Analysis of Continuous Data project

Thomas Sertijn, Bart Smets, Ilja Van Bever, Lieselot Van de Putte

2025-11-09

Protocol - Univariate part

Research question

During this research, we want to investigate how socio-economic disadvantage relates to violent crime rates. More specifically we want to explore the association between poverty and violent crime rates in the USA.

In his seminal work, Becker (1968) stated that the decision to commit crime is a rational choice where people weigh the benefits and costs against each other. It could then be argued that the incentive to commit crime is higher for people who have a lower income, as the benefits are larger for this group. Following this, we would then also expect that in communities with a higher poverty rate, there will also be higher crime rates. Depending on the results of our analysis, these results could be used to inform relevant policies. It would, for example, give another argument for the implementation of redistributive policies: if an effect is found, policymakers should take this reduction in violent crime into account, next to an economic benefit. Our analysis hopes to shed further light on this issue.

For the purpose of our research question, the following predictor variables have been selected:

- **PctPopUnderPov**: percentage of people under the poverty level (main predictor).
- **perCapInc**: per capita income. While similar to pctunderpoverty, this takes the whole income distribution into account and not just the lower end. If this average is lower, then we expect more crime to happen.
- **PctEmploy**: percentage of people 16 and over who are employed. We could argue that if more people are employed less people have an incentive to commit crime.
- **PctLess9thGrade**: percentage of people 25 and over with less than a 9th grade education. Education leads to a higher socio-economic standing, which would suggest that people have less reason to commit crime. We choose this variable for now, but as an alternative we could later use one of the following two variables if we would find them better suited as predictors: **PctNotHSGrad** (percentage of people 25 or over, that have not graduated highschool) or **PctBSorMore** (percentage of people 25 or over, with at least a bachelor's degree).
- **NumImmig**: total number of people known to be foreign born. Immigrants commiting more crimes is a commonly used right-wing argument against migration, and relevant as immigrants are often from a 'lower' socio-economic background.
- **racepctblac**: percentage of population that is african american. It is a common right wing argument as well that black people commit crime, because they are from a 'lower' socio-economic background.
- **agePct12t29**: percentage of population that is 12-29 in age. We include this because young people have had less time to build up their socio-economic status, as well as their brain being less developed, and might thus commit more crime.

Design of the study

Descriptive analysis

To get a first impression of the data, a descriptive analysis will be performed for the candidate predictor variables (all continuous). The datasets are checked for missing values. The most common univariate statistics are calculated: the mean, the standard deviation, the minimum, the first quartile, the median, the third quartile and the maximum.

The distributions of the variables are visualized by boxplots, QQ plots and histograms. Outliers are identified using Tukey's 1.5 x IQR rule. For the univariate descriptive statistics also the population size of the communities is considered. The population size can influence the reliability of the data points: small communities can have a higher probability to have more extreme values of the predictor and response variables by the fact that the denominator in the response variable (total number of violent crimes per 100K population) is smaller. In the regression phase this will be used to investigate the outlier values.

To find what the relationship is between the main predictor variable and the potential extra predictor variables, scatter plots with smoothers are made for the bivariate relationships and correlations are checked.

Linear regression

Before performing linear regression and building models, the dataset is split into a training set (80% of the data) and a test set (20% of the data).

To investigate the association between the main predictor variable and the response variable, a linear regression is fitted and the output is evaluated. The various statistics are calculated and discussed: estimate regression coefficients, the F-statistics (/t-statistics), the R squared, the MSE, the p-value, the confidence interval and standard error of the slope. We first present the general formula here, before we fill in the specific variables.

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

$$ViolentCrimesPerPop_i = \beta_0 + \beta_1 pctpopUnderPov_i + \epsilon_i$$

Confidence intervals are constructed. Based on this, outliers can be identified. Subsequently, the outliers are further evaluated, e.g. are outliers linked to communities with a small population size.

Assumption checks

For linear regression, multiple assumptions, such as linearity, independence of errors, homoscedasticity (constant variance of errors), and normality of errors, are made. During this research these assumptions have to be checked by: Plotting residuals vs. fitted values for the linearity and independence of errors, squared residuals vs. fitted values for homoscedasticity checks, normality checks by qq-plot of the residuals. To also take leverage into account, the studentized residuals will be plotted.

Protocol - Multivariate part

Model building

Forward stepwise regression Evaluation of adjusted R-squared, AIC, SBC Partial regression plots? In which functional form we let a variable enter the model?

Model fit and outliers

PRESS, studentized residual plots (transformations needed?), bijv. QQ-plots to predicted value of y/log(y), Also DFFITS, Cook's Distance, DFBETAS -> welke outliers hebben een grote invloed? Deleted residuals?

Interpretation of the parameters

Table 1: Project Schedule Overview

Deadline	Subject	Final_responsibility
3/11	Data extraction	Thomas
10/11	Descriptive analyses	Ilja
17/11	Model building	Bart
24/11	Model interpretation	Lieselot
24/11	Prediction with linear model	Ilja
1/12	Statistical discussion linear model	Bart
1/12	Fitting GLM	Lieselot
1/12	Fitting the final model	Thomas
8/12	Prediction with GLM	Ilja
8/12	Statistical discussion GLM	Thomas
8/12	Final conclusion and discussion	Lieselot

Data extraction:

```
library(readr)
library(dplyr)
library(stringr)

# The .arff header is usually inside the .names file:
url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/communities/communities.names"

# Read the text lines
lines <- read_lines(url)
start <- grep("Additional Variable Information", lines, ignore.case = TRUE)
end   <- grep("Summary Statistics:", lines, ignore.case = TRUE)

var_lines <- lines[str_starts(str_trim(lines), "--")]
var_names <- str_extract(var_lines, "(?<=-- )[^:]+")
var_names <- c("communityname", setdiff(var_names, "communityname"))
```

```
variable_names <- list("communityname","state","countyCode", "communityCode", "fold", "population", "hou
                       "racepctblack", "racePctWhite", "racePctAsian", "racePctHisp", "agePct12t21", "ag
                       "agePct65up", "numbUrban", "pctUrban", "medIncome", "pctWWage", "pctWFarmSelf",
                       "pctWPubAsst", "pctWRetire", "medFamInc", "perCapInc", "whitePerCap", "blackPerCa
                       "OtherPerCap", "HispPerCap", "NumUnderPov", "PctPopUnderPov", "PctLess9thGrade",
                       "PctUnemployed", "PctEmploy", "PctEmplManu", "PctEmplProfServ", "PctOccupManu",
                       "MalePctNevMarr", "FemalePctDiv", "TotalPctDiv", "PersPerFam", "PctFam2Par", "Pct
                       "PctTeen2Par", "PctWorkMomYoungKids", "PctWorkMom", "NumKidsBornNeverMar", "PctKi
                       "PctImmigRecent", "PctImmigRec5", "PctImmigRec8", "PctImmigRec10", "PctRecentImmi
```

```
                              "PctRecImmig10", "PctSpeakEnglOnly", "PctNotSpeakEnglWell", "PctLargHouseFam", "
                              "PersPerOwnOccHous", "PersPerRentOccHous", "PctPersOwnOccup", "PctPersDenseHous"
                              "HousVacant", "PctHousOccup", "PctHousOwnOcc", "PctVacantBoarded", "PctVacMore6Mo
                              "PctHousNoPhone", "PctWOFullPlumb", "OwnOccLowQuart", "OwnOccMedVal", "OwnOccHiQu
                              "RentMedian", "RentHighQ", "RentQrange", "MedRent", "MedRentPctHousInc", "MedOwnC
                              "NumInShelters", "NumStreet", "PctForeignBorn", "PctBornSameState", "PctSameHouse
                              "PctSameState85", "LemasSwornFT", "LemasSwFTPerPop", "LemasSwFTFieldOps", "LemasS
                              "LemasTotReqPerPop", "PolicReqPerOffic", "PolicPerPop", "RacialMatchCommPol", "Pc
                              "PctPolicHisp", "PctPolicAsian", "PctPolicMinor", "OfficAssgnDrugUnits", "NumKind
                              "LandArea", "PopDens", "PctUsePubTrans", "PolicCars", "PolicOperBudg", "LemasPctP
                              "LemasPctOfficDrugUn", "PolicBudgPerPop", "murders", "murdPerPop", "rapes", "rape
                              "assaults", "assaultPerPop", "burglaries", "burglPerPop", "larcenies", "larcPerPo
                              "arsons", "arsonsPerPop", "ViolentCrimesPerPop", "nonViolPerPop")
```

```
library(data.table)
violent_crimes_table <- fread("curl https://archive.ics.uci.edu/static/public/211/communities+and+crime+
```

```
colnames(violent_crimes_table) <- unlist(variable_names)
```

```
crimes_table_subset <- violent_crimes_table %>%
  select(communityname,state,countyCode, communityCode, fold, population,
         PctPopUnderPov, perCapInc, PctEmploy, PctLess9thGrade, PctNotHSGrad, PctBSorMore,
         NumImmig, racepctblack, agePct12t29, ViolentCrimesPerPop
         )
```

Design (tekst gekopieerd van website)

The source datasets needed to be combined via programming. Many variables are included so that algorithms that select or learn weights for attributes could be tested. However, clearly unrelated attributes were not included; attributes were picked if there was any plausible connection to crime (N=125), plus the crime variables which are potential dependent variables. The variables included in the dataset involve the community, such as the percent of the population considered urban, and the median family income, and involving law enforcement, such as per capita number of police officers, and percent of officers assigned to drug units. The crime attributes (N=18) that could be predicted are the 8 crimes considered 'Index Crimes' by the FBI)(Murders, Rape, Robbery, ...), per capita (actually per 100,000 population) versions of each, and Per Capita Violent Crimes and Per Capita Nonviolent Crimes).

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 datasets were omitted. Many communities are missing LEMAS data.

The per capita crimes variables were calculated using population values included in the 1995 FBI data (which differ from the 1990 Census values).

The per capita violent crimes variable was calculated using population and the sum of crime variables considered violent crimes in the United States: murder, rape, robbery, and assault. There was apparently some controversy in some states concerning the counting of rapes. These resulted in missing values for rape, which resulted in missing values for per capita violent crime. Many of these omitted communities were from the midwestern USA (Minnesota, Illinois, and Michigan have many of these).

The per capita nonviolent crime variable was calculated using the sum of crime variables considered nonviolent crimes in the United States: burglaries, larcenies, auto thefts and arsons. (There are many other types of crimes, these only include FBI 'Index Crimes')

Some further pre-processing of the dataset must be done. Choose the desirable dependent variable from among the 18 possible. It would not be interesting or appropriate to predict total crime (e.g. violent crime) while including subtotals (e.g. murders) as independent variables. There are also identifying variables (community name, county code, community code) that are not predictive, and would get in the way of some algorithms. Weka's Unsupervised Attribute Remove Filter can be used to remove unwanted attributes.

The FBI notes that use of this data to evaluate communities is over-simplistic, as many relevant factors are not included. For one example, communities with large numbers of visitors will have higher per capita crime (measured by residents) than communities with fewer visitors, other things being equal.

