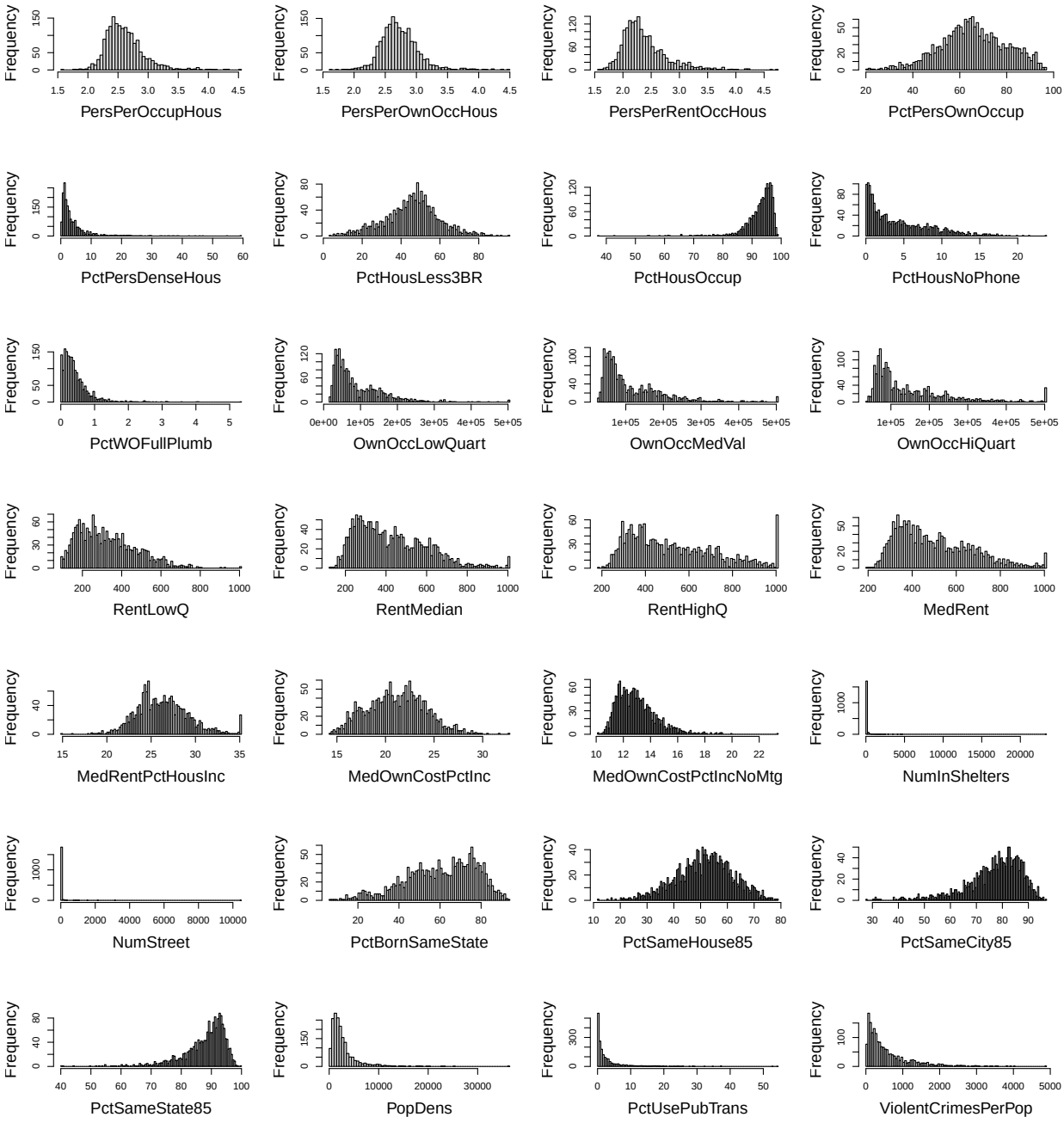


```r
# plot histograms of features between 25 and 48
par(mfrow = c(6, 4))
for (i in c(25:48)) {
    hist(data.matrix(communities.mod[i]), xlab = names(communities.mod[i]),
        main = NULL, cex.lab = 1.6, breaks = 100)
}
```




```r
# plot histogram of remaining features
par(mfrow = c(7, 4))
for (i in c(49:76)) {
    hist(data.matrix(communities.mod[i]), xlab = names(communities.mod[i]),
```

```
        main = NULL, cex.lab = 1.6, breaks = 100)
}
```




```
par(mfrow = c(1, 1))
```

As we can see from the plots, features lie in different ranges, also because a great part of them are percentages. In order to have all features in the same range, a preprocessing technique is applied, which scales values in the range $[0; 1]$.

```
library(caret)
```

```
# normalization of datasets

pp = preProcess(communities, method = "range")
communities <- predict(pp, communities)
communities.test <- predict(pp, communities.test)

pp = preProcess(communities.mod, method = "range")
communities.mod <- predict(pp, communities.mod)
communities.mod.test <- predict(pp, communities.mod.test)

pp = preProcess(communities.lm, method = "range")
communities.lm <- predict(pp, communities.lm)
communities.lm.test <- predict(pp, communities.lm.test)

attach(communities)
```

3 Residual analysis

3.1 Original data

After having cleaned the dataset from NA values, reduced the features number and normalized numerical values, a linear regression is provided, in order to determine a baseline of model fitting, before applying further and more advanced data transformations and regressions.

```
mod.out <- lm(ViolentCrimesPerPop ~ ., data = communities.lm)
summary(mod.out)
```

```
##
## Call:
## lm(formula = ViolentCrimesPerPop ~ ., data = communities.lm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32905 -0.03842 -0.00823  0.02701  0.48940
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.350668   0.147649   2.375 0.017658 *
## population              0.339874   0.253676   1.340 0.180490
## householdsize          -0.176601   0.096225  -1.835 0.066636 .
## racepctblack            0.214205   0.065794   3.256 0.001153 **
## racePctWhite            0.042145   0.061736   0.683 0.494912
## racePctAsian            0.016026   0.055958   0.286 0.774612
## racePctHisp            -0.032951   0.057068  -0.577 0.563742
## agePct12t21             0.105136   0.127073   0.827 0.408147
## agePct12t29            -0.448570   0.158959  -2.822 0.004829 **
## agePct16t24             0.271375   0.209268   1.297 0.194881
## pctUrban                0.025704   0.005580   4.606 4.40e-06 ***
## medIncome              -0.142657   0.106488  -1.340 0.180535
## pctWWage               -0.165350   0.081767  -2.022 0.043310 *
## pctWFarmSelf            0.044651   0.021681   2.059 0.039604 *
## pctWInvInc             -0.060125   0.045107  -1.333 0.182724
## pctWSocSec              0.064481   0.080806   0.798 0.424992
## pctWPubAsst             0.051831   0.034097   1.520 0.128669
## pctWRetire             -0.124768   0.033523  -3.722 0.000204 ***
## whitePerCap            -0.002811   0.078948  -0.036 0.971604
## blackPerCap            -0.021612   0.048324  -0.447 0.654762
## indianPerCap           -0.013805   0.055560  -0.248 0.803799
## AsianPerCap             0.041464   0.022319   1.858 0.063373 .
## HispPerCap              0.032536   0.022743   1.431 0.152725
## PctPopUnderPov         -0.121446   0.049532  -2.452 0.014310 *
## PctLess9thGrade        -0.226626   0.064453  -3.516 0.000449 ***
## PctNotHSGrad            0.142781   0.074993   1.904 0.057086 .
## PctBSorMore             0.071738   0.053075   1.352 0.176670
## PctUnemployed          -0.027688   0.037791  -0.733 0.463857
## PctEmploy               0.092500   0.061973   1.493 0.135728
## PctEmplManu            -0.039443   0.022749  -1.734 0.083121 .
## PctEmplProfServ        -0.020072   0.033273  -0.603 0.546418
## PctOccupManu           -0.008210   0.042444  -0.193 0.846639
## PctOccupMgmtProf        0.011366   0.057607   0.197 0.843619
## MalePctDivorce          0.723184   0.257600   2.807 0.005051 **
```

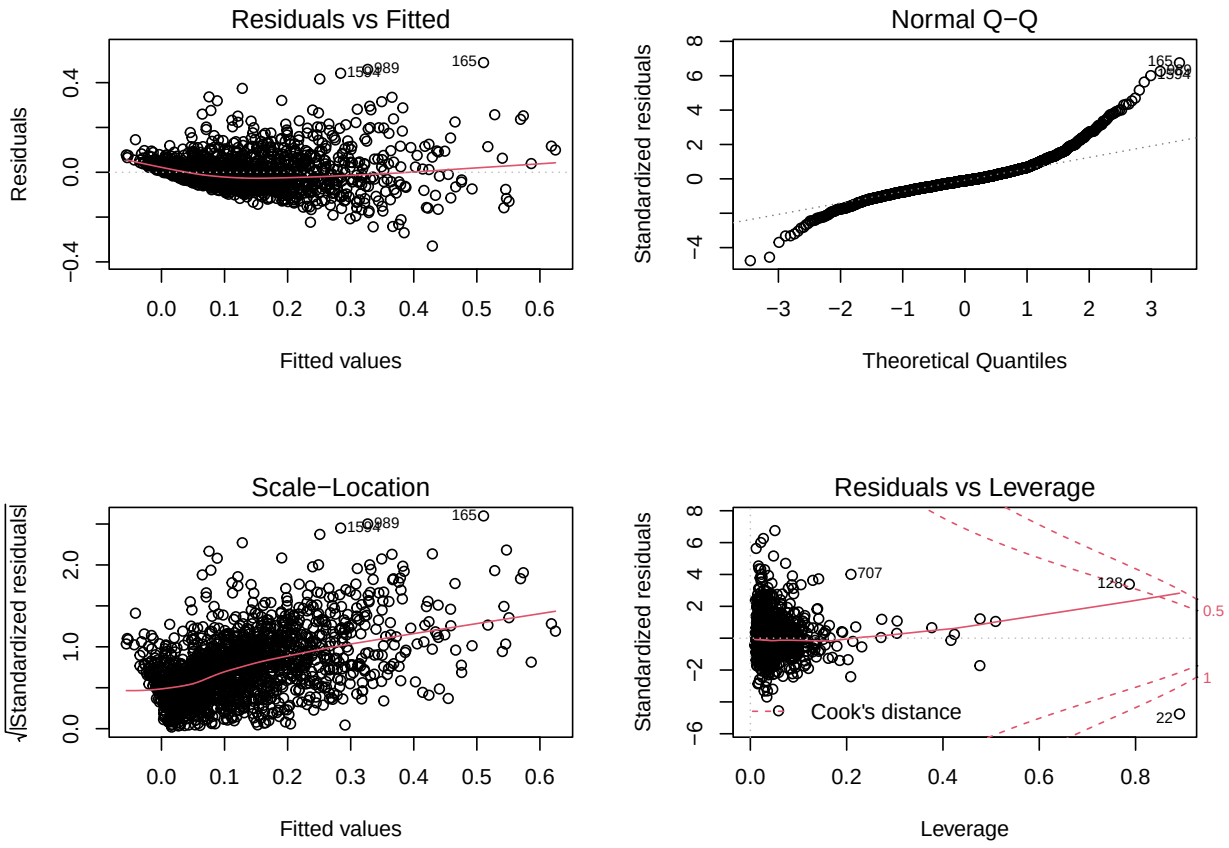
```
## MalePctNevMarr          0.119020   0.064695   1.840 0.065981 .
## FemalePctDiv            0.560530   0.295868   1.895 0.058323 .
## TotalPctDiv            -1.078034   0.501725  -2.149 0.031801 *
## PctFam2Par              0.061239   0.116482   0.526 0.599139
## PctKids2Par            -0.216477   0.105463  -2.053 0.040260 *
## PctYoungKids2Par        0.042397   0.037772   1.122 0.261836
## PctTeen2Par            -0.001709   0.031501  -0.054 0.956746
## PctWorkMomYoungKids     0.047675   0.037564   1.269 0.204555
## PctWorkMom             -0.114363   0.040158  -2.848 0.004454 **
## PctKidsBornNeverMar     0.236051   0.050526   4.672 3.22e-06 ***
## NumImmig               -0.103744   0.242509  -0.428 0.668855
## PctRecentImmig         -0.010907   0.037637  -0.290 0.772010
## PctSpeakEnglOnly       -0.105992   0.057363  -1.848 0.064813 .
## PctLargHouseFam         0.177780   0.167056   1.064 0.287391
## PctLargHouseOccup      -0.326054   0.184808  -1.764 0.077862 .
## PersPerOccupHous        0.630645   0.186093   3.389 0.000718 ***
## PersPerOwnOccHous      -0.150811   0.092567  -1.629 0.103451
## PersPerRentOccHous     -0.127580   0.063548  -2.008 0.044840 *
## PctPersOwnOccup        -0.071829   0.036515  -1.967 0.049330 *
## PctPersDenseHous        0.224221   0.087717   2.556 0.010667 *
## PctHousLess3BR          0.027374   0.036103   0.758 0.448430
## PctHousOccup           -0.048425   0.030963  -1.564 0.118010
## PctHousNoPhone          0.021224   0.028413   0.747 0.455176
## PctWOFullPlumb         -0.010359   0.029427  -0.352 0.724858
## OwnOccLowQuart          0.075977   0.139368   0.545 0.585717
## OwnOccMedVal           -0.009616   0.166835  -0.058 0.954045
## OwnOccHiQuart          -0.062582   0.071266  -0.878 0.379988
## RentLowQ               -0.139194   0.050355  -2.764 0.005767 **
## RentMedian             -0.038540   0.093125  -0.414 0.679033
## RentHighQ              -0.028523   0.050028  -0.570 0.568648
## MedRent                 0.172815   0.075010   2.304 0.021348 *
## MedRentPctHousInc       0.007195   0.022281   0.323 0.746776
## MedOwnCostPctInc       -0.002112   0.023428  -0.090 0.928173
## MedOwnCostPctIncNoMtg -0.086178   0.025600  -3.366 0.000779 ***
## NumInShelters           0.283073   0.284711   0.994 0.320243
## NumStreet              -0.307573   0.313772  -0.980 0.327104
## PctBornSameState       -0.007708   0.023346  -0.330 0.741322
## PctSameHouse85         -0.013293   0.037401  -0.355 0.722321
## PctSameCity85           0.008872   0.030032   0.295 0.767702
## PctSameState85          0.021755   0.038429   0.566 0.571397
## PopDens                -0.073533   0.043696  -1.683 0.092594 .
## PctUsePubTrans          0.020709   0.035840   0.578 0.563465
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0744 on 1718 degrees of freedom
## Multiple R-squared:  0.6716, Adjusted R-squared:  0.6572
## F-statistic: 46.84 on 75 and 1718 DF,  p-value: < 2.2e-16
```

The value observed for R^2 is 0.6716, while its adjusted version, $\bar{R}^2$, has value 0.6572.

The following 4 plots regard the behaviour of the linear model.

```
# plots regarding residual analysis
```

```
par(mfrow = c(2, 2))
mod.out <- lm(ViolentCrimesPerPop ~ ., data = communities.lm)
plot(mod.out)
```




```
par(mfrow = c(1, 1))
```

From the first plot we can see that residuals are almost horizontally distributed, with few values which lie far from the red line.

The Q-Q plot tells that the standardized residuals do not follow too well the assumption of normal distribution, given the strong tails that diverge from the theoretical line. We will try to handle this issue in the following sections.

From the Scale-Location graph we see that residuals are not so much equally distributed in all its range, thus not complying with the assumption of equal variance.

The last graph shows the presence of many outliers, the two most relevant corresponding to the records 128 and 22.

3.2 Log transform of response variable

A way to deal with the non compliance of the residuals with the hypothesis of normal distribution is to transform the response variable using a non linear function, in this case the choice is the logarithmic function. In order to handle the presence of 0's values in the response variable, the constant value 1 has been added to it before taking the logarithm.

```
# re-train the model applying the log to target values

ViolentCrimesPerPop.log <- log(ViolentCrimesPerPop + 1)
reg.out <- lm(ViolentCrimesPerPop.log ~ ., data = communities.lm)
summary(reg.out)$r.squared
```

```
## [1] 0.68915
```

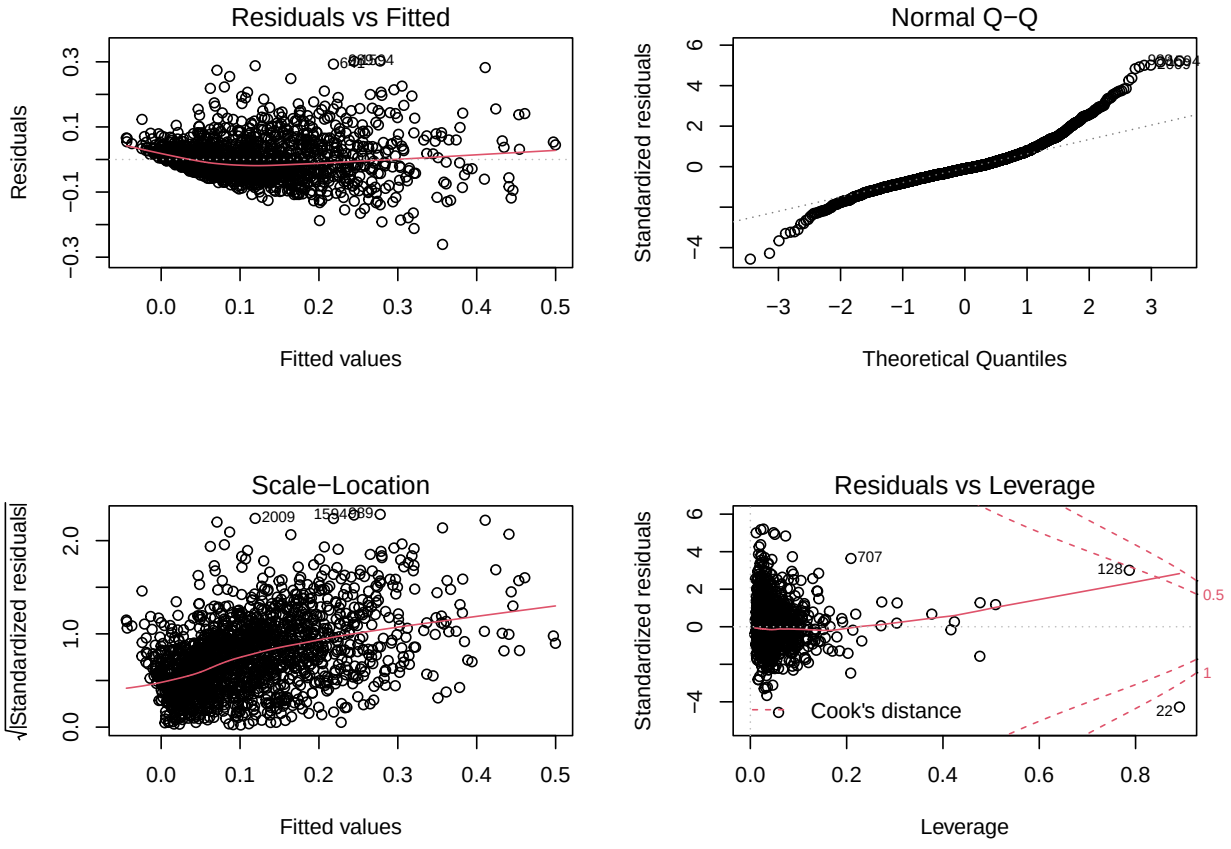
```
summary(reg.out)$adj.r.squared
```

```
## [1] 0.6755797
```

The transformation of data trough a logarithmic function brings better results for the numeric values R^2 and $\bar{R}^2$, which means that the model is slightly better than the previous one.

Let's now see from the plots how the linear regression model behaves, with this new response values.

```
par(mfrow = c(2, 2))
mod.out <- lm(ViolentCrimesPerPop.log ~ ., data = communities.lm)
plot(mod.out)
```




```
par(mfrow = c(1, 1))
```

From the first graph it is possible to see that the residuals lie in a smaller range than before.

Values in the Q-Q plot follow more the dotted diagonal, which means that now residuals' distribution is more similar to a normal distribution.

Records 22 and 128 are still highlighted as outliers.

Given this improvements, from now on, we will use the log transformed version for the response variable for linear regression models.

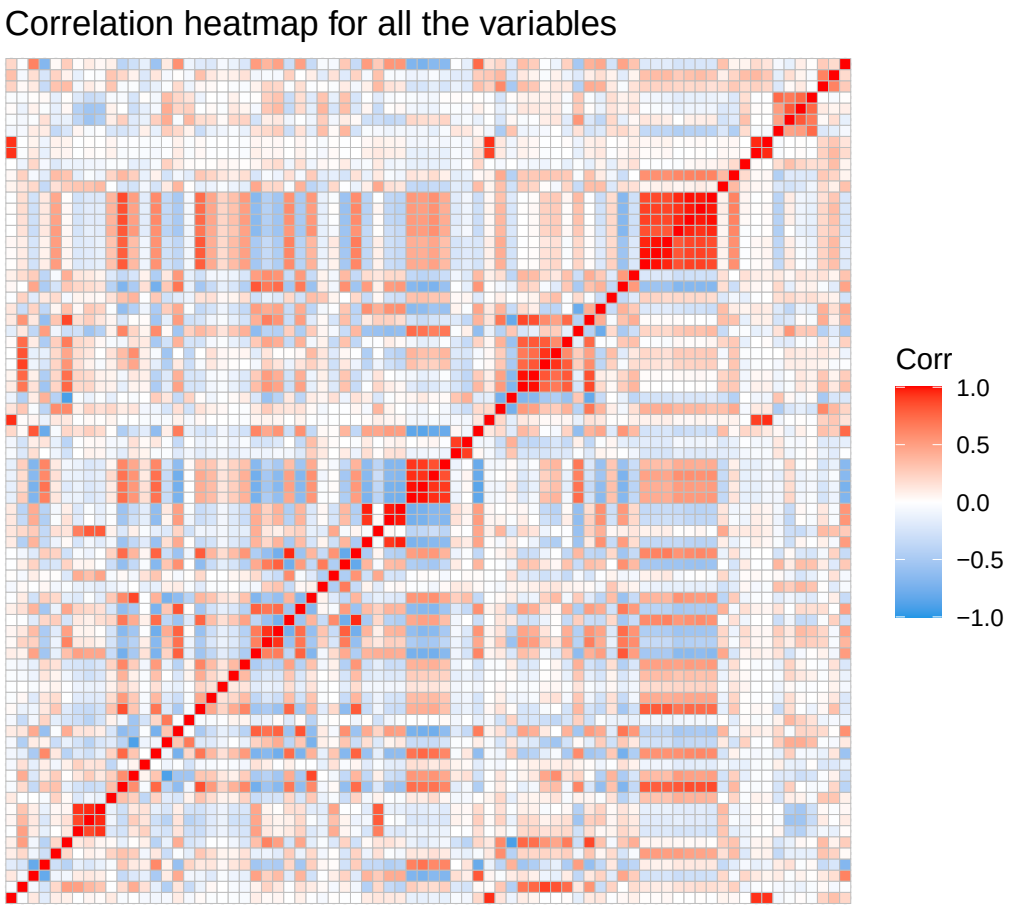
4 Feature selection

Previous regression models had to deal with 78 features, which is a high value. Easier models, which take into account only a minority of the features, can replace the previous linear regression, without a significant worsening of fit.

Therefore, the next and most important step is to detect and remove the less useful features in the dataset. To get an overview of the correlation between variables, a correlation matrix is computed.

4.1 Multicollinearity

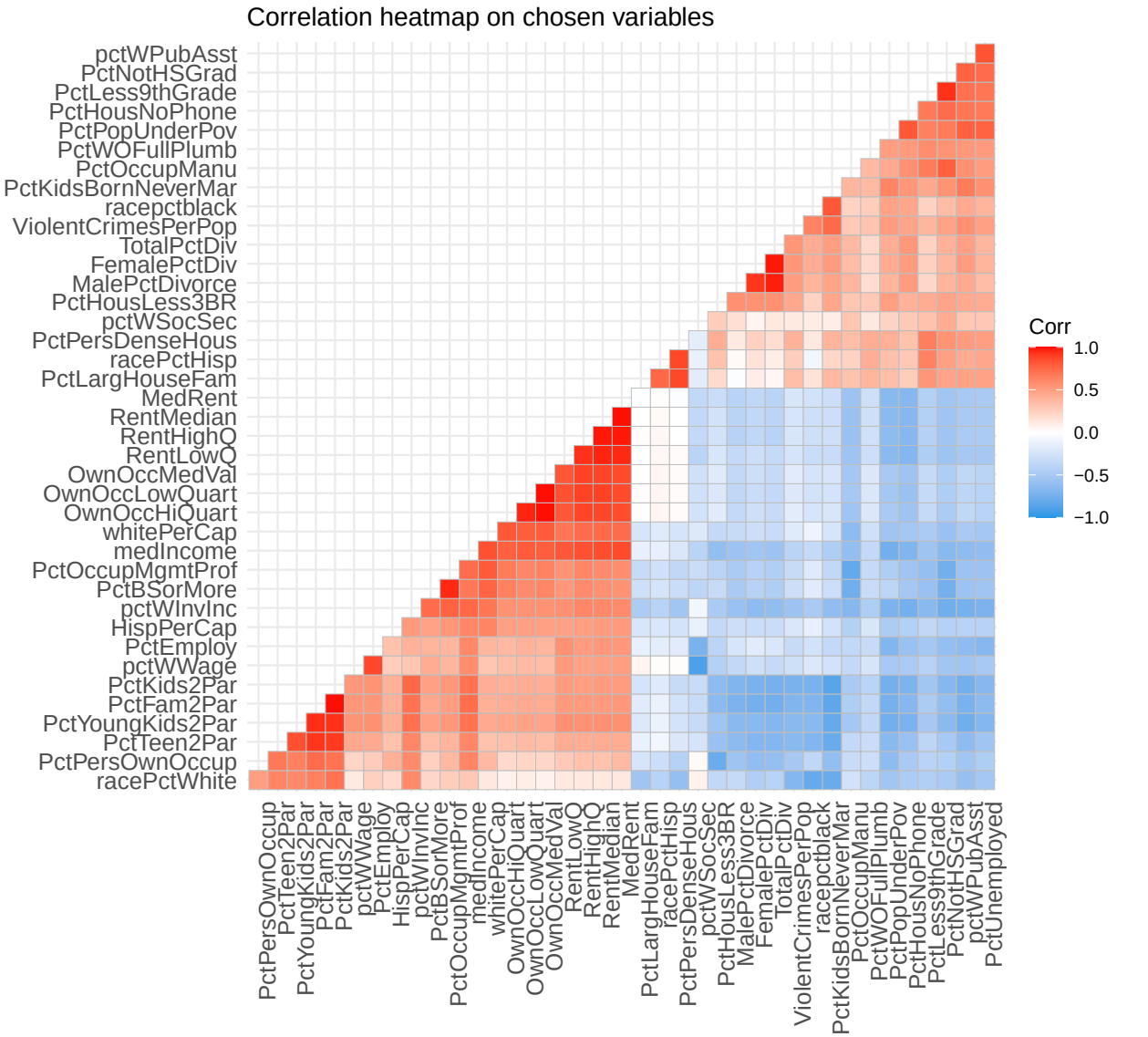
```
# display correlation matrix

corr <- cor(communities.mod)
library(ggcorrplot)
ggcorrplot(corr, tl.cex = 0, colors = c("1100FF", "white", "red"), tl.col = "white",
    title = "Correlation heatmap for all the variables")
```



To make the correlation plot easier to visualize, we computed the mean of the values of each row (and column) and kept only the rows (and columns) for which the mean of its elements is greater than a certain threshold. This allows to visualize only the one that are highly correlated with eachother.

To make the visualization even easier, the variables are clustered with a hierarchical clustering algorithm. In this way, variables that are most correlated with eachother are placed closer and it's easier to identify them. Also, only the lower diagonal is presented, in order to reduce the complexity of the plot.

```
corr.imp = cor(communities.mod)
diag(corr.imp) <- 0
to.remove = c()
for (i in c(1:dim(corr.imp)[1])) if (mean(abs(corr.imp)[i, ]) < 0.23) to.remove <- c(to.remove,
    i)
corr.imp <- corr.imp[-to.remove, ]
corr.imp <- corr.imp[, -to.remove]
ggcorrplot(corr.imp, colors = c("1100FF", "white", "red"), tl.srt = 90,
    hc.order = TRUE, type = "lower", title = "Correlation heatmap on chosen variables")
```



Now we want to identify highly correlated couples, each of whom will give a feature to delete from the model.

To do this, we filter and keep the pairs of variables that have a correlation > 0.9 between eachother.

```
# detect highly correlated pairs of features
upp <- lower.tri(corr, diag = FALSE)
corr.low <- replace(corr, !upp, 0)
corr.mask <- abs(corr.low) > 0.9
corrs <- which(corr.mask, arr.ind = TRUE, useNames = FALSE)

pair.a <- colnames(communities.mod)[corrs[, 1]]
pair.b <- colnames(communities.mod)[corrs[, 2]]

pairs.corr <- cbind(pair.a, pair.b)
pairs.corr
```

```
##       pair.a              pair.b
##  [1,] "NumImmig"          "population"
##  [2,] "NumInShelters"     "population"
##  [3,] "NumStreet"         "population"
##  [4,] "PctSpeakEnglOnly"  "racePctHisp"
##  [5,] "agePct16t24"       "agePct12t21"
##  [6,] "agePct16t24"       "agePct12t29"
##  [7,] "pctWSocSec"        "pctWWage"
##  [8,] "PctNotHSGrad"      "PctLess9thGrade"
##  [9,] "PctOccupMgmtProf"  "PctBSorMore"
## [10,] "FemalePctDiv"      "MalePctDivorce"
## [11,] "TotalPctDiv"       "MalePctDivorce"
## [12,] "TotalPctDiv"       "FemalePctDiv"
## [13,] "PctKids2Par"       "PctFam2Par"
## [14,] "PctYoungKids2Par"  "PctFam2Par"
## [15,] "PctTeen2Par"       "PctFam2Par"
## [16,] "PctYoungKids2Par"  "PctKids2Par"
## [17,] "PctTeen2Par"       "PctKids2Par"
## [18,] "NumStreet"         "NumImmig"
## [19,] "PctLargHouseOccup" "PctLargHouseFam"
## [20,] "PersPerOwnOccHous" "PersPerOccupHous"
## [21,] "OwnOccMedVal"      "OwnOccLowQuart"
## [22,] "OwnOccHiQuart"     "OwnOccLowQuart"
## [23,] "OwnOccHiQuart"     "OwnOccMedVal"
## [24,] "RentMedian"        "RentLowQ"
## [25,] "RentHighQ"         "RentLowQ"
## [26,] "MedRent"           "RentLowQ"
## [27,] "RentHighQ"         "RentMedian"
## [28,] "MedRent"           "RentMedian"
## [29,] "MedRent"           "RentHighQ"
## [30,] "NumStreet"         "NumInShelters"
```

4.1.1 VIF

The Variance Inflation Factor (VIF) is a measure of the amount of collinearity between a given set of features, in a regression model.