Data preparation and descriptive analysis

Since the outcome variable *ViolentCrimesPerPop* (total number of violent crimes per 100K population) is expressed relative to the population size, the variable *NumImmig* is converted (by dividing it by the population size and multiplying by 100%).

```
crimes_table_subset$ViolentCrimesPerPop <- as.numeric(crimes_table_subset$ViolentCrimesPerPop)
crimes_table_subset$PctImmig <- crimes_table_subset$NumImmig/crimes_table_subset$population*100
crimes_table_subset = crimes_table_subset[,-c('NumImmig', 'fold')]
```

```
sjlabelled::set_label(crimes_table_subset) <- c("communityname", "state", "countyCode", "communityCode"
or over, that have not graduated highschool (%)", "percentage of people 25 or over, with
at least a bachelor's degree (%)", "percentage of population that is african american (%)", "percentage
```

It is examined how many NA values are present in the database.

```
crimes_table_subset %>%
  pivot_longer(cols = where(is.numeric), names_to = "variable", values_to = "value") %>%
  group_by(variable) %>%
  summarise(
    NAs = sum(is.na(value))
  )
```

```
## # A tibble: 11 x 2
##    variable              NAs
##    <chr>               <int>
##  1 PctBSorMore             0
##  2 PctEmploy               0
##  3 PctImmig                0
##  4 PctLess9thGrade         0
##  5 PctNotHSGrad            0
##  6 PctPopUnderPov          0
##  7 ViolentCrimesPerPop   221
##  8 agePct12t29             0
##  9 perCapInc               0
## 10 population              0
## 11 racepctblack            0
```

```
na_subset <- crimes_table_subset %>%
  filter(is.na(ViolentCrimesPerPop)
  )
na_subset <- na_subset[,-'ViolentCrimesPerPop']
na_subset
```

```
##        communityname  state countyCode communityCode population PctPopUnderPov
##               <char> <char>     <char>        <char>      <int>          <num>
##   1:     Bemidjicity     MN          7          5068      11245          29.99
##   2:      NewUlmcity     MN         15         46042      13132           6.84
##   3:   Maplewoodcity     MN        123         40382      30954           6.22
##   4:    Plymouthcity     MN         53         51730      50889           3.36
##   5:     Pontiaccity     MI        125         65440      71166          26.67
##  ---
## 217:     Bristoltown     CT          3          8490      60640           4.35
## 218: Wilmettevillage     IL          ?             ?      26690           2.14
## 219: EastLansingcity     MI         65         24120      50677          33.77
## 220: CrystalLakecity     IL          ?             ?      24512           2.15
## 221:      Burtoncity     MI         49         12060      27617          14.28
##      perCapInc PctEmploy PctLess9thGrade PctNotHSGrad PctBSorMore racepctblack
##          <int>     <num>           <num>        <num>       <num>        <num>
##   1:      8483     52.44           12.15        23.06       25.28         0.53
##   2:     11907     65.62           16.28        25.41       15.31         0.06
##   3:     16459     68.12            4.40        14.64       20.28         2.52
##   4:     21908     78.05            1.57         5.56       41.39         1.61
##   5:      9847     51.07           12.04        37.61        7.95        42.20
##  ---
## 217:     16909     68.24           10.04        24.97       15.39         2.08
## 218:     38465     63.90            2.78         4.88       63.69         0.49
## 219:     11212     57.45            0.93         3.38       71.23         6.93
## 220:     17681     71.89            3.53        11.00       28.50         0.20
## 221:     12940     53.67            7.56        27.32        6.68         2.57
##      agePct12t29   PctImmig
##            <num>      <num>
##   1:       40.53  1.7429969
##   2:       25.03  0.9061834
##   3:       25.43  2.6232474
##   4:       26.94  2.6135314
##   5:       32.21  2.3058764
##  ---
## 217:       27.32  7.0052770
## 218:       18.30 13.0760584
## 219:       67.80 10.6872940
## 220:       25.81  4.2142624
## 221:       27.14  1.8177210
```

The only variable for which NA values are found is the outcome variable *ViolentCrimesPerPop*. The rows where the outcome variable has an NA value are removed, as these rows are not useful for the regression. It can be noted that the variables *countyCode* and *communityCode* are also frequently unknown.

```
crimes_table_subset = na.omit(crimes_table_subset)
colSums(crimes_table_subset == "?", na.rm = TRUE)
```

```
##       communityname               state          countyCode       communityCode
##                   0                   0                1174                1177
##          population      PctPopUnderPov           perCapInc           PctEmploy
##                   0                   0                   0                   0
##     PctLess9thGrade        PctNotHSGrad         PctBSorMore        racepctblack
##                   0                   0                   0                   0
```

```
##        agePct12t29 ViolentCrimesPerPop           PctImmig
##                  0                   0                  0
```

After removing NA values from the database univariate descriptives are calculated.

```r
str(crimes_table_subset)
```

```
## Classes 'data.table' and 'data.frame':   1994 obs. of  15 variables:
##  $ communityname      : chr  "BerkeleyHeightstownship" "Marpletownship" "Tigardcity" "Gloversvillecit
##   ..- attr(*, "label")= Named chr "communityname"
##   .. ..- attr(*, "names")= chr "communityname"
##  $ state              : chr  "NJ" "PA" "OR" "NY" ...
##   ..- attr(*, "label")= Named chr "state"
##   .. ..- attr(*, "names")= chr "state"
##  $ countyCode         : chr  "39" "45" "?" "35" ...
##   ..- attr(*, "label")= Named chr "countyCode"
##   .. ..- attr(*, "names")= chr "countyCode"
##  $ communityCode      : chr  "5320" "47616" "?" "29443" ...
##   ..- attr(*, "label")= Named chr "communityCode"
##   .. ..- attr(*, "names")= chr "communityCode"
##  $ population         : int  11980 23123 29344 16656 140494 28700 59459 74111 103590 31601 ...
##   ..- attr(*, "label")= Named chr "population"
##   .. ..- attr(*, "names")= chr "population"
##  $ PctPopUnderPov     : num  1.96 3.98 4.75 17.23 17.78 ...
##   ..- attr(*, "label")= Named chr "people under the poverty level (%)"
##   .. ..- attr(*, "names")= chr "PctPopUnderPov"
##  $ perCapInc          : int  29711 20148 16946 10810 11878 18193 12161 13554 10195 12929 ...
##   ..- attr(*, "label")= Named chr "per capita income ($)"
##   .. ..- attr(*, "names")= chr "perCapInc"
##  $ PctEmploy          : num  64.5 62 69.8 54.7 59 ...
##   ..- attr(*, "label")= Named chr "percentage of people 16 and over who are employed (%)"
##   .. ..- attr(*, "names")= chr "PctEmploy"
##  $ PctLess9thGrade    : num  5.81 5.61 2.8 11.05 8.76 ...
##   ..- attr(*, "label")= Named chr "percentage of people 25 and over with less than a 9th grade educat
##   .. ..- attr(*, "names")= chr "PctLess9thGrade"
##  $ PctNotHSGrad       : num  9.9 13.72 9.09 33.68 23.03 ...
##   ..- attr(*, "label")= Named chr "percentage of people 25\nor over, that have not graduated highscho
##   .. ..- attr(*, "names")= chr "PctNotHSGrad"
##  $ PctBSorMore        : num  48.2 29.9 30.1 10.8 20.7 ...
##   ..- attr(*, "label")= Named chr "percentage of people 25 or over, with\nat least a bachelor's degre
##   .. ..- attr(*, "names")= chr "PctBSorMore"
##  $ racepctblack       : num  1.37 0.8 0.74 1.7 2.51 ...
##   ..- attr(*, "label")= Named chr "percentage of population that is african american (%)"
##   .. ..- attr(*, "names")= chr "racepctblack"
##  $ agePct12t29        : num  21.4 21.3 25.9 25.2 32.9 ...
##   ..- attr(*, "label")= Named chr "percentage of population that is 12-29 in age (%)"
##   .. ..- attr(*, "names")= chr "agePct12t29"
##  $ ViolentCrimesPerPop: num  41 128 219 307 443 ...
##   ..- attr(*, "label")= Named chr "total number of violent crimes per 100K population"
##   .. ..- attr(*, "names")= chr "ViolentCrimesPerPop"
##  $ PctImmig           : num  10.66 8.3 5 2.04 1.49 ...
##   ..- attr(*, "label")= Named chr "percentage of immigrants (%)"
##   .. ..- attr(*, "names")= chr "PctImmig"
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
summary(crimes_table_subset)
```

```
##  communityname          state            countyCode        communityCode     
##  Length:1994        Length:1994        Length:1994        Length:1994       
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##    population      PctPopUnderPov     perCapInc       PctEmploy    
##  Min.   :  10005   Min.   : 0.640   Min.   : 5237   Min.   :24.82  
##  1st Qu.:  14359   1st Qu.: 4.692   1st Qu.:11548   1st Qu.:56.35  
##  Median :  22681   Median : 9.650   Median :13977   Median :62.27  
##  Mean   :  52251   Mean   :11.796   Mean   :15522   Mean   :61.78  
##  3rd Qu.:  43154   3rd Qu.:17.078   3rd Qu.:17775   3rd Qu.:67.50  
##  Max.   :7322564   Max.   :48.820   Max.   :63302   Max.   :84.67  
##  PctLess9thGrade   PctNotHSGrad    PctBSorMore     racepctblack  
##  Min.   : 0.200   Min.   : 2.09   Min.   : 1.63   Min.   : 0.00  
##  1st Qu.: 4.770   1st Qu.:14.20   1st Qu.:14.09   1st Qu.: 0.94  
##  Median : 7.920   Median :21.66   Median :19.62   Median : 3.15  
##  Mean   : 9.444   Mean   :22.70   Mean   :22.99   Mean   : 9.51  
##  3rd Qu.:12.245   3rd Qu.:29.66   3rd Qu.:28.93   3rd Qu.:11.96  
##  Max.   :49.890   Max.   :73.66   Max.   :73.63   Max.   :96.67  
##   agePct12t29    ViolentCrimesPerPop    PctImmig      
##  Min.   : 9.38   Min.   :   0.0      Min.   : 0.1778  
##  1st Qu.:24.38   1st Qu.: 161.7      1st Qu.: 2.0753  
##  Median :26.77   Median : 374.1      Median : 4.4935  
##  Mean   :27.62   Mean   : 589.1      Mean   : 7.6062  
##  3rd Qu.:29.18   3rd Qu.: 794.4      3rd Qu.: 9.5848  
##  Max.   :70.51   Max.   :4877.1      Max.   :60.4013  
```

```r
crimes_table_subset %>%
  pivot_longer(cols = where(is.numeric), names_to = "variable", values_to = "value") %>%
  group_by(variable) %>%
  summarise(
    min = min(value, na.rm = TRUE),
    q25 = quantile(value, 0.25, na.rm = TRUE),
    mean = mean(value, na.rm = TRUE),
    sd = sd(value, na.rm = TRUE),
    q75 = quantile(value, 0.75, na.rm = TRUE),
    max = max(value, na.rm = TRUE),
    n = n(),
    NAs = sum(is.na(value))
  )
```

```
## # A tibble: 11 x 9
##    variable                min     q25   mean     sd    q75    max     n   NAs
##    <chr>                 <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <int> <int>
##  1 PctBSorMore           1.63  1.41e+1 2.30e1 1.25e1 2.89e1 7.36e1  1994     0
##  2 PctEmploy            24.8   5.64e+1 6.18e1 8.11e0 6.75e1 8.47e1  1994     0
##  3 PctImmig              0.178 2.08e+0 7.61e0 8.70e0 9.58e0 6.04e1  1994     0
##  4 PctLess9thGrade       0.2   4.77e+0 9.44e0 6.84e0 1.22e1 4.99e1  1994     0
##  5 PctNotHSGrad          2.09  1.42e+1 2.27e1 1.11e1 2.97e1 7.37e1  1994     0
```

```
##  6 PctPopUnderPov          0.64  4.69e+0 1.18e1 8.51e0 1.71e1 4.88e1  1994     0
##  7 ViolentCrimesPerPop     0     1.62e+2 5.89e2 6.15e2 7.94e2 4.88e3  1994     0
##  8 agePct12t29             9.38  2.44e+1 2.76e1 6.15e0 2.92e1 7.05e1  1994     0
##  9 perCapInc            5237     1.15e+4 1.55e4 6.23e3 1.78e4 6.33e4  1994     0
## 10 population          10005     1.44e+4 5.23e4 2.02e5 4.32e4 7.32e6  1994     0
## 11 racepctblack            0     9.4 e-1 9.51e0 1.41e1 1.20e1 9.67e1  1994     0
```