The VIF factor of $\hat{\beta}_j$ is computed with the formula:

$$VIF(\hat{\beta}\_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}}$$

.

The higher the VIF and the most collinearity is associated to that feature. A value of 5 or 10 is usually associated with a high problem of collinearity.

```
library(car)
mod.out <- lm(ViolentCrimesPerPop.log ~ ., data = communities.lm)
vifs <- sort(vif(mod.out), decreasing = TRUE)
head(vifs, 15)
```

```
##       TotalPctDiv      FemalePctDiv    MalePctDivorce      OwnOccMedVal
##        2724.08875         864.97776         575.48967         279.85167
##       agePct16t24  PersPerOccupHous PctLargHouseOccup    OwnOccLowQuart
##         141.27767         135.83081         130.12288         125.26963
##        PctFam2Par   PctLargHouseFam       PctKids2Par        RentMedian
##         125.05346         120.93063         115.46361         110.22585
##           MedRent       agePct12t29     OwnOccHiQuart
##          82.70208          82.54356          77.89692
```

```
tail(vifs, 15)
```

```
##           PctEmplManu      MedOwnCostPctInc            pctWRetire
##              4.794877             4.604908              4.364947
##               PopDens        PctUsePubTrans     MedRentPctHousInc
##              3.688187             3.603772              3.421963
## MedOwnCostPctIncNoMtg          PctHousOccup              pctUrban
##              2.539980             2.220927              2.009363
##            HispPerCap        PctWOFullPlumb          pctWFarmSelf
##              1.930611             1.907710              1.800525
##           blackPerCap           AsianPerCap          indianPerCap
##              1.509871             1.360196              1.108667
```

VIF's values are well above the 5-10 threshold for most of the variables. From the pairs of correlated variable, we now remove the member of the pair that has the highest VIF index: the idea is, for each pair of correlated variables, to keep the one that has the lighter multicollinearity problem.

```
# detect the feature with highest VIF in every couple

drops <- c()
for (i in c(1:length(pair.a))) {
    if (vifs[pair.a[i]] > vifs[pair.b[i]]) {
        drops <- c(drops, pair.a[i])
    } else {
        drops <- c(drops, pair.b[i])
    }
}
drops
```

```
##  [1] "population"        "NumInShelters"     "NumStreet"
```

```
##  [4] "racePctHisp"       "agePct16t24"       "agePct16t24"
##  [7] "pctWWage"          "PctNotHSGrad"      "PctBSorMore"
## [10] "FemalePctDiv"      "TotalPctDiv"       "TotalPctDiv"
## [13] "PctFam2Par"        "PctFam2Par"        "PctFam2Par"
## [16] "PctKids2Par"       "PctKids2Par"       "NumStreet"
## [19] "PctLargHouseOccup" "PersPerOccupHous"  "OwnOccMedVal"
## [22] "OwnOccLowQuart"    "OwnOccMedVal"      "RentMedian"
## [25] "RentHighQ"         "MedRent"           "RentMedian"
## [28] "RentMedian"        "MedRent"           "NumStreet"
```

These are the variables that we removed after this process.

```
# drop detected features

communities.cor1 <- communities.lm[, !(names(communities.lm) %in% drops)]
```

4.2 Stepwise selection

From the model obtained after filtering for correlation and VIF, we apply backward (and forward) stepwise selection to select the best attributes. Both backward and forward selection are computed with three different techniques: Mallow C_p, BIC and $\bar{R}^2$.

4.2.1 Backward Selection

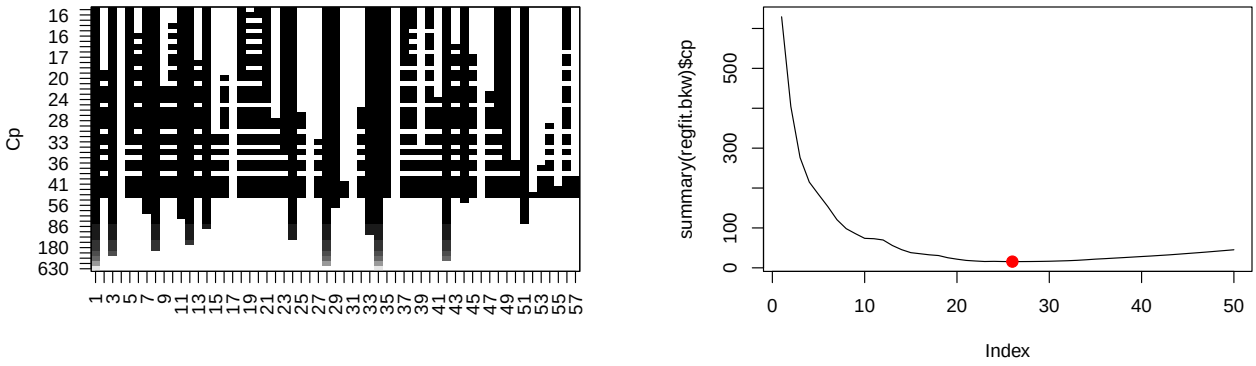
```
library(leaps)
```

```
## Warning: il pacchetto 'leaps' è stato creato con R versione 4.1.3
```

```
# perform backward selection
regfit.bkw <- regsubsets(ViolentCrimesPerPop.log ~ ., data = communities.cor1,
    nvmax = 50, method = "backward")
```

```

# plots of backward selection using Mallow C_p

par(mfrow = c(1, 2))
plot(regfit.bkw, scale = "Cp", labels = NULL)
plot(summary(regfit.bkw)$cp, type = "l")
min.cp <- which.min(summary(regfit.bkw)$cp)
points(min.cp, summary(regfit.bkw)$cp[min.cp], col = "red", cex = 2, pch = 20)
```

Mallow's Cp

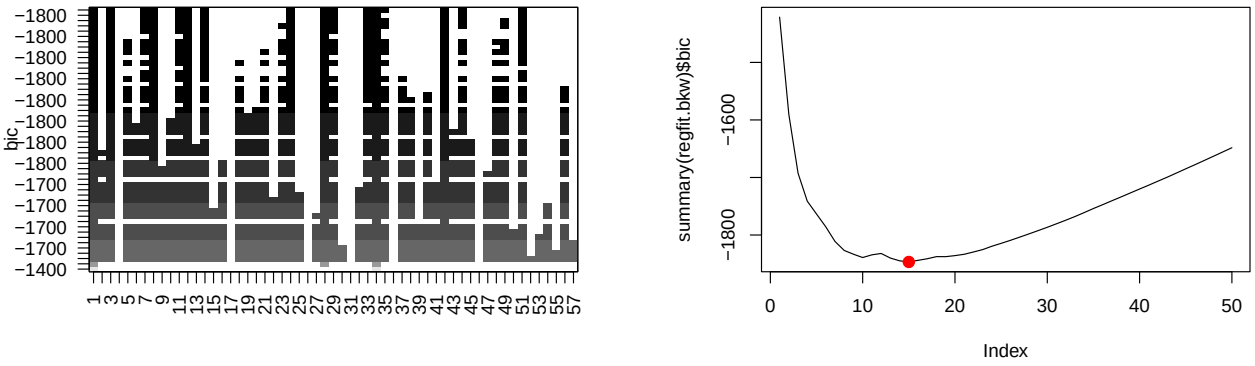


```r
# coefficients associated to the best model identified above

best.bkw.cp <- coef(regfit.bkw, min.cp)
best.bkw.cp
```

```
##           (Intercept)          racepctblack          racePctAsian
##            0.12192216            0.15001421            0.04601539
##          agePct12t29              pctUrban            pctWInvInc
##          -0.14388432            0.02102092           -0.07868379
##           pctWSocSec            pctWRetire           AsianPerCap
##           0.19828404           -0.09841806            0.03890744
##        PctPopUnderPov       PctLess9thGrade             PctEmploy
##          -0.03875247           -0.08161727            0.08665601
##          PctEmplManu        MalePctDivorce        MalePctNevMarr
##          -0.02809835            0.12476878            0.13863295
##           PctWorkMom   PctKidsBornNeverMar               NumImmig
##          -0.07956096            0.27292864            0.15601626
##      PctSpeakEnglOnly       PctLargHouseFam    PersPerRentOccHous
##          -0.05298329           -0.04742835            0.05029255
##      PctPersDenseHous          PctHousOccup              RentLowQ
##           0.17015897           -0.06514902           -0.08623350
##     MedRentPctHousInc MedOwnCostPctIncNoMtg               PopDens
##           0.03847294           -0.07779244           -0.04849019
```

```r
# plots of backward selection using BIC

par(mfrow = c(1, 2))
plot(regfit.bkw, scale = "bic", labels = NULL)
plot(summary(regfit.bkw)$bic, type = "l")
min.bic <- which.min(summary(regfit.bkw)$bic)
points(min.bic, summary(regfit.bkw)$bic[min.bic], col = "red", cex = 2,
    pch = 20)
```

BIC



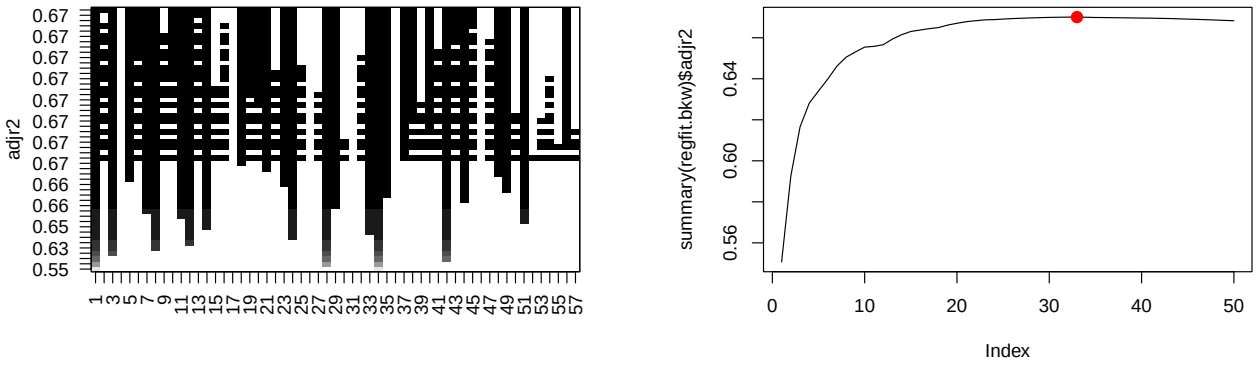

```r
par(mfrow = c(1, 1))
```

```r
# coefficients associated to the best model identified above

best.bkw.bic <- coef(regfit.bkw, min.bic)
best.bkw.bic
```

```
##            (Intercept)          racepctblack           agePct12t29
##             0.15959984            0.14098144           -0.16777194
##                pctUrban             pctWInvInc             pctWSocSec
##             0.02246333            -0.08061441             0.08468187
##             pctWRetire            PctEmplManu         MalePctDivorce
##            -0.07581803            -0.03557266             0.11610007
##         MalePctNevMarr             PctWorkMom    PctKidsBornNeverMar
##             0.11777949            -0.05369801             0.25837145
##                NumImmig       PctPersDenseHous           PctHousOccup
##             0.15429085             0.14010271            -0.07180897
## MedOwnCostPctIncNoMtg
##            -0.08227640
```

```
# plots of backward selection using adjusted R^2

par(mfrow = c(1, 2))
plot(regfit.bkw, scale = "adjr2", labels = NULL)
plot(summary(regfit.bkw)$adjr2, type = "l")
max.adjr2 <- which.max(summary(regfit.bkw)$adjr2)
points(max.adjr2, summary(regfit.bkw)$adjr2[max.adjr2], col = "red", cex = 2,
    pch = 20)
```

Adjusted R^2




```
par(mfrow = c(1, 1))
```

```
# coefficients associated to the best model identified above

best.bkw.adjr2 <- coef(regfit.bkw, max.adjr2)
best.bkw.adjr2
```

```
##            (Intercept)         householdsize          racepctblack
##             0.10841716            -0.05145884             0.14880167
##          racePctAsian           agePct12t21           agePct12t29
##             0.03733482             0.10501501            -0.21622394
##                pctUrban          pctWFarmSelf             pctWInvInc
##             0.02439646             0.02006856            -0.08384066
##             pctWSocSec           pctWPubAsst             pctWRetire
##             0.17528916             0.02753543            -0.08617026
##           AsianPerCap            HispPerCap         PctPopUnderPov
##             0.03418966             0.02356866            -0.08300862
##       PctLess9thGrade             PctEmploy            PctEmplManu
```

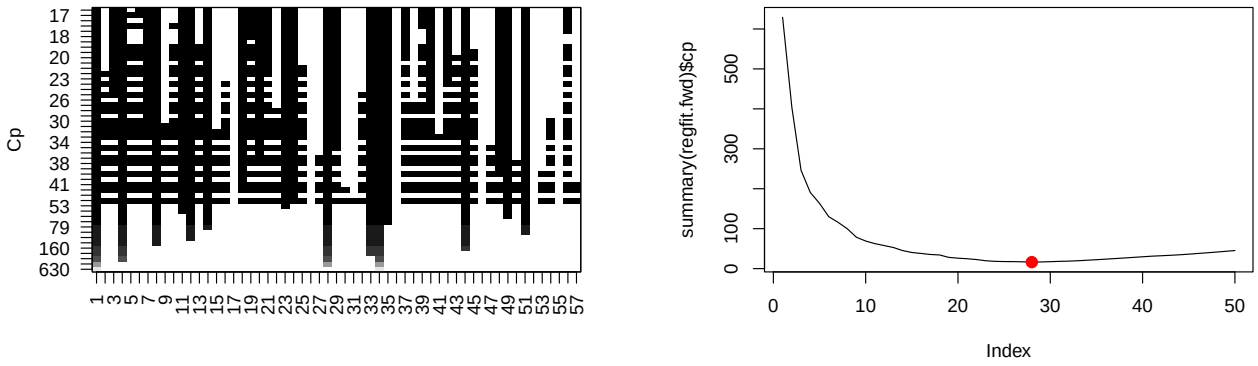
```
##           -0.09484525            0.09154115           -0.02539887
##        MalePctDivorce        MalePctNevMarr            PctWorkMom
##            0.11371291            0.15282455           -0.07887624
##   PctKidsBornNeverMar             NumImmig      PctSpeakEnglOnly
##            0.26580298            0.15591752           -0.05203819
##       PctLargHouseFam   PersPerRentOccHous      PctPersDenseHous
##           -0.04346998            0.06836384            0.17319861
##        PctHousLess3BR         PctHousOccup        PctHousNoPhone
##            0.02528786           -0.05981019            0.02928636
##              RentLowQ    MedRentPctHousInc MedOwnCostPctIncNoMtg
##           -0.08623171            0.03965401           -0.07105720
##               PopDens
##           -0.05410926
```

4.2.2 Forward Selection

```
library(leaps)
# perform forward selection

regfit.fwd <- regsubsets(ViolentCrimesPerPop.log ~ ., data = communities.cor1,
    nvmax = 50, method = "forward")
```

```
# plots of forward selection using Mallow C_p

par(mfrow = c(1, 2))
plot(regfit.fwd, scale = "Cp", labels = NULL)
plot(summary(regfit.fwd)$cp, type = "l")
min.cp <- which.min(summary(regfit.fwd)$cp)
points(min.cp, summary(regfit.fwd)$cp[min.cp], col = "red", cex = 2, pch = 20)
```

Mallow's Cp



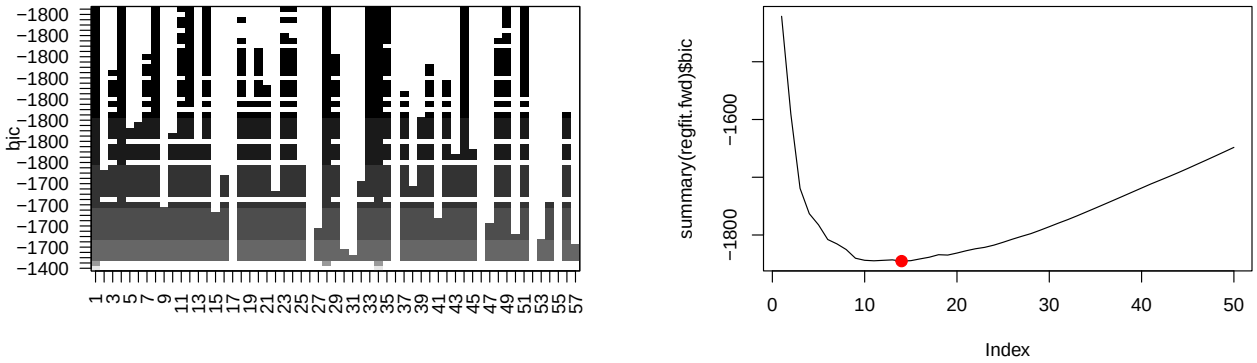

```
# coefficients associated to the best model identified above

best.fwd.cp <- coef(regfit.fwd, min.cp)
best.fwd.cp
```

```
##           (Intercept)          racepctblack          racePctWhite
##            0.19364967            0.10688302           -0.04313868
##           agePct12t21          agePct12t29             pctUrban
##            0.08166594           -0.20201400            0.02205673
##           pctWInvInc           pctWSocSec           pctWRetire
##           -0.09846003            0.17610102           -0.09384763
##           AsianPerCap           HispPerCap        PctPopUnderPov
##            0.03405023            0.02434368           -0.04663672
##       PctLess9thGrade            PctEmploy          PctEmplManu
##           -0.08614612            0.09408025           -0.02381247
##        MalePctDivorce        MalePctNevMarr           PctWorkMom
##            0.12029233            0.13337729           -0.07790728
##   PctKidsBornNeverMar              NumImmig      PctSpeakEnglOnly
##            0.27024348            0.15741890           -0.05890994
##     PersPerOwnOccHous    PersPerRentOccHous      PctPersDenseHous
##           -0.06103809            0.03858251            0.13533365
##          PctHousOccup              RentLowQ     MedRentPctHousInc
##           -0.05888170           -0.08191166            0.03842442
## MedOwnCostPctIncNoMtg               PopDens
```

```
##           -0.07124320           -0.04262494
```

```
# plots of forward selection using BIC


par(mfrow = c(1, 2))
plot(regfit.fwd, scale = "bic", labels = NULL)
plot(summary(regfit.fwd)$bic, type = "l")
min.bic <- which.min(summary(regfit.fwd)$bic)
points(min.bic, summary(regfit.fwd)$bic[min.bic], col = "red", cex = 2,
    pch = 20)
```

BIC



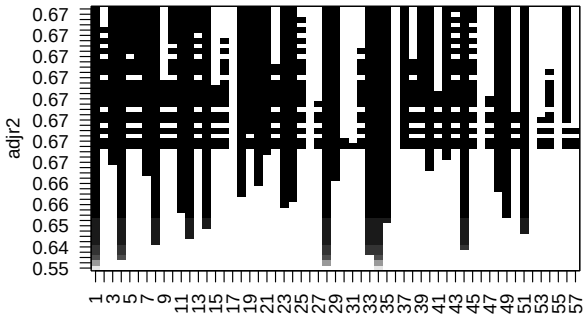

```
par(mfrow = c(1, 1))
```

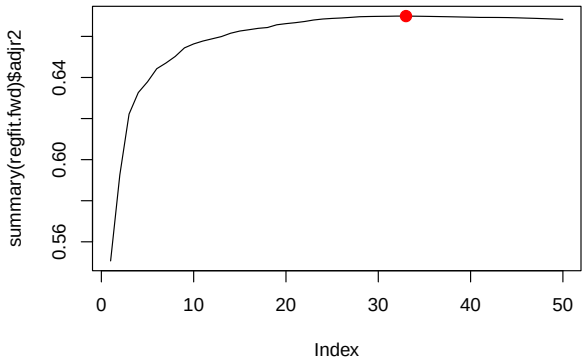
```
# coefficients associated to the best model identified above

best.fwd.bic <- coef(regfit.fwd, min.bic)
best.fwd.bic
```

```
##           (Intercept)          racePctWhite              pctUrban
##            0.17967876            -0.14134241            0.02397484
##            pctWInvInc              pctWSocSec            pctWRetire
##           -0.06094576             0.19259881           -0.08310283
##             PctEmploy           PctEmplManu        MalePctDivorce
##            0.07776521            -0.03087204             0.12055983
##            PctWorkMom   PctKidsBornNeverMar               NumImmig
##           -0.07155928             0.31695292             0.16611344
##          PctHousOccup     MedRentPctHousInc MedOwnCostPctIncNoMtg
##           -0.07478277             0.03297785            -0.08017816
```

```
# plots of forward selection using adjusted R^2

par(mfrow = c(1, 2))
plot(regfit.fwd, scale = "adjr2", labels = NULL)
plot(summary(regfit.fwd)$adjr2, type = "l")
max.adjr2 <- which.max(summary(regfit.fwd)$adjr2)
points(max.adjr2, summary(regfit.fwd)$adjr2[max.adjr2], col = "red", cex = 2,
    pch = 20)
```

Adjusted R^2






```
par(mfrow = c(1, 1))
```

```
# coefficients associated to the best model identified above

best.fwd.adjr2 <- coef(regfit.fwd, max.adjr2)
best.fwd.adjr2
```

```
##           (Intercept)          racepctblack          racePctWhite
##           0.132528806           0.145249022          -0.001745548
##          racePctAsian           agePct12t21           agePct12t29
##           0.037230451           0.097402927          -0.215317208
##              pctUrban          pctWFarmSelf            pctWInvInc
##           0.024464015           0.018218380          -0.092008317
##            pctWSocSec           pctWPubAsst            pctWRetire
##           0.160702129           0.026568907          -0.084041802
##           AsianPerCap            HispPerCap        PctPopUnderPov
##           0.034287816           0.024009632          -0.075609759
##       PctLess9thGrade             PctEmploy           PctEmplManu
##          -0.092166631           0.097389124          -0.023635009
##        MalePctDivorce        MalePctNevMarr            PctWorkMom
##           0.108219833           0.134899771          -0.077887351
##   PctKidsBornNeverMar             NumImmig      PctSpeakEnglOnly
##           0.266923741           0.154007676          -0.055833776
##     PersPerOwnOccHous    PersPerRentOccHous      PctPersDenseHous
##          -0.058563621           0.040379735           0.147487300
##        PctHousLess3BR         PctHousOccup        PctHousNoPhone
##           0.023964238          -0.059253366           0.025470482
##               RentLowQ    MedRentPctHousInc MedOwnCostPctIncNoMtg
##          -0.083711001           0.040283711          -0.069114135
##                PopDens
##          -0.048702148
```

4.3 Lasso regression

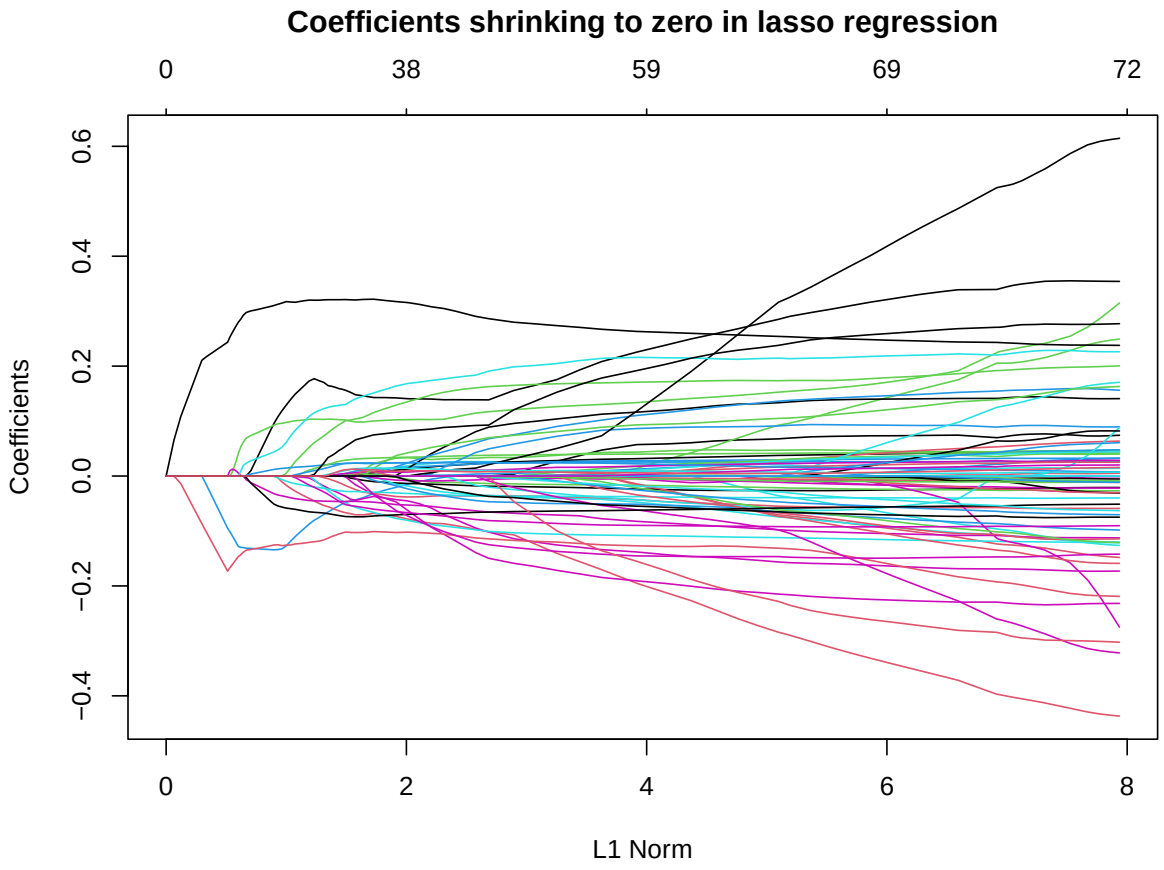
The LASSO regression represents a valid alternative to the traditional linear regression. They differ because LASSO adds a penalty term, whose aim is to shrink to 0 coefficients related to less important features for the model. We perform LASSO regression on the full set of features.

```
library(glmnet)
# alpha=0 for ridge, alpha=1 for lasso
lasso.mod <- glmnet(communities.lm, ViolentCrimesPerPop, alpha = 1)
best.lasso <- coef(lasso.mod, s = 0.002)  #specific lambda to make a comparison
# best.lasso <- predict(lasso.mod, s=10, type='coefficients')
best.lasso <- data.frame(as.matrix(best.lasso))
colnames(best.lasso) <- "lasso"
best.lasso <- best.lasso[best.lasso$lasso != 0, , drop = FALSE]
best.lasso
```

```
##                               lasso
## (Intercept)            0.2654405709
## population             0.1639435757
## racepctblack           0.0811044616
## racePctWhite          -0.0636620733
## agePct12t29           -0.0568729166
## pctUrban               0.0210296215
## pctWWage              -0.0234824916
## pctWInvInc            -0.0058159938
## pctWRetire            -0.0289167922
## AsianPerCap            0.0218918130
## PctEmplManu           -0.0241897853
## MalePctDivorce         0.1034186963
## PctKids2Par           -0.1117497604
## PctWorkMom            -0.0459517921
## PctKidsBornNeverMar    0.3208415735
## PctPersDenseHous       0.1257167301
## PctHousLess3BR         0.0009539489
## PctHousOccup          -0.0690626531
## MedRentPctHousInc      0.0161799603
## MedOwnCostPctIncNoMtg -0.0351204011
## NumInShelters          0.0351834789
## PctSameCity85          0.0037529360
## PctUsePubTrans         0.0026632490
```

With the following plot we can see how this regression shrinks to 0 many coefficients, by increasing the λ constant.

```
# plots of LASSO coefficents

par(mar = c(4, 5, 6, 2))
plot(lasso.mod, main = "Coefficients shrinking to zero in lasso regression")
```


Coefficients shrinking to zero in lasso regression
0
38
59
69
72
Coefficients
0.6
0.4
0.2
0.0
−0.2
−0.4
0
2
4
6
8
L1 Norm

4.4 Summing up the results

We performed 3 methods to select a subset of important features:

- Correlation + VIF + Forward selection
- Correlation + VIF + Backward selection
- LASSO regression

Each one of this method gave us a subset of important features. Now we want to summarize this into a unique set of important features that can help us to get some insights on the "risk factors" for violent crime.

Among the three, the BIC is the one that penalises the most models with many variables. So we choose it in order to compare stepwise selections with LASSO regression.

```
# pick best configurations using BIC technique
best.bkw.bic <- data.frame(best.bkw.bic)
best.fwd.bic <- data.frame(best.fwd.bic)
```

These important features are selected by summing the coefficients of the three methods and picking the highest values according to their absolute value.