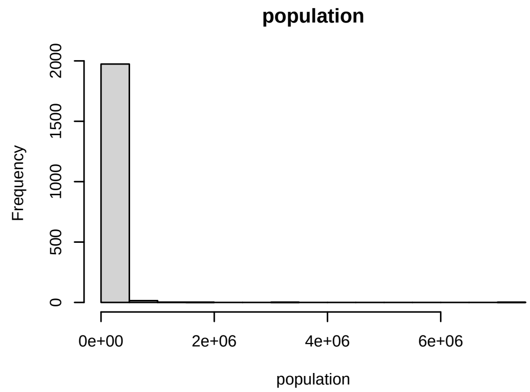
```
na_subset %>%
  pivot_longer(cols = where(is.numeric), names_to = "variable", values_to = "value") %>%
  group_by(variable) %>%
  summarise(
    min = min(value, na.rm = TRUE),
    q25 = quantile(value, 0.25, na.rm = TRUE),
    mean = mean(value, na.rm = TRUE),
    sd = sd(value, na.rm = TRUE),
    q75 = quantile(value, 0.75, na.rm = TRUE),
    max = max(value, na.rm = TRUE),
    n = n(),
    NAs = sum(is.na(value))
  )
```

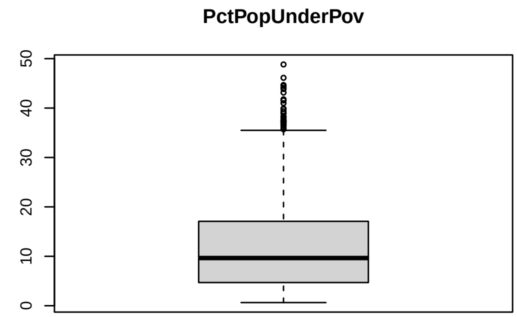
```
## # A tibble: 10 x 9
##    variable              min      q25     mean      sd    q75    max     n   NAs
##    <chr>               <dbl>    <dbl>    <dbl>   <dbl>  <dbl>  <dbl> <int> <int>
##  1 PctBSorMore          3.27     14.2     23.6  1.42e1 3.09e1 7.92e1   221     0
##  2 PctEmploy           33.7      57.4     64.2  9.71e0 7.04e1 8.45e1   221     0
##  3 PctImmig             0.445     1.99     4.94 4.62e0 6.27e0 2.84e1   221     0
##  4 PctLess9thGrade      0.41      3.54     6.86 4.12e0 9.64e0 2.12e1   221     0
##  5 PctNotHSGrad         1.46     11.1     18.7  9.63e0 2.54e1 5.26e1   221     0
##  6 PctPopUnderPov       1.25      3.49    10.0  9.25e0 1.33e1 5.8 e1   221     0
##  7 agePct12t29         17.4      25.0     27.9  6.48e0 2.94e1 6.78e1   221     0
##  8 perCapInc         5622     12205    16342.   6.72e3 1.81e4 6.25e4   221     0
##  9 population       10066     14903    60937.   2.26e5 4.18e4 2.78e6   221     0
## 10 racepctblack         0.03      0.49     7.76 1.54e1 6.4 e0 9.28e1   221     0
```

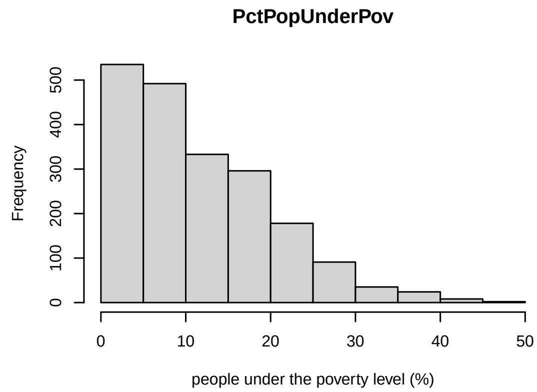
To gain insight into the univariate distributions, boxplots and histograms are generated.

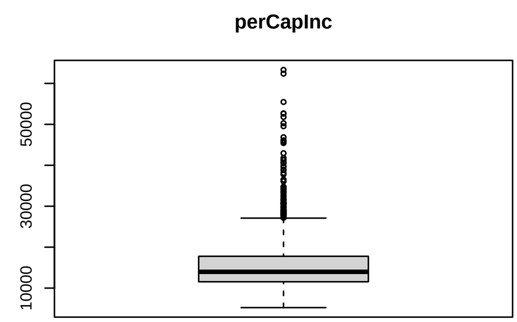
```
numeric_cols <- sapply(crimes_table_subset, is.numeric)
crimes_table_subset_num <- crimes_table_subset[, ..numeric_cols]
par(mfrow = c(4,2))
for(columnname in names(crimes_table_subset_num)){
  column <- crimes_table_subset_num[[columnname]]
  boxplot(column,
          main = columnname
          )
  hist(column,
       main = columnname,
       xlab = get_label(column)
          )
}
```

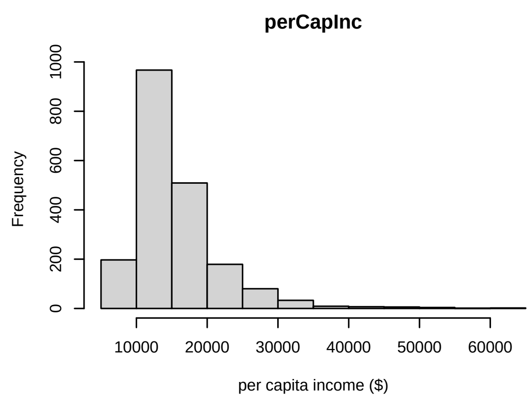

population
0e+00
2e+06
4e+06
6e+06

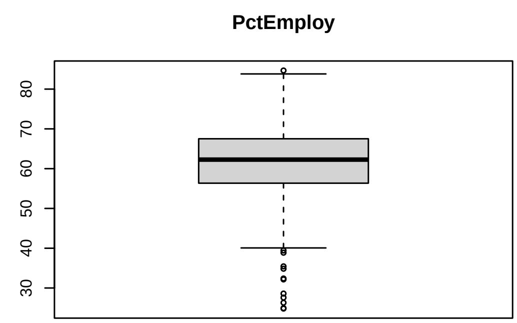

population
Frequency
0
500
1000
1500
2000
0e+00
2e+06
4e+06
6e+06
population

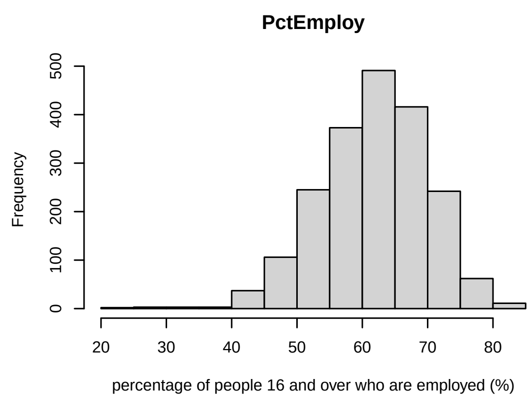

PctPopUnderPov
0
10
20
30
40
50

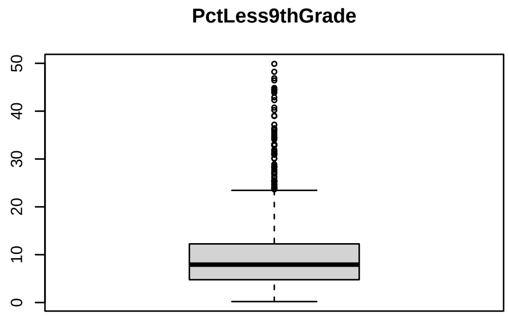

PctPopUnderPov
Frequency
0
100
200
300
400
500
0
10
20
30
40
50
people under the poverty level (%)

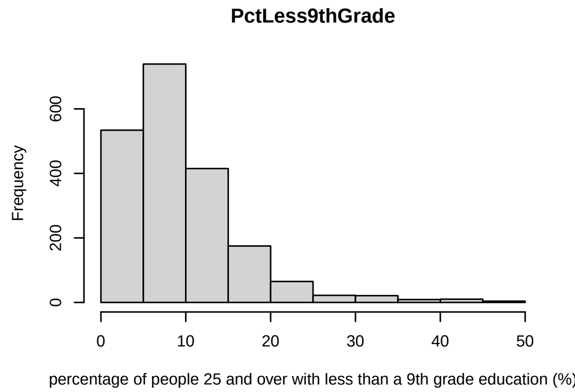

perCapInc
10000
30000
50000

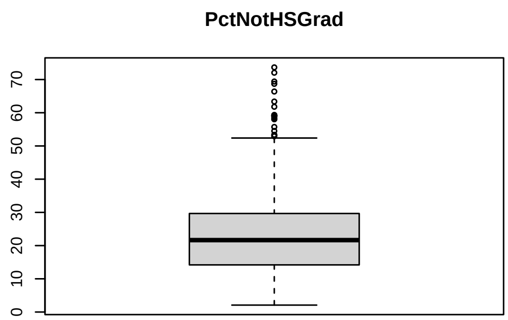

perCapInc
Frequency
0
200
400
600
800
1000
10000
20000
30000
40000
50000
60000
per capita income ($)