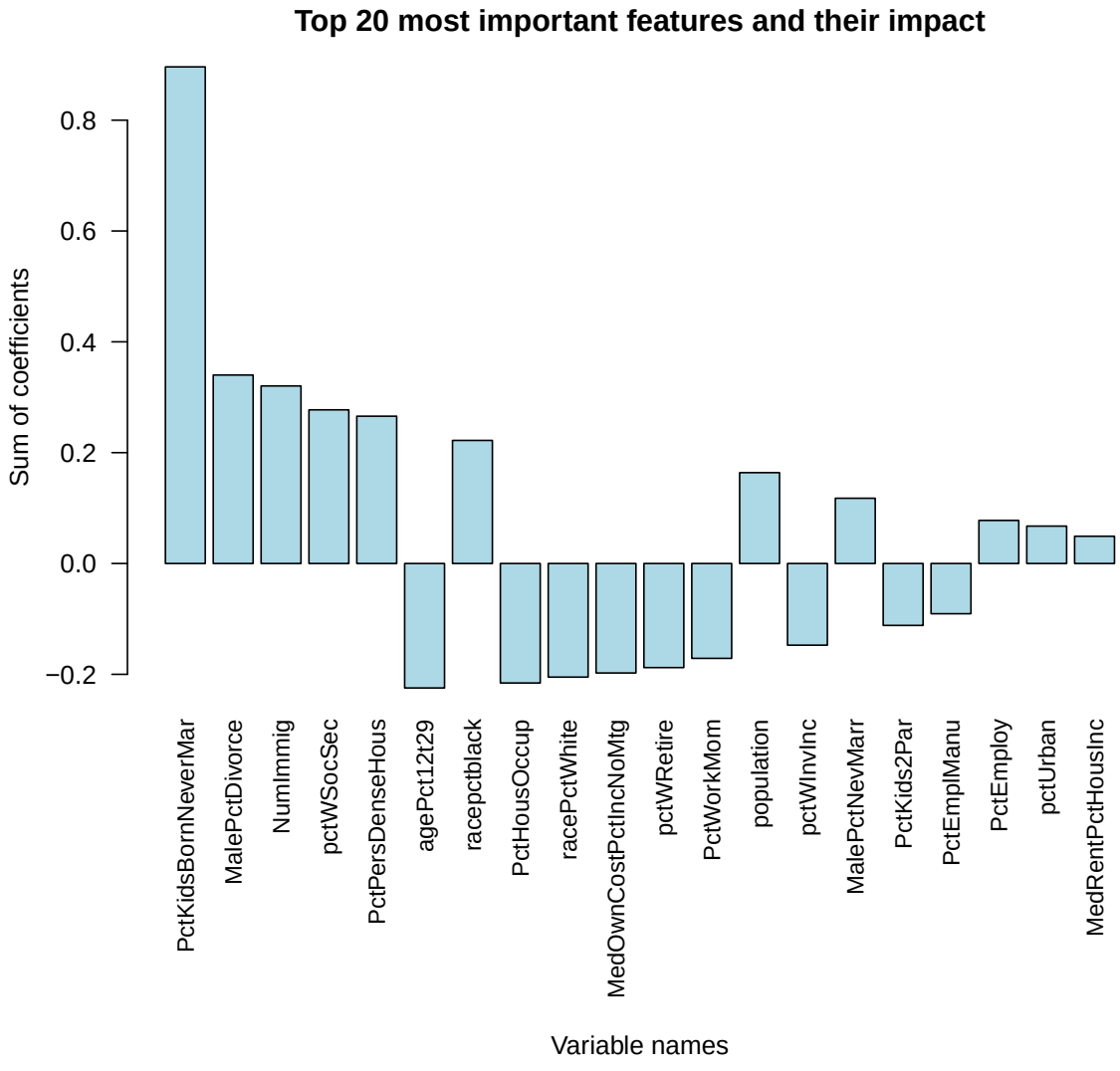
```
library(dplyr)
library(tibble)

#sum coefficients of different methods
important.features <- full_join(best.lasso %>% rownames_to_column(),
          best.bkw.bic %>% rownames_to_column(), by = "rowname") %>%
  full_join(.,best.fwd.bic %>% rownames_to_column(), by = "rowname") %>% # perfom outer join
  mutate_if(is.numeric,coalesce,0) %>% # na's to 0
  mutate(Result = lasso + best.bkw.bic + best.fwd.bic) # sum each column

important.features <- important.features[order(abs(important.features$Result),

                                               decreasing = TRUE),] # order by sum

important.features <- important.features[important.features$rowname != "(Intercept)", ,

                                         drop = FALSE]
important.features
```

```
##                   rowname         lasso best.bkw.bic best.fwd.bic        Result
## 15   PctKidsBornNeverMar  0.3208415735   0.25837145   0.31695292  0.8961659440
## 12        MalePctDivorce  0.1034186963   0.11610007   0.12055983  0.3400786016
## 26              NumImmig  0.0000000000   0.15429085   0.16611344  0.3204042963
## 24            pctWSocSec  0.0000000000   0.08468187   0.19259881  0.2772806866
## 16      PctPersDenseHous  0.1257167301   0.14010271   0.00000000  0.2658194356
## 5            agePct12t29 -0.0568729166  -0.16777194   0.00000000 -0.2246448585
## 3           racepctblack  0.0811044616   0.14098144   0.00000000  0.2220859029
## 18          PctHousOccup -0.0690626531  -0.07180897  -0.07478277 -0.2156543883
## 4           racePctWhite -0.0636620733   0.00000000  -0.14134241 -0.2050044809
## 20 MedOwnCostPctIncNoMtg -0.0351204011  -0.08227640  -0.08017816 -0.1975749618
## 9             pctWRetire -0.0289167922  -0.07581803  -0.08310283 -0.1878376508
## 14            PctWorkMom -0.0459517921  -0.05369801  -0.07155928 -0.1712090886
## 2             population  0.1639435757   0.00000000   0.00000000  0.1639435757
## 8             pctWInvInc -0.0058159938  -0.08061441  -0.06094576 -0.1473761713
## 25        MalePctNevMarr  0.0000000000   0.11777949   0.00000000  0.1177794899
```

```
## 13           PctKids2Par -0.1117497604   0.00000000   0.00000000 -0.1117497604
## 11           PctEmplManu -0.0241897853  -0.03557266  -0.03087204 -0.0906344901
## 27             PctEmploy  0.0000000000   0.00000000   0.07776521  0.0777652101
## 6                pctUrban  0.0210296215   0.02246333   0.02397484  0.0674677906
## 19     MedRentPctHousInc  0.0161799603   0.00000000   0.03297785  0.0491578087
## 21         NumInShelters  0.0351834789   0.00000000   0.00000000  0.0351834789
## 7                pctWWage -0.0234824916   0.00000000   0.00000000 -0.0234824916
## 10            AsianPerCap  0.0218918130   0.00000000   0.00000000  0.0218918130
## 22         PctSameCity85  0.0037529360   0.00000000   0.00000000  0.0037529360
## 23        PctUsePubTrans  0.0026632490   0.00000000   0.00000000  0.0026632490
## 17        PctHousLess3BR  0.0009539489   0.00000000   0.00000000  0.0009539489
```

The following histogram plots the 20 most important features.

```
par(mar = c(12, 5, 3, 2))
barplot(`colnames<-`(rbind(important.features[1:20, ]$Result), (important.features[1:20,
    ]$rowname)), las = 2, main = "Top 20 most important features and their impact",
    col = "lightblue", cex.names = 0.9)
title(xlab = "Variable names", mgp = c(11, 1, 0))
title(ylab = "Sum of coefficients", mgp = c(3, 1, 0))
```

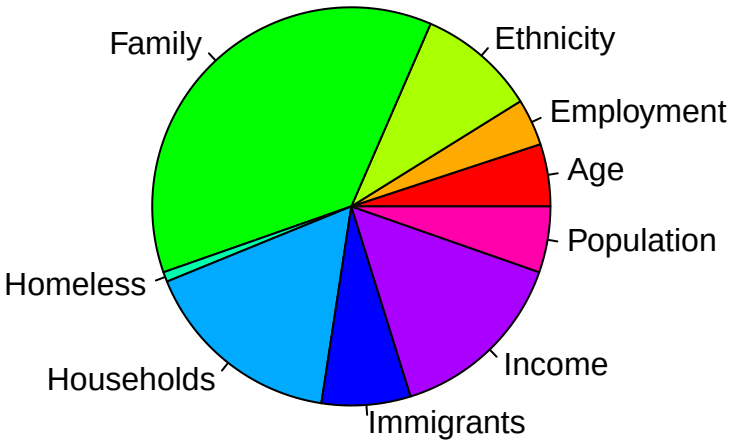


```
feature <- colnames(communities)
cat <- c("Names", "Names", "Population", "Households", "Ethnicity", "Ethnicity",
    "Ethnicity", "Ethnicity", "Age", "Age", "Age", "Population", "Income",
    "Income", "Income", "Income", "Income", "Income", "Income", "Income",
    "Income", "Income", "Income", "Income", "Income", "Education", "Education",
    "Education", "Employment", "Employment", "Employment", "Employment",
    "Employment", "Employment", "Family", "Family", "Family", "Family",
    "Family", "Family", "Family", "Family", "Family", "Family", "Family",
    "Immigrants", "Immigrants", "Immigrants", "Households", "Households",
    "Households", "Households", "Households", "Households", "Households",
    "Households", "Households", "Households", "Households", "Households",
    "Households", "Households", "Households", "Households", "Households",
    "Households", "Households", "Households", "Households", "Homeless",
    "Homeless", "Immigrants", "Households", "Population", "Population",
    "Population", "Population", "Target")
features.data <- data.frame(feature, cat)
```

```
# associated every 'important feature' to its category

important.features.cat <- inner_join(important.features, features.data,
    by = c(rowname = "feature"))
```

With the following pie chart, the aim is to visualize which categories of features are the most relevant for our models.

```
par(mar=c(0,3,0,3))
#pie chart of categories
important.cat <- aggregate(abs(important.features.cat$Result),

                           by=list(Category=important.features.cat$cat),

                           FUN=sum)

pie(important.cat$x, labels = important.cat$Category,
    col = rainbow(9) )
```



As we can see, familiar aspects are the most relevant, together income and household aspects.

5 Best regressor

After having determined the most important features that are linked to the target variable, we want to test if this subset of features can be used to perform prediction on new data. In practice, we want to test the generalization capabilities of our model.

5.1 Ridge regression

Ridge regression is another type of regularized regression: in this case an $L2$ norm penalty term is applied to shrink coefficients. This technique is of less help in detecting a subset of relevant features, since the coefficients are not usually shrank to 0, but provides a model that does not overfit on training data and should generalize well. So, in our case, it is used to compare its performances with a linear model trained only on the selected subset of features in order to evaluate the generalization capabilities of our model. Note that the ridge regression model is trained on the whole set of features.

```
# Ridge regression
ridge.mod <- cv.glmnet(as.matrix(communities.lm), ViolentCrimesPerPop,
    alpha = 0, nfolds = 5)
ridge.mod$lambda.min  #best lambda found with cross validation
```

```
## [1] 0.009404901
```

```
predictions.ridge.train <- predict(ridge.mod, as.matrix(communities.lm),
    s = ridge.mod$lambda.min)
predictions.ridge.test <- predict(ridge.mod, as.matrix(communities.lm.test),
    s = ridge.mod$lambda.min)
```

```
# evaluate Ridge regressions over train and test sets
library(Metrics)
ridge.train <- rmse(ViolentCrimesPerPop, predictions.ridge.train)
ridge.test <- rmse(communities.mod.test$ViolentCrimesPerPop, predictions.ridge.test)
ridge.train
```

```
## [1] 0.07430465
```

```
ridge.test
```

```
## [1] 0.07426181
```

5.2 Linear regression with selected features

The results provided by the Ridge regression are compared to a linear regressor which considers only the selected features as in section 4.4.

```
mod.out <- lm(ViolentCrimesPerPop ~ ., data = communities.lm[important.features$rowname[-2]])
summary(mod.out)
```

```
##
## Call:
## lm(formula = ViolentCrimesPerPop ~ ., data = communities.lm[important.features$rowname[-2]])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32897 -0.03736 -0.00768  0.02472  0.52228
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.361108   0.065643   5.501 4.33e-08 ***
```

```
## PctKidsBornNeverMar    0.265373   0.041835   6.343 2.85e-10 ***
## NumImmig              -0.328279   0.198088  -1.657 0.097649 .
## pctWSocSec             0.030965   0.057840   0.535 0.592465
## PctPersDenseHous       0.122341   0.046102   2.654 0.008033 **
## agePct12t29           -0.236662   0.054969  -4.305 1.76e-05 ***
## racepctblack           0.145750   0.040637   3.587 0.000344 ***
## PctHousOccup          -0.082960   0.027197  -3.050 0.002320 **
## racePctWhite          -0.000138   0.036802  -0.004 0.997008
## MedOwnCostPctIncNoMtg -0.112213   0.021466  -5.227 1.92e-07 ***
## pctWRetire            -0.078816   0.025468  -3.095 0.002001 **
## PctWorkMom            -0.074377   0.018919  -3.931 8.77e-05 ***
## population             0.458568   0.246991   1.857 0.063532 .
## pctWInvInc            -0.097099   0.028352  -3.425 0.000630 ***
## MalePctNevMarr         0.101380   0.042801   2.369 0.017961 *
## PctKids2Par           -0.165371   0.037210  -4.444 9.37e-06 ***
## PctEmplManu           -0.036104   0.013330  -2.708 0.006825 **
## PctEmploy              0.104006   0.042207   2.464 0.013828 *
## pctUrban               0.023774   0.004968   4.785 1.85e-06 ***
## MedRentPctHousInc      0.029411   0.016696   1.762 0.078317 .
## NumInShelters          0.108011   0.201350   0.536 0.591728
## pctWWage              -0.074124   0.064452  -1.150 0.250275
## AsianPerCap            0.041177   0.021833   1.886 0.059459 .
## PctSameCity85         -0.002388   0.018729  -0.128 0.898537
## PctUsePubTrans        -0.007857   0.029196  -0.269 0.787879
## PctHousLess3BR         0.032866   0.022225   1.479 0.139386
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07575 on 1768 degrees of freedom
## Multiple R-squared:  0.6496, Adjusted R-squared:  0.6447
## F-statistic: 131.1 on 25 and 1768 DF,  p-value: < 2.2e-16
```

```
# evaluate predictions over train and test set
predictions.lm.train <- predict(mod.out, communities.lm)
predictions.lm.test <- predict(mod.out, communities.lm.test)
lm.train <- rmse(ViolentCrimesPerPop, predictions.lm.train)
lm.test <- rmse(communities.mod.test$ViolentCrimesPerPop, predictions.lm.test)
lm.train
```

```
## [1] 0.07519941
```

```
lm.test
```

```
## [1] 0.07435598
```

```
# comparison between these last two models
comparison <- data.frame(rbind(cbind(ridge.train, ridge.test), cbind(lm.train,
    lm.test)))
colnames(comparison) <- c("Ridge Regression", "LM with selected variables")
rownames(comparison) <- c("Train", "Test")
comparison
```

```
##       Ridge Regression LM with selected variables
## Train       0.07430465                 0.07426181
## Test        0.07519941                 0.07435598
```

The results, in terms of Root Mean Squared Error, are comparable, so the feature that we selected are

actually good predictors.

6 Does ethnicity play a role?

In this section we want to inspect if ethnicity does play a role in the context of crime rate, and if some commonplace about ethnicity and crime can be debunked.

```
# correlation between african american people and violent crimes
cor(racepctblack, ViolentCrimesPerPop)
```

```
## [1] 0.6296182
```

Regression models provided in this study highlight a strong correlation between the percentage of African-american people living in a community and the number of violent crimes. Our aim is to determine if this link is actually the effect of some other risk factors with are associated with specific ethnicity groups.