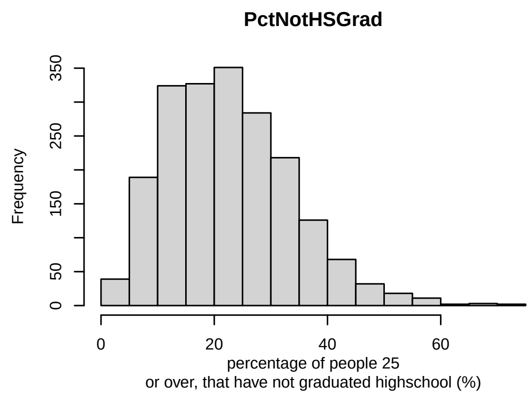

PctEmploy
30
40
50
60
70
80

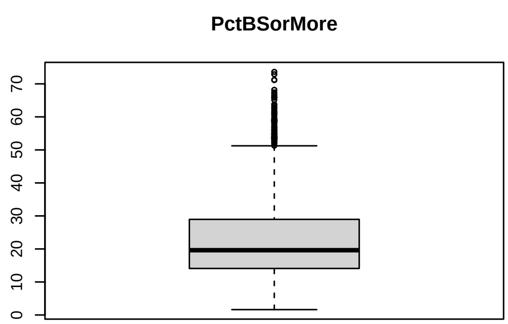

PctEmploy
Frequency
0
100
200
300
400
500
20
30
40
50
60
70
80
percentage of people 16 and over who are employed (%)

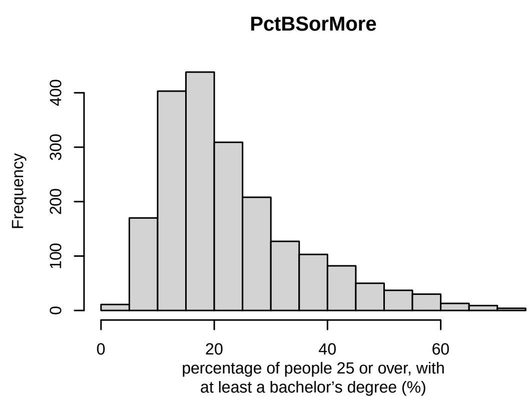

PctLess9thGrade

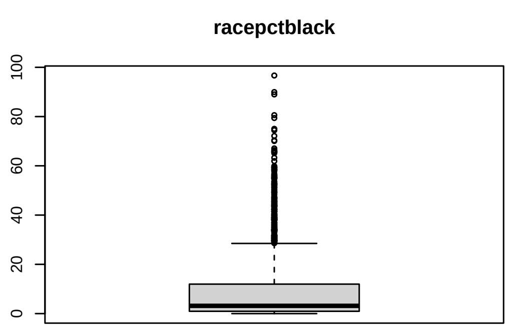

PctLess9thGrade
Frequency
percentage of people 25 and over with less than a 9th grade education (%)

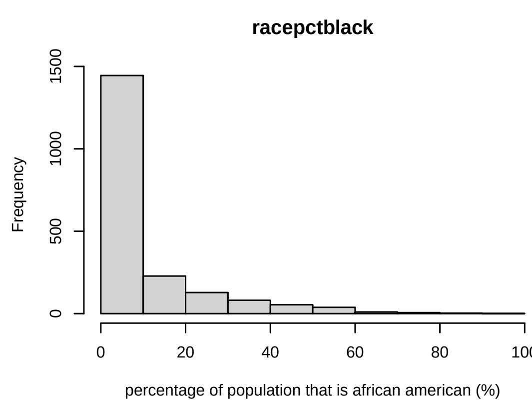

PctNotHSGrad

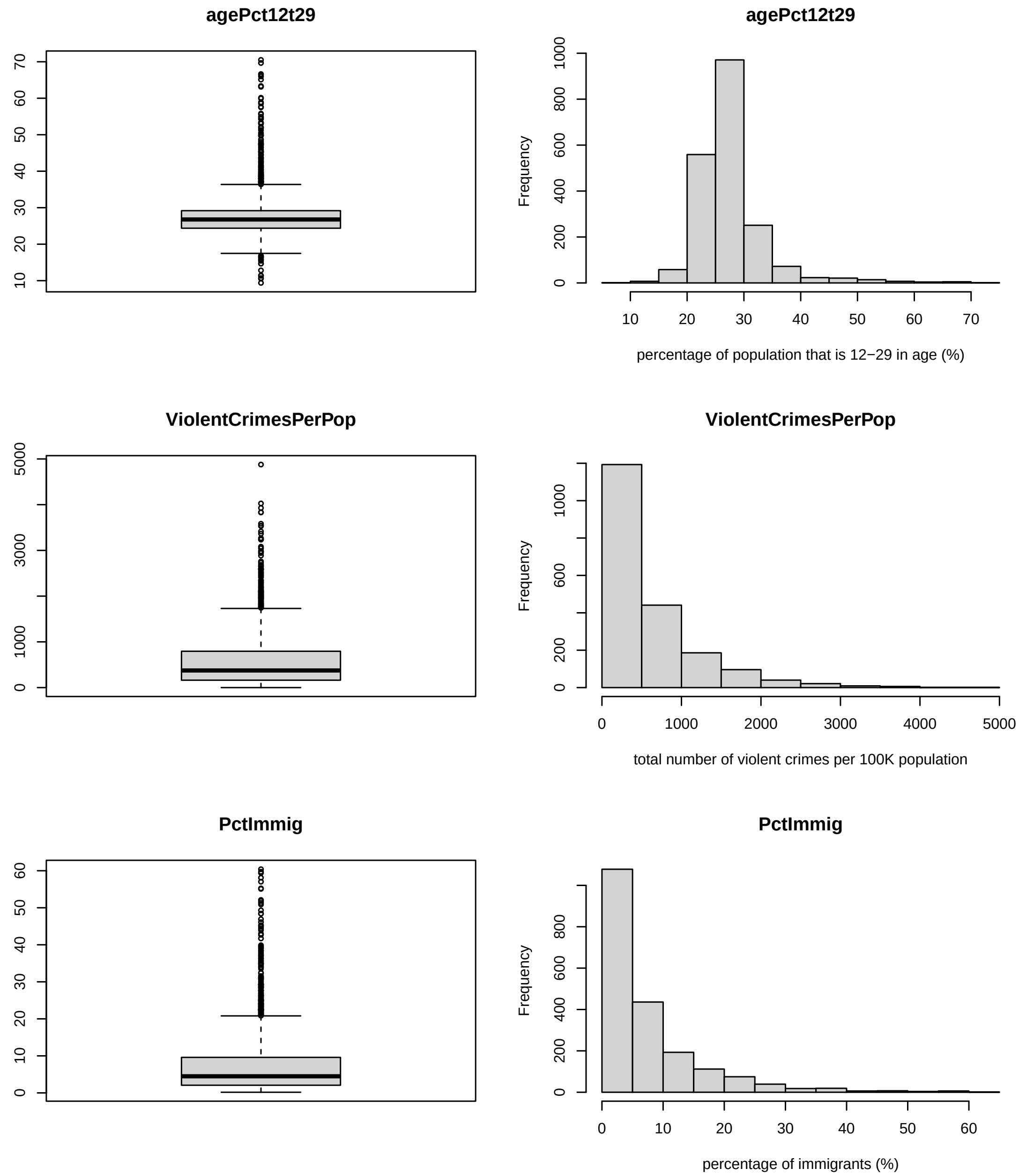

PctNotHSGrad
Frequency
percentage of people 25
or over, that have not graduated highschool (%)


PctBSorMore

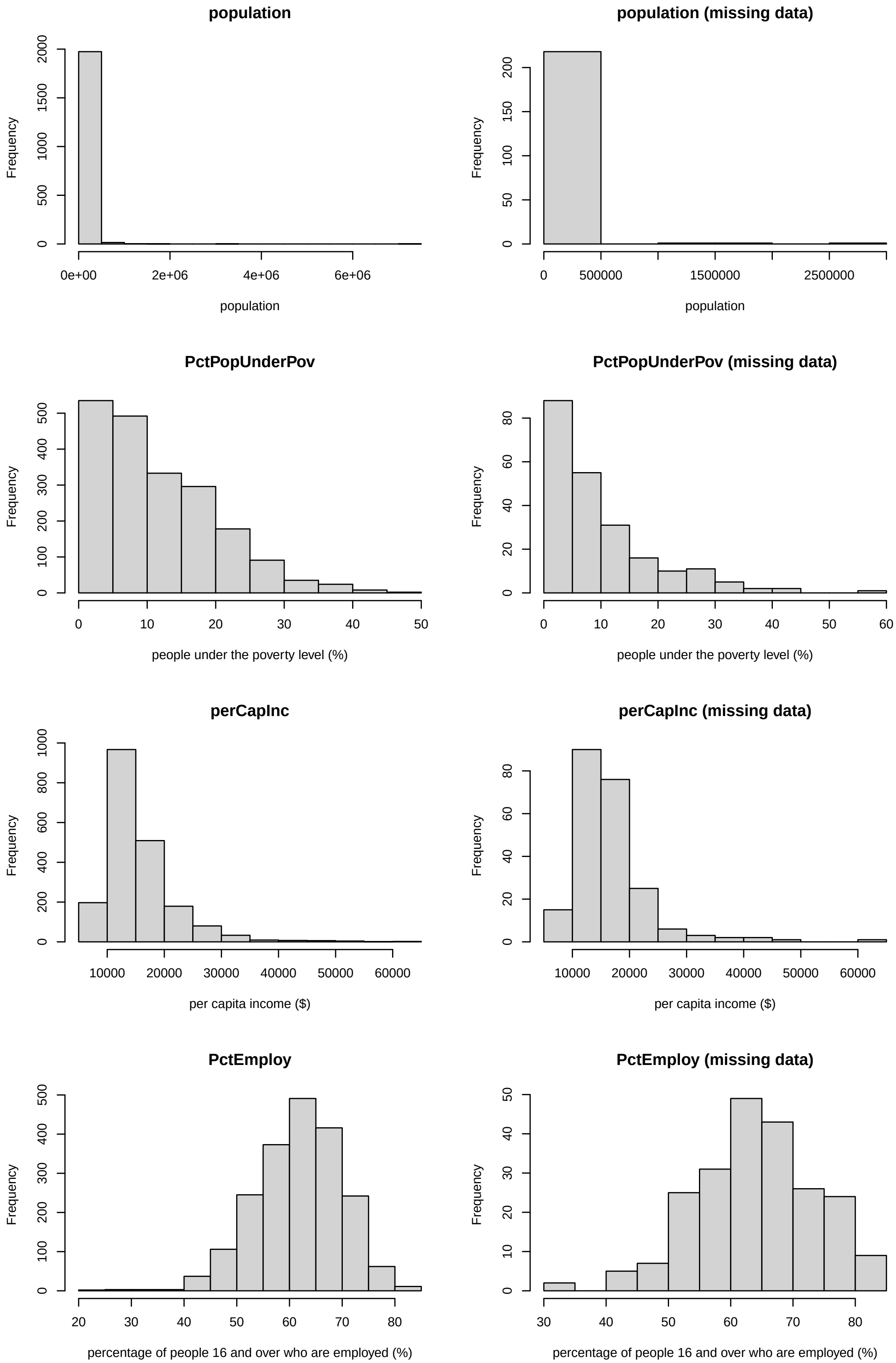

PctBSorMore
Frequency
percentage of people 25 or over, with
at least a bachelor's degree (%)