6.1 Correlations

Let's see which are the most correlated variables to `racepctblack`, the percentage of African-american population in the community.

```
# most correlated features to 'racepctblack'

mask.black <- corr[, which(colnames(communities.mod) == "racepctblack")]
corr.black <- mask.black[abs(mask.black) > 0.4]  # remove 'racePctWhite' from the list
corr.black <- corr.black[names(corr.black) != "racePctWhite"]
corr.black <- corr.black[names(corr.black) != "racepctblack"]

print.data.frame(data.frame(corr.black))
```

```
##                     corr.black
## pctWInvInc          -0.4940365
## pctWPubAsst          0.4449627
## PctPopUnderPov       0.4806546
## FemalePctDiv         0.4267180
## TotalPctDiv          0.4282007
## PctFam2Par          -0.7023813
## PctKids2Par         -0.7334863
## PctYoungKids2Par    -0.6589150
## PctTeen2Par         -0.6922130
## PctKidsBornNeverMar  0.8084031
## PctHousNoPhone       0.4622684
## ViolentCrimesPerPop  0.6296182
```

Let's see which are the most correlated variables to `racePctWhite`, the percentage of Caucasian population in the community.

```
# most correlated features to 'racepctblack'

mask.white <- corr[, which(colnames(communities.mod) == "racePctWhite")]
corr.white <- mask.white[abs(mask.white) > 0.4]  # remove black, whites
corr.white <- corr.white[names(corr.white) != "racePctWhite"]
corr.white <- corr.white[names(corr.white) != "racepctblack"]
corr.white <- corr.white[names(corr.white) != "racePctHisp"]

corr.white <- data.frame(corr.white)
print(corr.white)
```

```
##                     corr.white
## pctWInvInc           0.5957524
```

```
## pctWPubAsst         -0.5871231
## PctPopUnderPov      -0.5331686
## PctLess9thGrade     -0.4545836
## PctNotHSGrad        -0.4842490
## PctUnemployed       -0.5224124
## FemalePctDiv        -0.4455735
## TotalPctDiv         -0.4059730
## PctFam2Par           0.6417969
## PctKids2Par          0.7022949
## PctYoungKids2Par     0.6020222
## PctTeen2Par          0.6150993
## PctKidsBornNeverMar -0.7975698
## PctSpeakEnglOnly     0.4015022
## PctLargHouseFam     -0.5400657
## PctLargHouseOccup   -0.4631292
## PersPerRentOccHous  -0.4685509
## PctPersOwnOccup      0.5024596
## PctPersDenseHous    -0.5954049
## PctHousNoPhone      -0.4698247
## ViolentCrimesPerPop -0.6774057
```

We select only the features that they have in common.

```
merged <- merge(corr.black, corr.white, by = 0)
colnames(merged) <- c("variables", "african-american", "caucasian")
```

We associate each variable with its correlation on `ViolentCrimesPerPop`

```
violence.corr <- c()
for (i in merged[, 1]) {
    violence.corr <- c(violence.corr, (corr[i, "ViolentCrimesPerPop"]))
}
```

And keep only the variables that presents a correlation with `ViolentCrimesPerPop` superior to 0.5.

```
merged <- cbind(merged, violence.corr)
merged <- merged[abs(merged$violence.corr) > 0.5, , drop = FALSE]
merged
```

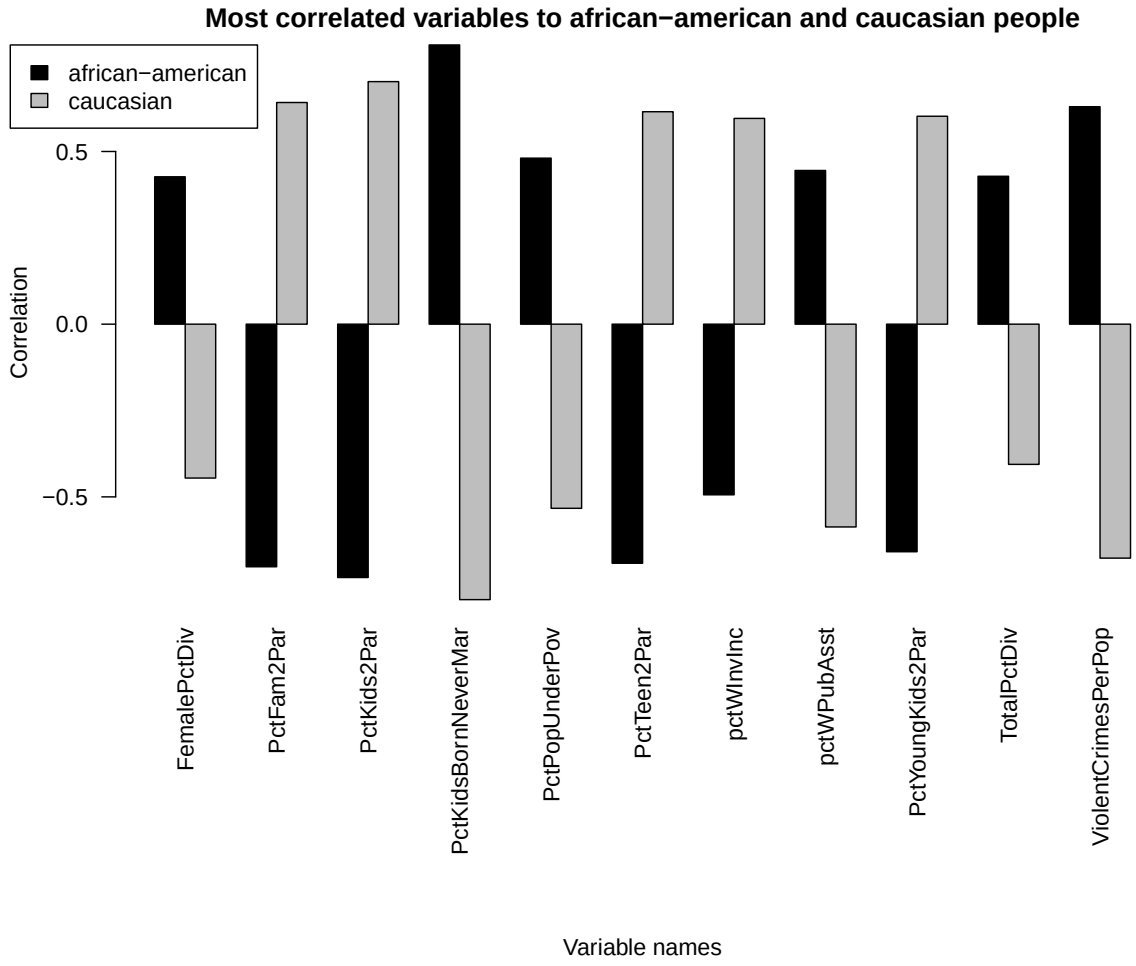
```
##              variables african-american  caucasian violence.corr
## 1         FemalePctDiv        0.4267180 -0.4455735     0.5394077
## 2           PctFam2Par       -0.7023813  0.6417969    -0.6989846
## 4          PctKids2Par       -0.7334863  0.7022949    -0.7278658
## 5  PctKidsBornNeverMar        0.8084031 -0.7975698     0.7402755
## 6       PctPopUnderPov        0.4806546 -0.5331686     0.5096517
## 7          PctTeen2Par       -0.6922130  0.6150993    -0.6559506
## 8           pctWInvInc       -0.4940365  0.5957524    -0.5625545
## 9          pctWPubAsst        0.4449627 -0.5871231     0.5683616
## 10    PctYoungKids2Par       -0.6589150  0.6020222    -0.6554837
## 11         TotalPctDiv        0.4282007 -0.4059730     0.5382447
## 12 ViolentCrimesPerPop        0.6296182 -0.6774057     1.0000000
```

The previous table shows that `racepctblack` is highly correlated with the other features which have high coefficients in our regressions, while `racePctWhite` is linked to many features which have negative coefficients.

The following barplot highlights how different are the correlations of `racepctblack` and `racePctWhite` for all these features. Actually we see that for what concerns "risk factors" for crime, African-american and Caucasian population presents a very contrasting behaviour: those features that are positively correlated

with African-american people are negatively correlated to Caucasian people, and vice-versa. This suggests that the background – in terms of family, social and economic aspects – are very different between these two ethnicity group, and thus the different behaviour with respect to crime.

```
par(mar = c(13, 5, 2, 2))
barplot(`colnames<-`(t(merged[2:3]), merged[, 1]), beside = TRUE, legend.text = TRUE,
    args.legend = list(x = "topleft", inset = c(-0.1, 0)), col = c("black",
        "grey"), main = "Most correlated variables to african-american and caucasian people",
    las = 2)

title(xlab = "Variable names", mgp = c(12, 1, 0))
title(ylab = "Correlation", mgp = c(3, 1, 0))
```



6.2 Partial correlations

Let now consider the partial correlation of these two ethnies with respect to the other features. Partial correlation measures the degree of association between two random variables, with the effect of a set of controlling random variables removed.

The partial correlation between `racepctblack` and `ViolentCrimesPerPop` is 0.078, a strong decrease when compared to the correlation value of 0.629.

```
mod1 <- lm(racepctblack ~ . - ViolentCrimesPerPop, data = communities.mod)
mod2 <- lm(ViolentCrimesPerPop ~ . - racepctblack, data = communities.mod)
```

```
cor(residuals(mod1), residuals(mod2))
```

```
## [1] 0.07830609
```

```
S <- cov(communities.mod)
part.cor <- -cov2cor(solve(S))
```

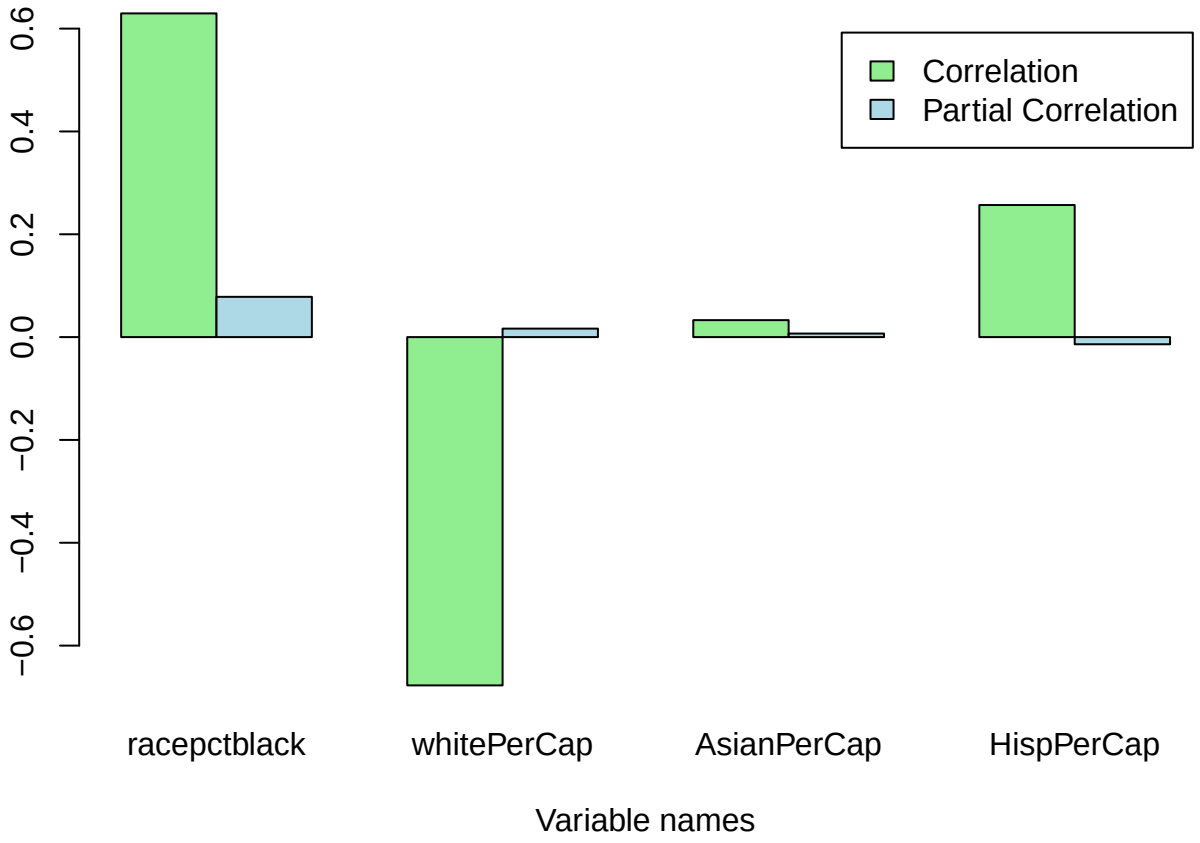
In the following barplot we compare, for each ethnic group, the correlation and partial correlation with crime rate. We see a very strong "decrease" for `racepctblack`: this further reinforces the intuition that this ethnic group is highly –negatively– influenced by its environmental background.

The partial correlation values for the four ethnic groups are actually not much different, suggesting that, filtering out the background of each group, ethnicity do not play a significant role in determining the crime rate.

```
cor(HispPerCap, ViolentCrimesPerPop)
```

```
## [1] -0.2180066
```

```
corr <- cor(communities.mod)
S <- cov(communities.mod)
part.cor <- -cov2cor(solve(S))


par(mar = c(4, 3, 1, 0))
ethicity.cor <- c()
ethicity.cor <- c(ethicity.cor, corr[3, 76])
ethicity.cor <- c(ethicity.cor, corr[4, 76])
ethicity.cor <- c(ethicity.cor, corr[5, 76])
ethicity.cor <- c(ethicity.cor, corr[6, 76])
ethicity.cor <- t(as.matrix(ethicity.cor))
rownames(ethicity.cor) <- c("Correlation")
ethicity.part.cor <- c()
ethicity.part.cor <- c(ethicity.part.cor, part.cor[3, 76])
ethicity.part.cor <- c(ethicity.part.cor, part.cor[4, 76])
ethicity.part.cor <- c(ethicity.part.cor, part.cor[5, 76])
ethicity.part.cor <- c(ethicity.part.cor, part.cor[6, 76])
ethicity.part.cor <- t(as.matrix(ethicity.part.cor))
rownames(ethicity.part.cor) <- c("Partial Correlation")
ethicity.names <- c("racepctblack", "whitePerCap", "AsianPerCap", "HispPerCap")
barplot(`colnames<-`(rbind(ethicity.cor, ethicity.part.cor), (ethicity.names)),
    beside = TRUE, col = c("lightgreen", "lightblue"), legend.text = TRUE)
title(xlab = "Variable names", mgp = c(3, 1, 0))
title(ylab = "Numeric value", mgp = c(3, 1, 0))
```


0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Correlation
Partial Correlation
racepctblack
whitePerCap
AsianPerCap
HispPerCap
Variable names

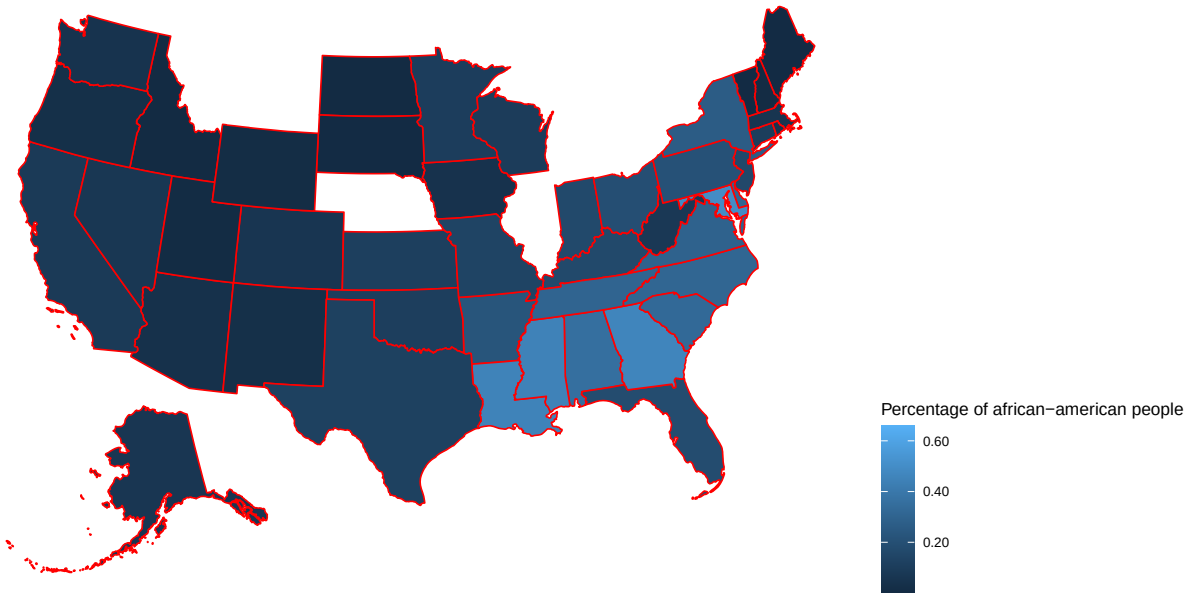
7 Data over US states

Now the focus is over states of the US territory, instead of communities. Considering areas with an elevated presence of a given ethnic group, the aim is to discover the features that are more linked to violent crimes in these zones, in particular it is of interest the comparison with the case of the whole US.