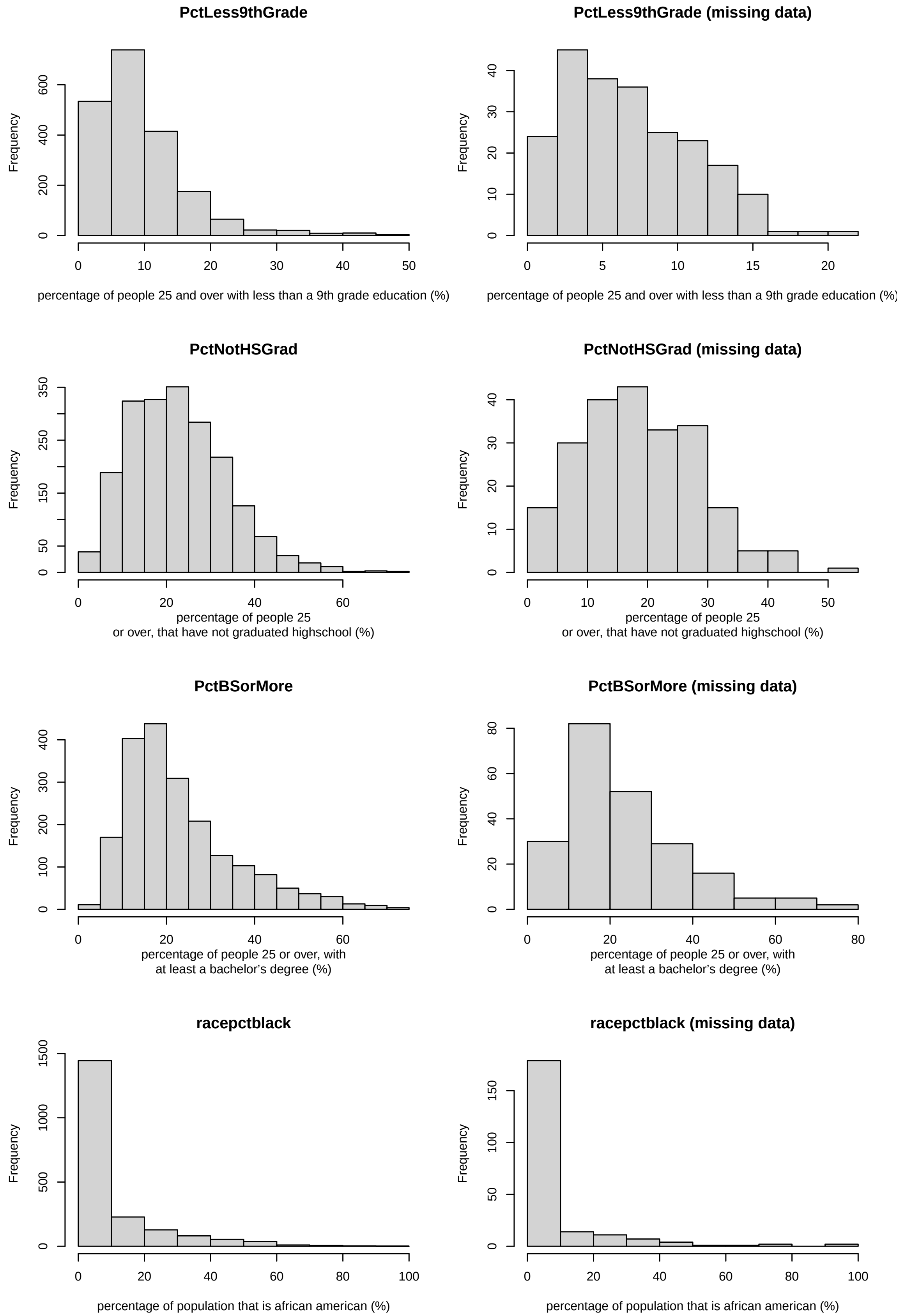

racepctblack

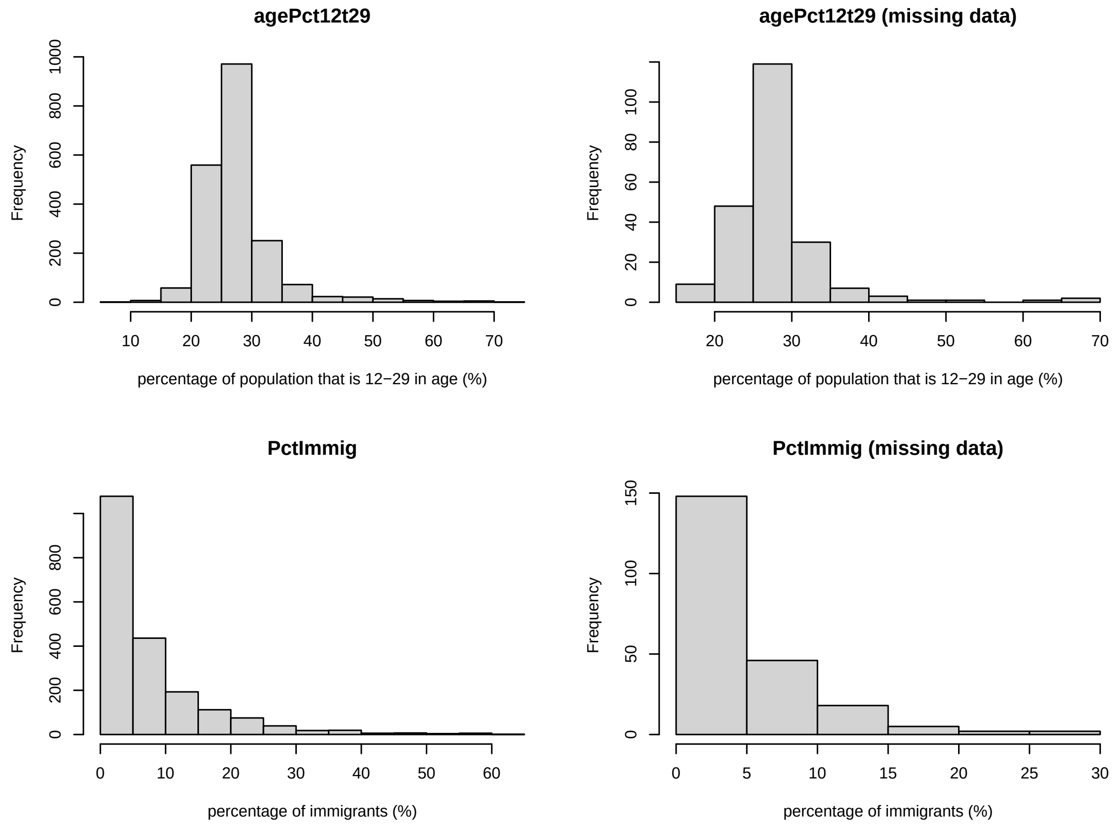

racepctblack
Frequency
percentage of population that is african american (%)


agePct12t29
agePct12t29
Frequency
percentage of population that is 12−29 in age (%)
ViolentCrimesPerPop
ViolentCrimesPerPop
Frequency
total number of violent crimes per 100K population
PctImmig
PctImmig
Frequency
percentage of immigrants (%)

```r
numeric_cols_na <- sapply(na_subset, is.numeric)
crimes_table_subset_num_na <- na_subset[, ..numeric_cols_na]
par(mfrow = c(4,2))
for(columnname in names(crimes_table_subset_num_na)){
  column <- crimes_table_subset_num[[columnname]]
  column_na <- crimes_table_subset_num_na[[columnname]]
  hist(column,
       main = columnname,
       xlab = get_label(column)
          )
  hist(column_na,
       main = paste(columnname, "(missing data)"),
       xlab = get_label(column)
          )
}
```


population
Frequency
population
population (missing data)
Frequency
population
PctPopUnderPov
Frequency
people under the poverty level (%)
PctPopUnderPov (missing data)
Frequency
people under the poverty level (%)
perCapInc
Frequency
per capita income ($)
perCapInc (missing data)
Frequency
per capita income ($)
PctEmploy
Frequency
percentage of people 16 and over who are employed (%)
PctEmploy (missing data)
Frequency
percentage of people 16 and over who are employed (%)


PctLess9thGrade
Frequency
percentage of people 25 and over with less than a 9th grade education (%)
PctLess9thGrade (missing data)
Frequency
percentage of people 25 and over with less than a 9th grade education (%)
PctNotHSGrad
Frequency
percentage of people 25
or over, that have not graduated highschool (%)
PctNotHSGrad (missing data)
Frequency
percentage of people 25
or over, that have not graduated highschool (%)
PctBSorMore
Frequency
percentage of people 25 or over, with
at least a bachelor's degree (%)
PctBSorMore (missing data)
Frequency
percentage of people 25 or over, with
at least a bachelor's degree (%)
racepctblack
Frequency
percentage of population that is african american (%)
racepctblack (missing data)
Frequency
percentage of population that is african american (%)


agePct12t29
Frequency
0
200
400
600
800
1000
10
20
30
40
50
60
70
percentage of population that is 12−29 in age (%)
agePct12t29 (missing data)
Frequency
0
20
40
60
80
100
20
30
40
50
60
70
percentage of population that is 12−29 in age (%)
PctImmig
Frequency
0
200
400
600
800
0
10
20
30
40
50
60
percentage of immigrants (%)
PctImmig (missing data)
Frequency
0
50
100
150
0
5
10
15
20
25
30
percentage of immigrants (%)

The correlation matrix shows the extent to which the variables in the dataset are correlated with each other. Below, all variables are listed, sorted from highest to lowest correlation with the outcome variable.

- *racepctblack* ($r = 0.63$)
- *PctPopUnderPov* ($r = 0.51$)
- *PctNotHSGrad* ($r = 0.47$)
- *PctLess9thGrade* ($r = 0.37$)
- *PctEmploy* ($r = -0.32$)
- *perCapInc* ($r = -0.32$)
- *PctBSorMore* ($r = 0.3$)
- *PctImmig* ($r = 0.19$)
- *agePct12t29* ($r = 0.11$)

It's important to mention that these correlations are indicators of an association, not of a causation.

The following predictors are highly correlated with each other. Therefore, it is best not to include them together in a model later.

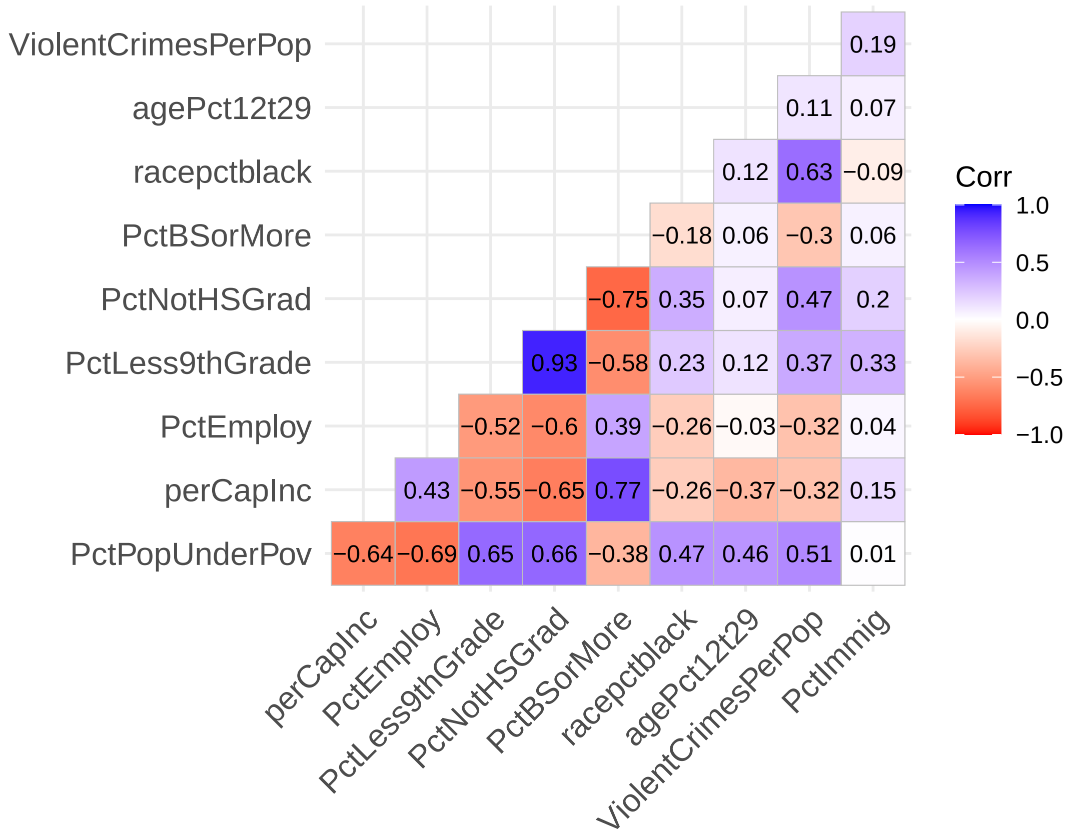
- *PctNotHSGrad* and *PctLess9thGrade* ($r = 0.93$)
- *perCapInc* and *PctBSorMore* ($r = 0.77$)
- *PctNotHSGrad* and *PctBSorMore* ($r = -0.75$)

However the choice for predictors for the model will be dealt with thoroughly during the model building.

It is noticeable that the variable *racepctblack* is the one most strongly correlated with the outcome variable ($r = 0.63$), even more than *PctPopUnderPov*, the head predictor that was chosen for this research.

It is noticable the the variables *PctImmig* and *PctImmig* have correlation coefficients tthat are really low.

```r
cor_matrix <- cor(crimes_table_subset_num[,-'population'])
cor_values <- as.data.frame(as.table(cor_matrix))

library(ggcorrplot)
ggcorrplot(cor_matrix, lab = TRUE, type = "lower",
           lab_size = 3, colors = c("red", "white", "blue"))
```



The following scatter plots were generated:

- for each variable, a scatter plot showing the relationship with the outcome variable *ViolentCrimesPerPop*;
- for each variable, a scatter plot showing the relationship with the main predictor variable *PctPopUnderPov*.

The first series of scatter plots indicates that not all variables have a linear relationship with *ViolentCrimesPerPop*. In particular, the following variables do not appear to exhibit a clear linear trend:

- *perCapInc*
- *agePct12t29* (which also had a very low correlation coefficient)

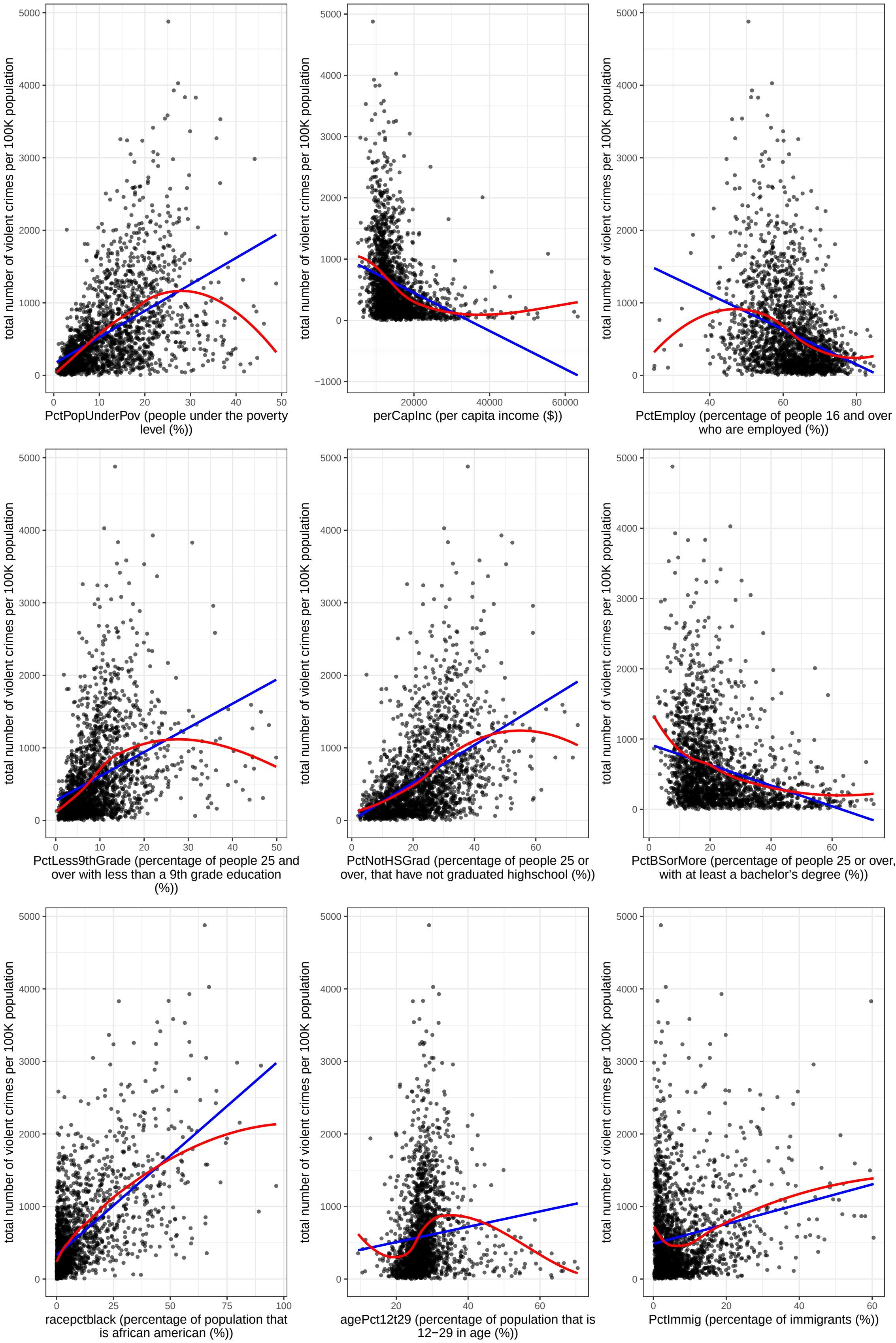
Other variables show a somewhat linear pattern, although this trend is often distorted in the extreme regions of the x-axis.

The second series of scatter plots suggests that some variables exhibit a linear relationship with the main predictor *PctPopUnderPov*. In particular, the following variables appear to show a fairly linear trend:

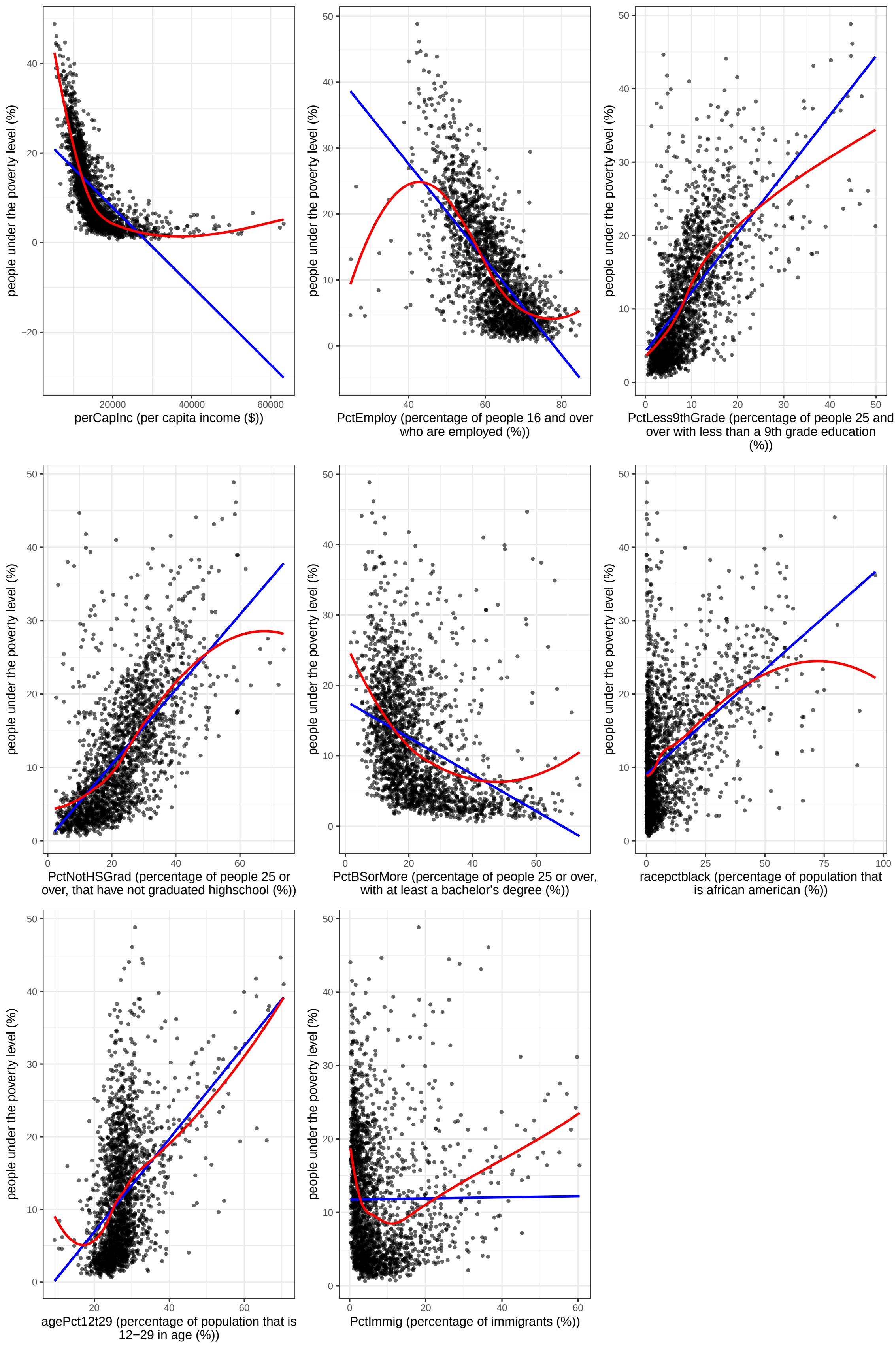
- *PctEmploy*
- *PctLess9thGrade*
- *PctNotHSGrad*