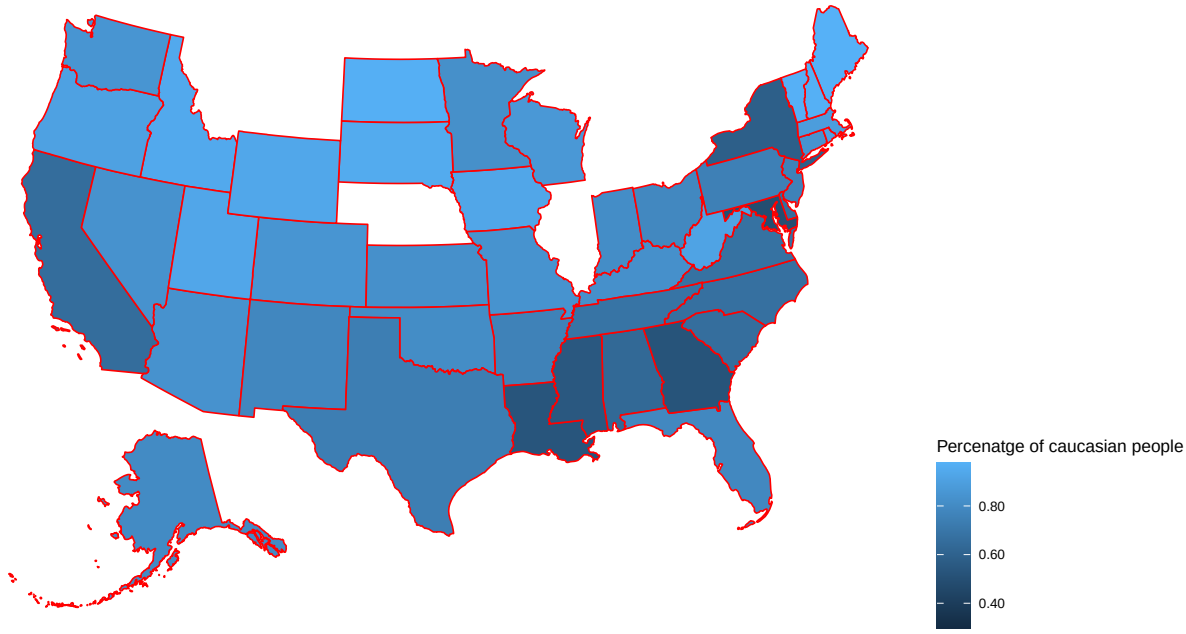
7.1 Maps plot

To visualise the distribution of data, records are grouped by state. Numerical features of every state are computed with a weighted mean over the different communities belonging to that specific territory.

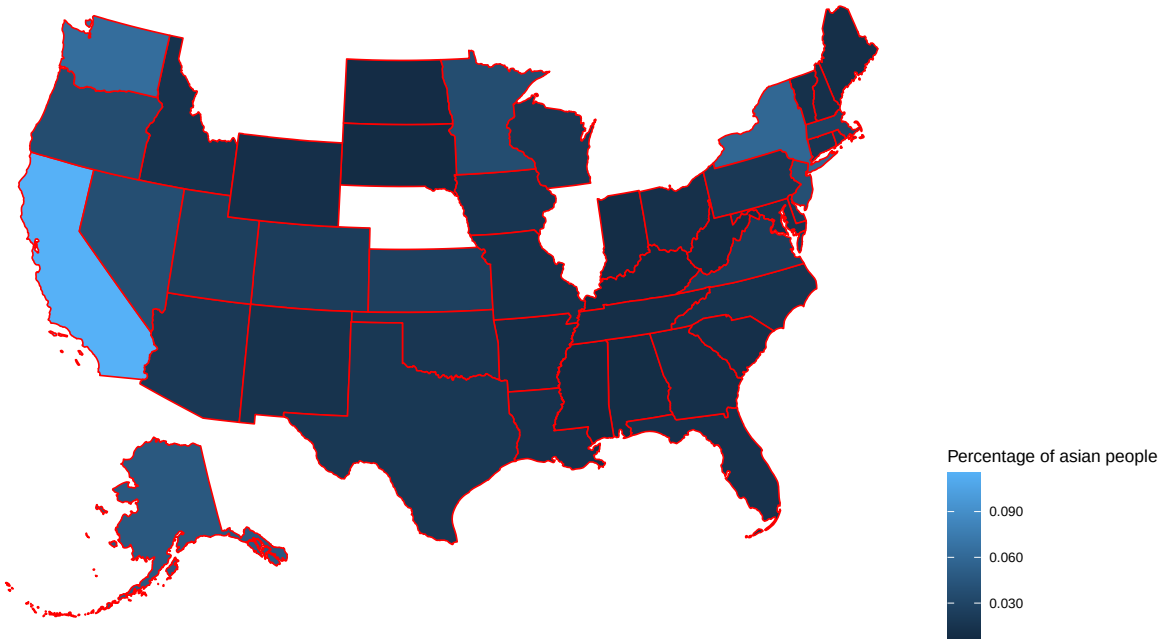
```
library(dplyr)
library(ggplot2)
library(usmap)


pre_state <- communities.tt  #dataset with 1994 rows and 80 columns

pre_state$totBlack <- pre_state$population * pre_state$racepctblack/100
pre_state$totWhite <- pre_state$population * pre_state$racePctWhite/100
pre_state$totAsian <- pre_state$population * pre_state$racePctAsian/100
pre_state$totHisp <- pre_state$population * pre_state$racePctHisp/100
pre_state$ViolentCrimesbyPop <- pre_state$population * pre_state$ViolentCrimesPerPop

pre_state2 <- aggregate(cbind(pre_state$population, pre_state$totBlack,
    pre_state$totWhite, pre_state$totAsian, pre_state$totHisp, pre_state$ViolentCrimesPerPop),
    by = list(Category = pre_state$state), FUN = sum)  #dataset grouped by state

colnames(pre_state2) <- c("state", "totPop", "totBlack", "totWhite", "totAsian",
    "totHisp", "totViolentCrimesPerState")

pre_state2$totBlack_pct <- pre_state2$totBlack/pre_state2$totPop
pre_state2$totWhite_pct <- pre_state2$totWhite/pre_state2$totPop
pre_state2$totAsian_pct <- pre_state2$totAsian/pre_state2$totPop
pre_state2$totHisp_pct <- pre_state2$totHisp/pre_state2$totPop
pre_state2$totViolentCrimes <- pre_state2$totViolentCrimesPerState/pre_state2$totPop

state_vec = c("AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "DC", "FL",
    "GA", "ID", "IN", "IA", "KS", "KY", "LA", "ME", "MD", "MA", "MN", "MS",
    "MO", "NV", "NH", "NJ", "NM", "NC", "NY", "NC", "ND", "OH", "OK", "OR",
    "PA", "RI", "SC", "SD", "TN", "TX", "UT", "VT", "VA", "WV", "WI", "WY",
    "WA")

plot_usmap(data = pre_state2, values = "totBlack_pct", include = state_vec,
    color = "red") + scale_fill_continuous(name = "Percentage of african-american people",
    label = scales::comma) + theme(legend.position = "right")
```



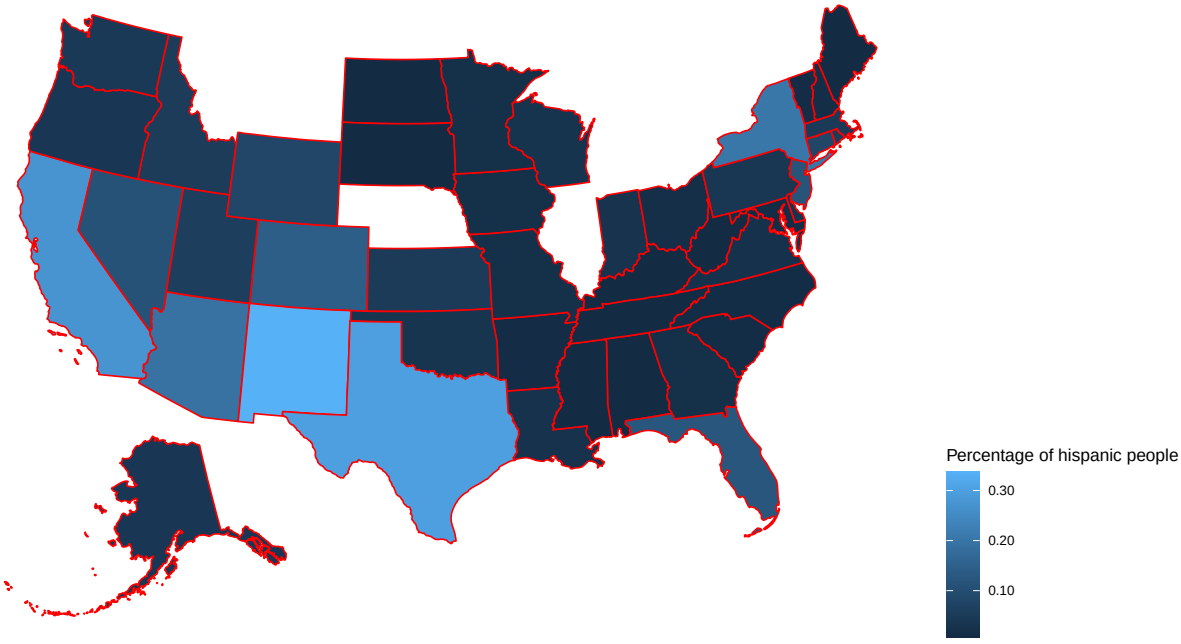

```
plot_usmap(data = pre_state2, values = "totWhite_pct", include = state_vec,
    color = "red") + scale_fill_continuous(name = "Percenatge of caucasian people",
    label = scales::comma) + theme(legend.position = "right")
```



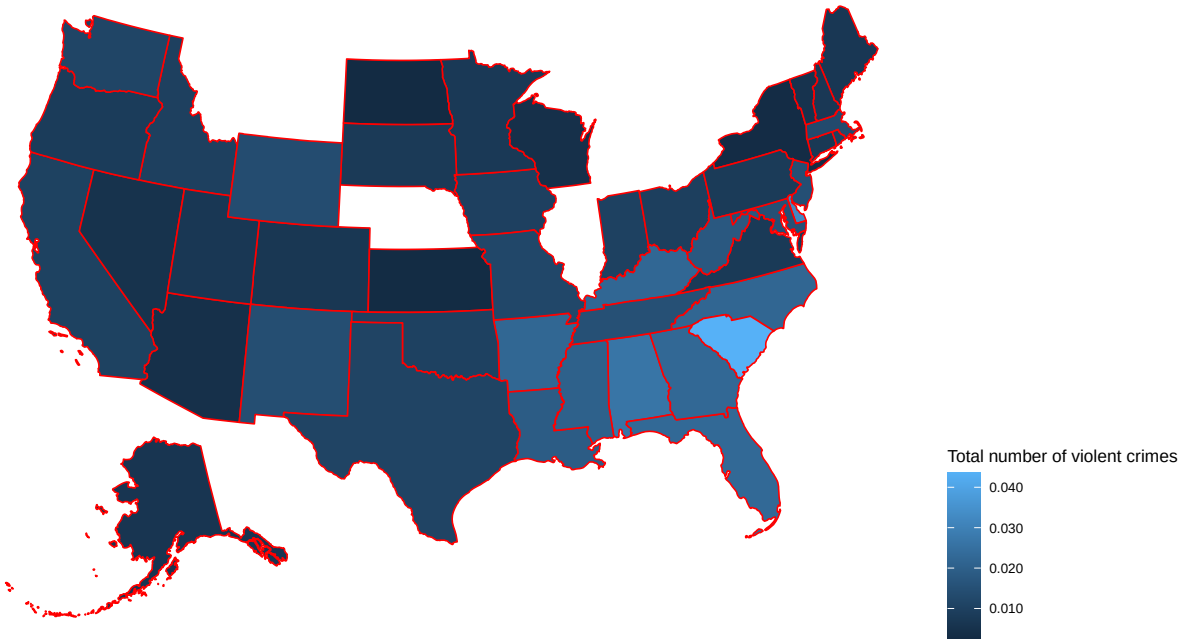

```
plot_usmap(data = pre_state2, values = "totAsian_pct", include = state_vec,
    color = "red") + scale_fill_continuous(name = "Percentage of asian people",
    label = scales::comma) + theme(legend.position = "right")
```




```
plot_usmap(data = pre_state2, values = "totHisp_pct", include = state_vec,
    color = "red") + scale_fill_continuous(name = "Percentage of hispanic people",
    label = scales::comma) + theme(legend.position = "right")
```




```
plot_usmap(data = pre_state2, values = "totViolentCrimes", include = state_vec,
    color = "red") + scale_fill_continuous(name = "Total number of violent crimes",
    label = scales::comma) + theme(legend.position = "right")
```



By the previous maps, it is possible to highlight a south-west area which has a significant presence of hispanic people, while a south-east area has a high percentage of African-american people.

The south-west area considered is formed by: California, Nevada, New Mexico, Arizona and Texas.

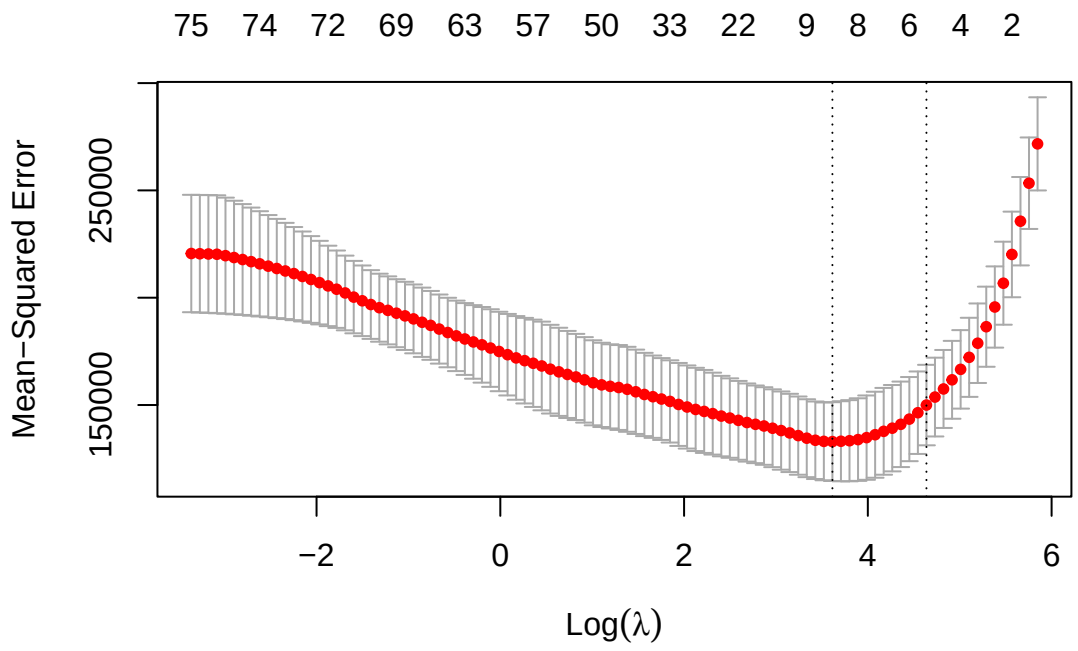
The south-east area considered comprehends: Louisiana, Mississippi, Alabama, Georgia, South Carolina, North Carolina and Tennessee.

```
drops2 <- c("communityname", "OtherPerCap")
communities.tt <- communities.tt[, !(names(communities.tt) %in% drops2)]

communities.SW <- communities.tt[which(communities.tt$state == "CA" | communities.tt$state ==
    "TX" | communities.tt$state == "NM" | communities.tt$state == "AZ" |
    communities.tt$state == "NV"), ]
communities.SW_violent <- communities.SW["ViolentCrimesPerPop"]

communities.SE <- communities.tt[which(communities.tt$state == "LA" | communities.tt$state ==
    "MS" | communities.tt$state == "AL" | communities.tt$state == "GE" |
    communities.tt$state == "SC", communities.tt$state == "NC", communities.tt$state ==
    "TN"), ]

communities.SE_violent <- communities.SE["ViolentCrimesPerPop"]

drops3 <- c("state", "ViolentCrimesPerPop")
communities.SW <- communities.SW[, !(names(communities.SW) %in% drops3)]
communities.SE <- communities.SE[, !(names(communities.SE) %in% drops3)]

ppSW = preProcess(communities.SW, method = "range")
ppSE = preProcess(communities.SE, method = "range")

communities.SW <- predict(ppSW, communities.SW)
communities.SE <- predict(ppSE, communities.SE)
```