This implies that these variables are probably not suitable as additional predictors when *PctPopUnderPov* is already included in the model.

```r
x_vars <- colnames(crimes_table_subset_num)
dict_labels <- setNames(sapply(x_vars, function(x_var) get_label(crimes_table_subset_num[[x_var]])), x_

library(ggplot2)
library(patchwork)
df <- crimes_table_subset_num[,-c("population", "fold")]
y_var <- "ViolentCrimesPerPop"
x_vars <- setdiff(colnames(df), y_var)
plots <- lapply(x_vars, function(x_var) {
    ggplot(df, aes_string(x_var,y_var)) +
    geom_point(alpha = 0.6, size = 0.7) +
    geom_smooth(method = "lm", color = "blue", se = FALSE)+
    geom_smooth(method = "loess", color = "red", se = FALSE)+
    theme_bw(base_size = 8) +
    labs(x = str_wrap(paste(x_var, " (", dict_labels[x_var], ")", sep = ""), width = 45))
}
)

# Print 9 plots per pg
print(wrap_plots(plots, ncol = 3))
```


total number of violent crimes per 100K population
PctPopUnderPov (people under the poverty level (%))
perCapInc (per capita income ($))
PctEmploy (percentage of people 16 and over who are employed (%))
PctLess9thGrade (percentage of people 25 and over with less than a 9th grade education (%))
PctNotHSGrad (percentage of people 25 or over, that have not graduated highschool (%))
PctBSorMore (percentage of people 25 or over, with at least a bachelor's degree (%))
racepctblack (percentage of population that is african american (%))
agePct12t29 (percentage of population that is 12−29 in age (%))
PctImmig (percentage of immigrants (%))

```r
df <- crimes_table_subset_num[,-c("ViolentCrimesPerPop", "population", "fold")]
y_var <- "PctPopUnderPov"
x_vars <- setdiff(colnames(df), y_var)
plots <- lapply(x_vars, function(x_var) {
    ggplot(df, aes_string(x_var,y_var))+
    geom_point(alpha = 0.6, size = 0.7) +
    geom_smooth(method = "lm", color = "blue", se = FALSE)+
    geom_smooth(method = "loess", color = "red", se = FALSE)+
    theme_bw(base_size = 8) +
    labs(x = str_wrap(paste(x_var, " (", dict_labels[x_var], ")", sep = ""), width = 45))
}
)

# Print 9 plots per pg
print(wrap_plots(plots, ncol = 3))
```


people under the poverty level (%)
perCapInc (per capita income ($))
PctEmploy (percentage of people 16 and over who are employed (%))
PctLess9thGrade (percentage of people 25 and over with less than a 9th grade education (%))
PctNotHSGrad (percentage of people 25 or over, that have not graduated highschool (%))
PctBSorMore (percentage of people 25 or over, with at least a bachelor's degree (%))
racepctblack (percentage of population that is african american (%))
agePct12t29 (percentage of population that is 12−29 in age (%))
PctImmig (percentage of immigrants (%))

The scatter plots reveal one outlier for the ViolentCrimesPerPop variable. This outlier is the community Chestercity.

```
crimes_table_subset[order(crimes_table_subset_num$ViolentCrimesPerPop, decreasing=TRUE), , drop = FALSE]
```

```
##          communityname  state countyCode communityCode population
##                 <char> <char>     <char>        <char>      <int>
##    1:       Chestercity     PA         45         13208      41856
##    2:       Atlantacity     GA          ?             ?     394017
##    3:         Newarkcity     NJ         13         51000     275221
##    4:    Alexandriacity     LA          ?             ?      49188
##    5:         Miamicity     FL          ?             ?     358548
##   ---
## 1990:        Harvardtown     MA         27         28950      12329
## 1991:    Ogdensburgcity     NY         89         54485      13521
## 1992: Cranberrytownship     PA         19         16920      14816
## 1993:        Oswegocity     NY         75         55574      19195
## 1994:       Spencercity     IA         41         93955      11066
##       PctPopUnderPov perCapInc PctEmploy PctLess9thGrade PctNotHSGrad
##                <num>     <int>     <num>           <num>        <num>
##    1:          25.16      9115     50.54           13.42        37.84
##    2:          27.29     15279     56.97           10.96        30.13
##    3:          26.34      9424     51.51           21.97        48.79
##    4:          28.78     10887     51.28           14.08        31.37
##    5:          31.17      9799     53.19           30.89        52.37
##   ---
## 1990:           3.88     17937     82.48            0.66         2.93
## 1991:          13.97     11213     44.61           10.81        32.03
## 1992:           2.54     16494     71.33            1.97         9.51
## 1993:          19.05     11758     51.56            8.71        26.92
## 1994:           9.87     12805     66.26            6.32        14.70
##       PctBSorMore racepctblack agePct12t29 ViolentCrimesPerPop   PctImmig
##             <num>        <num>       <num>               <num>      <num>
##    1:        7.68        65.17       29.11             4877.06  2.0355505
##    2:       26.65        67.07       30.26             4026.59  3.3891939
##    3:        8.55        58.46       31.88             3928.03 18.6842574
##    4:       18.40        49.29       27.45             3834.10  1.1364560
##    5:       12.79        27.39       24.63             3829.21 59.7208742
##   ---
## 1990:       42.39        12.22       39.12                7.79  4.5259145
## 1991:       11.85         8.27       29.81                7.60  5.0809851
## 1992:       29.48         0.47       25.19                6.64  1.4916307
## 1993:       19.63         0.73       32.15                5.35  2.4277156
## 1994:       16.09         0.05       24.23                0.00  0.4518344
```

Model Building

Before performing linear regression and building models, the dataset is randomly split into a training set (80% of the data) and a holdout set (20% of the data). This holdout set will be used to validate the final model.

```r
n <- nrow(crimes_table_subset_num)
training <- sample(1:n, size = floor(0.8 * n))
train_data <- crimes_table_subset_num[training, ]
test_data <- crimes_table_subset_num[-training, ]

cat("Training set size:", nrow(train_data), "\n")
```

```
## Training set size: 1595
```

```r
cat("Test set size:", nrow(test_data), "\n")
```

```
## Test set size: 399
```

Univariate linear regression

The simple univariate regression equation we estimate with the training set is given as follows:

$$ViolentCrimesPerPop_i = \beta_0 + \beta_1 \cdot PctPopUnderPov_i + \epsilon_i$$

```r
fit <- lm(ViolentCrimesPerPop ~ PctPopUnderPov, data = train_data)
summary(fit)
```

```
##
## Call:
## lm(formula = ViolentCrimesPerPop ~ PctPopUnderPov, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1577.8  -253.4  -105.3   158.7  2903.6
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     164.534     22.386    7.35 3.16e-13 ***
## PctPopUnderPov   35.120      1.533   22.91  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 526.2 on 1593 degrees of freedom
## Multiple R-squared:  0.2478, Adjusted R-squared:  0.2473
## F-statistic: 524.8 on 1 and 1593 DF,  p-value: < 2.2e-16
```

We show the relevant statistics to be discussed in this section:

```r
cat("Regression equation: ViolentCrimesPerPop =",
    round(coef(fit)[1], 2), "+",
    round(coef(fit)[2], 2), "* PctPopUnderPov\n\n")
```

```
## Regression equation: ViolentCrimesPerPop = 164.53 + 35.12 * PctPopUnderPov
```

```r
# R-squared
cat("R-squared:", round(summary(fit)$r.squared, 4), "\n")
```

```
## R-squared: 0.2478
```

```r
cat("Adjusted R-squared:", round(summary(fit)$adj.r.squared, 4), "\n")
```

```
## Adjusted R-squared: 0.2473
```

```r
# MSE
mse_simple <- mean(fit$residuals^2)
cat("MSE:", round(mse_simple, 2), "\n")
```

```
## MSE: 276550.6
```

```r
# Confidence intervals for coefficients
cat("\n95% Confidence Intervals:\n")
```

```
##
## 95% Confidence Intervals:
```

```r
print(confint(fit))
```

```
##                    2.5 %    97.5 %
## (Intercept)    120.62559 208.44317
## PctPopUnderPov  32.11251  38.12673
```

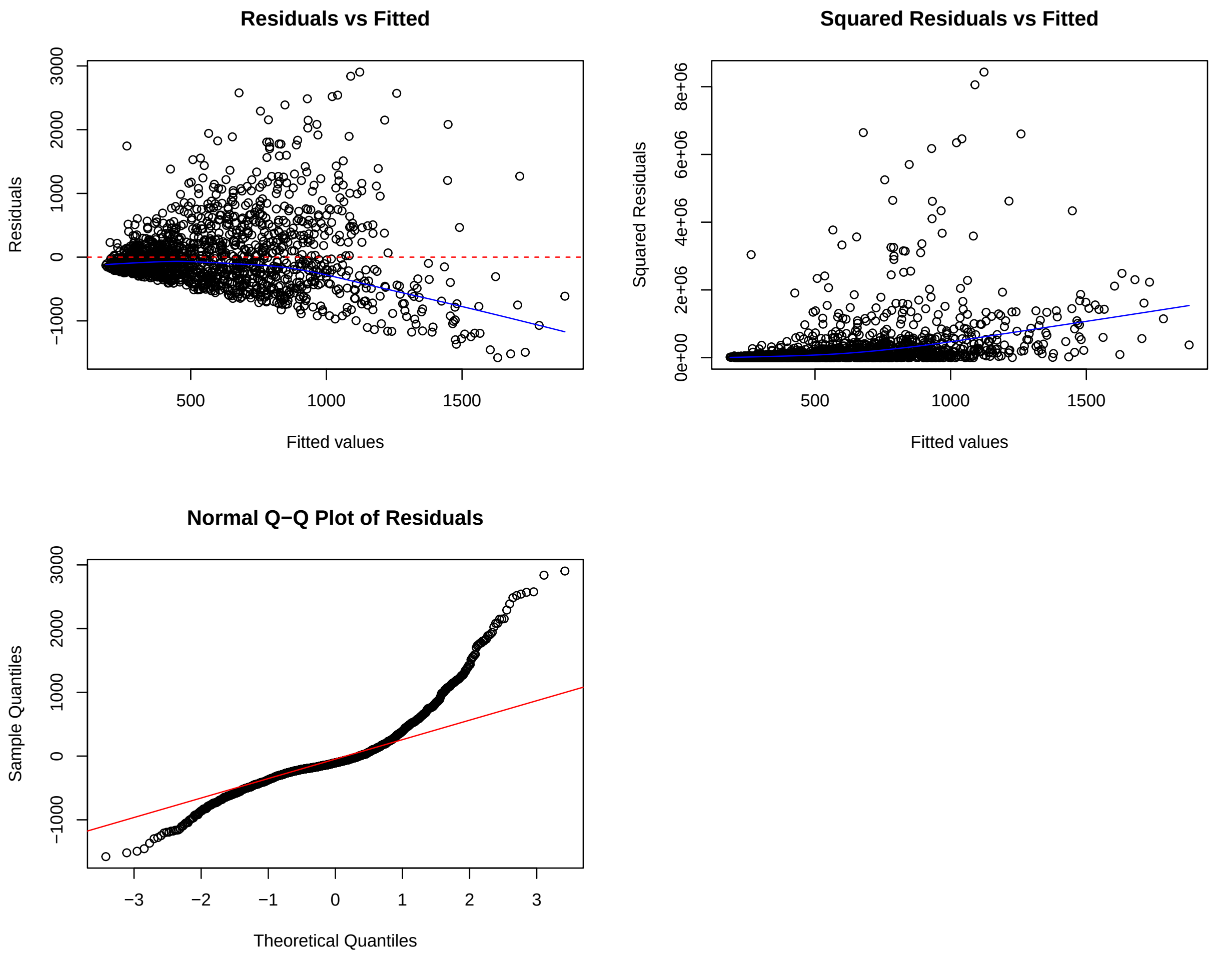
$PctPopUnderPov =$ increase in violent crimes per 100K population if poverty rate increases by one percentage point

Assumption checks

We check assumptions linearity, independence of errors, homoscedasticity, and normality of errors.