7.2 LASSO regression

In order to detect which features are more linked to violent crimes in this areas, LASSO regressions are provided. The most important features are chosen according to the absolute value of the coefficients.

```
X <- communities.SW
y <- communities.SW_violent$ViolentCrimesPerPop

set.seed(1)

# train and test split

train <- sample(1:nrow(X), nrow(X)/2)
test <- (-train)
y.test <- y[test]

X = as.matrix(X)

# cross validation for values of Lasso
cv.lassoSW <- cv.glmnet(X[train, ], y[train], alpha = 1, nfold = 5)

plot(cv.lassoSW)
```

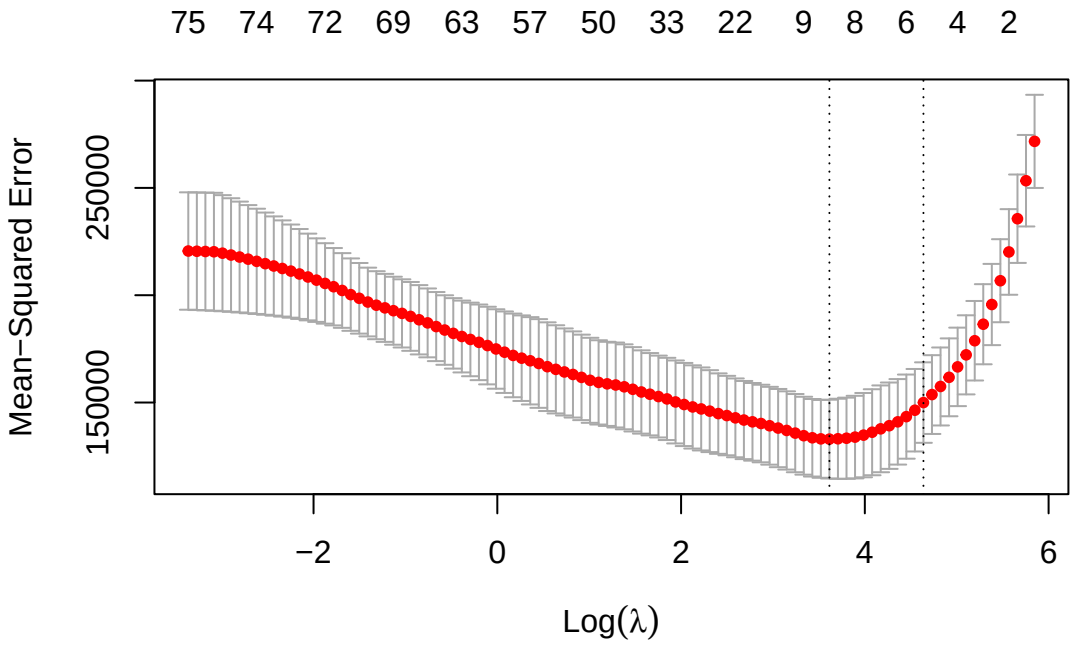



```
# label best parameter
i.bestlam <- which.min(cv.lassoSW$cvm)

bestlam <- cv.lassoSW$lambda[i.bestlam]

# train new LASSO over the best parameter
lassoSW <- glmnet(X, y, alpha = 1)
```

```r
pred <- predict(lassoSW, type = "coefficients", s = bestlam)

best.lasso <- coef(lassoSW, s = bestlam)
# best.lasso <- predict(lasso.mod, s=10, type='coefficients')
best.lasso <- data.frame(as.matrix(best.lasso))
best.lasso <- best.lasso[best.lasso$s1 != 0, , drop = FALSE]

# pick the 10 mostly linked features to crime
head(arrange(best.lasso, desc(abs(s1))), n = 15)  #return features linked to highest values
```

```
##                             s1
## (Intercept)         1099.81500
## PctKids2Par         -806.98893
## PctKidsBornNeverMar  804.75393
## NumInShelters        346.24517
## racepctblack         327.82325
## racePctWhite        -216.53329
## PopDens              194.23241
## agePct16t24         -139.13249
## pctWFarmSelf        -137.73299
## NumStreet            105.70260
## PctWorkMom           -37.71710
## TotalPctDiv           26.04256
```

```r
X <- communities.SE
y <- communities.SE_violent$ViolentCrimesPerPop

set.seed(1)

# train and test split
train <- sample(1:nrow(X), nrow(X)/7 * 6)
test <- (-train)
y.test <- y[test]

X = as.matrix(X)

# cross validation for values of Lasso
cv.lassoSE <- cv.glmnet(X[train, ], y[train], alpha = 1, nfold = 5)

plot(cv.lassoSW)
```




```
i.bestlam <- which.min(cv.lassoSE$cvm)

bestlam <- cv.lassoSE$lambda[i.bestlam]

# train new LASSO over the best parameter
lassoSE <- glmnet(X, y, alpha = 1)
pred <- predict(lassoSE, type = "coefficients", s = bestlam)

best.lasso <- coef(lassoSE, s = bestlam)
# best.lasso <- predict(lasso.mod, s=10, type='coefficients')
best.lasso <- data.frame(as.matrix(best.lasso))
best.lasso <- best.lasso[best.lasso$s1 != 0, , drop = FALSE]

# pick the 10 mostly linked features to crime
head(arrange(best.lasso, desc(abs(s1))), n = 10)  #return features linked to highest values
```

```
##                          s1
## (Intercept)      1984.179491
## PctKids2Par     -1145.550614
## racePctWhite     -647.153830
## PctUsePubTrans    324.178472
## pctUrban          141.586791
## pctWFarmSelf      -76.849164
## MalePctDivorce     47.275075
## PctHousNoPhone     25.565523
## PctBSorMore       -14.928211
## PctPersOwnOccup    -3.497896
```

Comparing the most influent features of these two zones with the ones regarding the whole dataset, confirms that the aspects more linked to violent crimes regard families. More in detail, the rate of kids with both

parents is, in these areas, the most important feature, highly negatively correlated to violent crimes. In south-east states there is also a strong contribute from features regarding the percentage of people who use public transport, which usually is linked to poverty, and the urbanisation, given that these states are highly densely populated.

8 Conclusion

Through this paper, we have studied the influence of several factors, belonging to different categories, over the violent crimes of US communities. From these analysis it can be stated that the most important features regard family conditions, with an high remark over the situation of kids.

Another important chapter is dedicated to explain the high correlation between violent crimes and the percentage of African-american people, which is due to bad socio-economic conditions this ethnic group experiences.

The South-Easter and South-Western zones, which have a considerable percentage of, –respectively– Hispanic people and African-american people, share the set of important features found for the whole US case.

Appendix: Attribute Information

We report the meaning of each variable as described in the source of the dataset.

- `state`: US state (by number) - not counted as predictive above, but if considered, should be consided nominal (nominal)
- `communityname`: community name - not predictive - for information only (string)
- `householdsize`: mean people per household (numeric - decimal)

Population

- `population`: population for community (numeric - decimal)
- `PopDens`: population density in persons per square mile (numeric - decimal)
- `PctUsePubTrans`: percent of people using public transit for commuting (numeric - decimal)
- `pctUrban`: percentage of people living in areas classified as urban (numeric - decimal)
- `PctSameCity85`: percent of people living in the same city as in 1985 (5 years before) (numeric - decimal)
- `PctSameState85`: percent of people living in the same state as in 1985 (5 years before) (numeric - decimal)

Ethnicity

- `racepctblack`: percentage of population that is african american (numeric - decimal)
- `racePctWhite`: percentage of population that is caucasian (numeric - decimal)
- `racePctAsian`: percentage of population that is of asian heritage (numeric - decimal)
- `racePctHisp`: percentage of population that is of hispanic heritage (numeric - decimal)

Age

- `agePct12t21`: percentage of population that is 12-21 in age (numeric - decimal)
- `agePct12t29`: percentage of population that is 12-29 in age (numeric - decimal)
- `agePct16t24`: percentage of population that is 16-24 in age (numeric - decimal)
- `pctUrban`: percentage of people living in areas classified as urban (numeric - decimal)

Income

- `medIncome`: median household income (numeric - decimal)
- `pctWWage`: percentage of households with wage or salary income in 1989 (numeric - decimal)
- `pctWFarmSelf`: percentage of households with farm or self employment income in 1989 (numeric - decimal)
- `pctWInvInc`: percentage of households with investment / rent income in 1989 (numeric - decimal)
- `pctWSocSec`: percentage of households with social security income in 1989 (numeric - decimal)
- `pctWPubAsst`: percentage of households with public assistance income in 1989 (numeric - decimal)
- `pctWRetire`: percentage of households with retirement income in 1989 (numeric - decimal)
- `whitePerCap`: per capita income for caucasians (numeric - decimal)
- `blackPerCap`: per capita income for african americans (numeric - decimal)
- `indianPerCap`: per capita income for native americans (numeric - decimal)
- `AsianPerCap`: per capita income for people with asian heritage (numeric - decimal)
- `OtherPerCap`: per capita income for people with 'other' heritage (numeric - decimal)
- `HispPerCap`: per capita income for people with hispanic heritage (numeric - decimal)
- `PctPopUnderPov`: percentage of people under the poverty level (numeric - decimal)

Education

- `PctLess9thGrade`: percentage of people 25 and over with less than a 9th grade education (numeric - decimal)
- `PctNotHSGrad`: percentage of people 25 and over that are not high school graduates (numeric - decimal)
- `PctBSorMore`: percentage of people 25 and over with a bachelors degree or higher education (numeric - decimal)

Employment

- PctUnemployed: percentage of people 16 and over, in the labor force, and unemployed (numeric - decimal)
- PctEmploy: percentage of people 16 and over who are employed (numeric - decimal)
- PctEmplManu: percentage of people 16 and over who are employed in manufacturing (numeric - decimal)
- PctEmplProfServ: percentage of people 16 and over who are employed in professional services (numeric - decimal)
- PctOccupManu: percentage of people 16 and over who are employed in manufacturing (numeric - decimal)
- PctOccupMgmtProf: percentage of people 16 and over who are employed in management or professional occupations (numeric - decimal)

Family

- MalePctDivorce: percentage of males who are divorced (numeric - decimal)
- MalePctNevMarr: percentage of males who have never married (numeric - decimal)
- FemalePctDiv: percentage of females who are divorced (numeric - decimal)
- TotalPctDiv: percentage of population who are divorced (numeric - decimal)
- PctFam2Par: percentage of families (with kids) that are headed by two parents (numeric - decimal)

Children

- PctKids2Par: percentage of kids in family housing with two parents (numeric - decimal)
- PctYoungKids2Par: percent of kids 4 and under in two parent households (numeric - decimal)
- PctTeen2Par: percent of kids age 12-17 in two parent households (numeric - decimal)
- PctWorkMomYoungKids: percentage of moms of kids 6 and under in labor force (numeric - decimal)
- PctWorkMom: percentage of moms of kids under 18 in labor force (numeric - decimal)
- NumKidsBornNeverMar: number of kids born to never married (numeric - expected to be integer) (called NumIlleg in normalized)
- PctKidsBornNeverMar: percentage of kids born to never married (numeric - decimal) (called PctIlleg in normalized)

Immigrants

- NumImmig: total number of people known to be foreign born (numeric - decimal)
- PctRecentImmig: percent of *population* who have immigrated within the last 3 years (numeric - decimal)
- PctSpeakEnglOnly: percent of people who speak only English (numeric - decimal)

Households

- PctLargHouseFam: percent of family households that are large (6 or more) (numeric - decimal)
- PctLargHouseOccup: percent of all occupied households that are large (6 or more people) (numeric - decimal)
- PersPerOccupHous: mean persons per household (numeric - decimal)
- PersPerOwnOccHous: mean persons per owner occupied household (numeric - decimal)
- PersPerRentOccHous: mean persons per rental household (numeric - decimal)
- PctPersOwnOccup: percent of people in owner occupied households (numeric - decimal)
- PctPersDenseHous: percent of persons in dense housing (more than 1 person per room) (numeric - decimal)
- PctHousLess3BR: percent of housing units with less than 3 bedrooms (numeric - decimal)
- PctHousOccup: percent of housing occupied (numeric - decimal)
- PctHousNoPhone: percent of occupied housing units without phone (in 1990, this was rare!) (numeric - decimal)
- PctWOFullPlumb: percent of housing without complete plumbing facilities (numeric - decimal)
- OwnOccLowQuart: owner occupied housing - lower quartile value (numeric - decimal)
- OwnOccMedVal: owner occupied housing - median value (numeric - decimal)
- OwnOccHiQuart: owner occupied housing - upper quartile value (numeric - decimal)
- RentLowQ: rental housing - lower quartile rent (numeric - decimal)

- RentMedian: rental housing - median rent (Census variable H32B from file STF1A) (numeric - decimal)
- RentHighQ: rental housing - upper quartile rent (numeric - decimal)
- MedRent: median gross rent (Census variable H43A from file STF3A - includes utilities) (numeric - decimal)
- MedRentPctHousInc: median gross rent as a percentage of household income (numeric - decimal)
- MedOwnCostPctInc: median owners cost as a percentage of household income - for owners with a mortgage (numeric - decimal)
- MedOwnCostPctIncNoMtg: median owners cost as a percentage of household income - for owners without a mortgage (numeric - decimal)

Homeless

- NumInShelters: number of people in homeless shelters (numeric - decimal)
- NumStreet: number of homeless people counted in the street (numeric - decimal)
- PctBornSameState: percent of people born in the same state as currently living (numeric - decimal)
- PctSameHouse85: percent of people living in the same house as in 1985 (5 years before) (numeric - decimal)
- PctSameCity85: percent of people living in the same city as in 1985 (5 years before) (numeric - decimal)
- PctSameState85: percent of people living in the same state as in 1985 (5 years before) (numeric - decimal)

Police

- LemasSwFTPerPop: sworn full time police officers per 100K population (numeric - decimal)
- LemasSwFTFieldPerPop: sworn full time police officers in field operations (on the street as opposed to administrative etc) per 100K population (numeric - decimal)
- LemasTotReqPerPop: total requests for police per 100K popuation (numeric - decimal)
- PolicCars: number of police cars (numeric - decimal)
- PolicBudgPerPop: police operating budget per population (numeric - decimal)

Police and ethnicity

- RacialMatchCommPol: a measure of the racial match between the community and the police force. High values indicate proportions in community and police force are similar (numeric - decimal)
- PctPolicWhite: percent of police that are caucasian (numeric - decimal)
- PctPolicBlack: percent of police that are african american (numeric - decimal)
- PctPolicHisp: percent of police that are hispanic (numeric - decimal)
- PctPolicAsian: percent of police that are asian (numeric - decimal)
- PctPolicMinor: percent of police that are minority of any kind (numeric - decimal)

Police for drugs

- OfficAssgnDrugUnits: number of officers assigned to special drug units (numeric - decimal)
- PolicAveOTWorked: police average overtime worked (numeric - decimal)

Population density

- PopDens: population density in persons per square mile (numeric - decimal)
- PctUsePubTrans: percent of people using public transit for commuting (numeric - decimal)

Target variables

- ViolentCrimesPerPop: total number of violent crimes per 100K popuation (numeric - decimal)