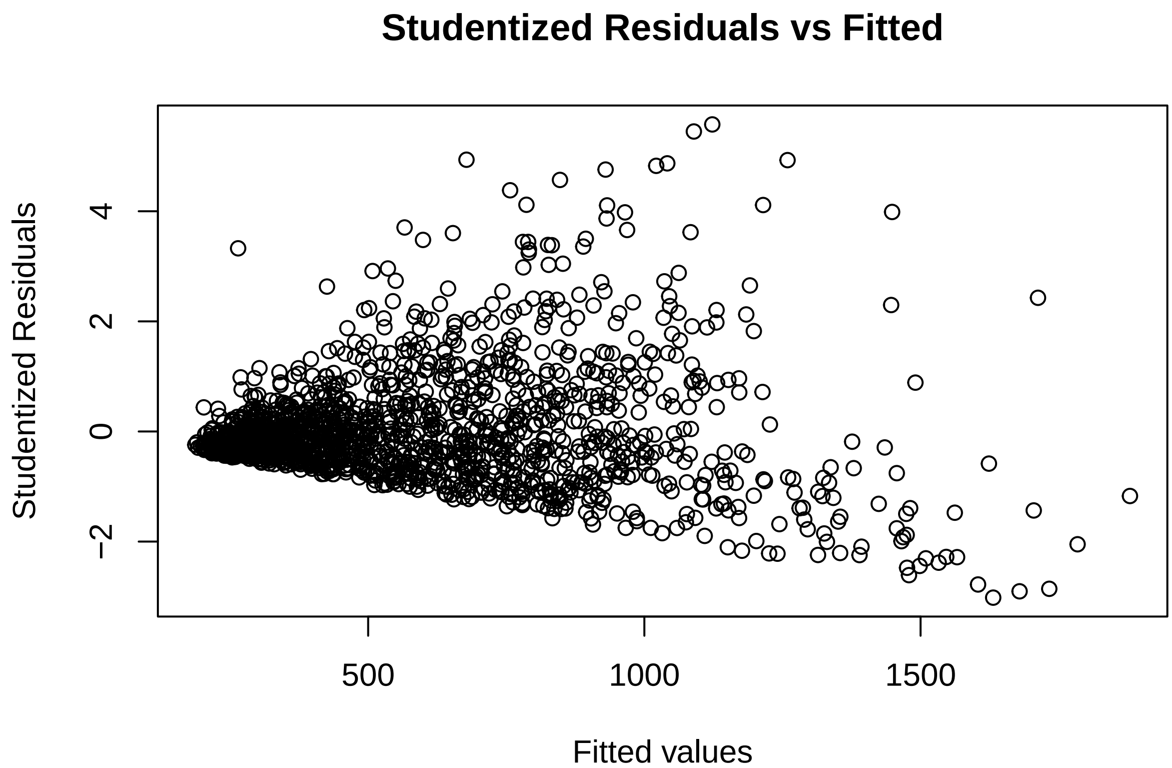
```r
par(mfrow = c(2, 2))

#Residuals vs Fitted
plot(fit$fitted.values, fit$residuals,
     xlab = "Fitted values", ylab = "Residuals",
     main = "Residuals vs Fitted")
abline(h = 0, col = "red", lty = 2)
lines(lowess(fit$fitted.values, fit$residuals), col = "blue")

# Squared residuals vs Fitted
plot(fit$fitted.values, fit$residuals^2,
     xlab = "Fitted values", ylab = "Squared Residuals",
     main = "Squared Residuals vs Fitted")
lines(lowess(fit$fitted.values, fit$residuals^2), col = "blue")
```

```r
# QQ-plot of residuals (normality)
qqnorm(fit$residuals, main = "Normal Q-Q Plot of Residuals")
qqline(fit$residuals, col = "red")
 par(mfrow = c(1, 1))
```



Studentized residuals

```r
# Studentized residuals plot
stud_res <- rstudent(fit)
plot(fit$fitted.values, stud_res,
     xlab = "Fitted values", ylab = "Studentized Residuals",
     main = "Studentized Residuals vs Fitted")
```




```
outliers_simple <- which(abs(stud_res) > 2)
```

Table to get a visual illustration of whether outliers are more common in small pop

```
# outliers are more present in small pop?
if(length(outliers_simple) > 0) {
  outlier_data <- train_data[outliers_simple, .(population, ViolentCrimesPerPop, PctPopUnderPov)]
  print(outlier_data)
}
```

```
##     population ViolentCrimesPerPop PctPopUnderPov
##          <int>               <num>          <num>
##  1:      23451             2344.56          17.55
##  2:      22754             2523.46          17.84
##  3:      37986             3583.48          24.97
##  4:      26623             3540.57          24.40
##  5:     164164             3047.66          22.79
##  6:      10201             2650.26          36.51
##  7:      10005             2119.93          24.78
##  8:      16027             1730.67          12.03
##  9:      10690              295.76          34.98
## 10:      12361              196.79          33.88
## 11:      49847             2037.32           9.78
## 12:      73552             2941.62          17.71
## 13:     109602             2078.85          16.47
```

```
## 14:       61945              2956.98          21.84
## 15:       14903              2018.97          19.62
## 16:       86905              1651.73           9.35
## 17:       30996              1938.43          15.96
## 18:       42523              2186.47          20.44
## 19:      228537              2017.65          18.88
## 20:       36118              2507.51          11.43
## 21:       12135              2584.96          17.53
## 22:       34311              2982.31          44.08
## 23:       12001              2451.06          19.59
## 24:       23478               212.45          34.88
## 25:       29541               352.88          39.35
## 26:       36830              2581.96          29.23
## 27:      175795              3255.71          14.62
## 28:       12915               150.31          40.99
## 29:       10170               137.89          32.74
## 30:       18666              1678.99          11.93
## 31:       10014                47.11          28.09
## 32:       28653               178.75          37.33
## 33:       12652               284.75          38.96
## 34:       30705               302.47          38.30
## 35:       52456               219.36          37.98
## 36:       33892              2572.91          25.56
## 37:       13547                39.87          28.82
## 38:       95706              2299.87          29.04
## 39:       75695              2089.28          18.75
## 40:       16491              1964.10          20.32
## 41:       71349              2541.38          13.92
## 42:      141686              1908.96          17.07
## 43:       17406              2008.28          13.67
## 44:       20807              2885.57          22.90
## 45:       22906              2193.23          25.55
## 46:      358548              3829.21          31.17
## 47:       70218              2264.09          21.73
## 48:       50627              2493.17          17.83
## 49:       22122              2728.14          20.77
## 50:       50961              1846.45          13.25
## 51:       10034                76.24          30.65
## 52:       29925              2594.03          17.80
## 53:      736014              3081.26          21.87
## 54:       13051                63.81          30.22
## 55:      139739              2288.32          27.51
## 56:       10398               278.92          33.20
## 57:       33830               370.67          39.91
## 58:       61921              2082.86          22.49
## 59:      372242              2601.60          18.82
## 60:       17363              2210.88          23.20
## 61:       26326              1607.89          10.36
## 62:       14302                53.73          41.77
## 63:       45549              2344.68          21.57
## 64:       25158              2605.96          19.02
## 65:       87492              3530.78          36.56
## 66:      176664              1849.26          16.79
## 67:       26866              3364.91          29.91
```

```
## 68:      88675             2065.29          18.05
## 69:     741952             1682.47          12.47
## 70:      20651              712.74          46.12
## 71:     222103             1807.31           7.43
## 72:      10404             1759.00          14.79
## 73:     275221             3928.03          26.34
## 74:      18942             2241.85          25.11
## 75:      30326             1884.24          18.65
## 76:      34590              112.41          37.43
## 77:     672971             1681.08          12.81
## 78:      43467             3414.57          21.79
## 79:      47669             1964.89          17.60
## 80:    3485398             2414.77          18.86
## 81:      15520             2089.32          10.57
## 82:     219531             2978.69          26.17
## 83:     395934             1787.20          10.83
## 84:    7322564             2097.71          19.29
## 85:     606900             3048.38          16.87
## 86:     265968             2466.68          24.82
## 87:      15023             1987.15          10.97
## 88:      98052             2109.96          21.17
## 89:      35701             1678.08           9.60
## 90:      13024             2649.80          20.64
## 91:      12822              161.69          43.13
## 92:      49998             1818.82          15.48
## 93:      21265              239.75          44.66
## 94:      61018             2423.47          12.38
## 95:      10864             2333.62          25.07
## 96:     394017             4026.59          27.29
## 97:     280015             3235.45          19.44
## 98:     574283             1968.89          18.71
## 99:      19378             2008.66           2.85
##     population ViolentCrimesPerPop PctPopUnderPov
```

Model selection

We use an all-possible regressions procedure to select predictor variables. We include models with a maximum of 5 predictor variables and only models with 0, 1, or 2 education predictor variables. The best model is chosen based on the Bayesian Information Criterion.

```r
library(leaps)

# Define predictor variables for model selection
predictors <- c("PctPopUnderPov", "perCapInc", "PctEmploy",
                "PctLess9thGrade", "PctNotHSGrad", "PctBSorMore",
                "racepctblack", "agePct12t29", "PctImmig")

# Educ variables
educ <- c("PctLess9thGrade", "PctNotHSGrad", "PctBSorMore")

# data for model selection
data <- train_data[, c("ViolentCrimesPerPop", predictors), with = FALSE]
```

```r
# Test reg with max 5 predictors
all_combos <- regsubsets(ViolentCrimesPerPop ~ ., data = data,
                         nvmax = 5, nbest = 10, method = "exhaustive")
all_combos_sum <- summary(all_combos)
```

```r
# Create results dataframe
results <- data.frame(
  n_predictors = apply(all_combos_sum$which[, -1], 1, sum),
  predictors = apply(all_combos_sum$which[, -1], 1, function(x)
    paste(names(x)[x], collapse = ", ")),
  rsq = all_combos_sum$rsq,
  adjrsq = all_combos_sum$adjr2,
  cp = all_combos_sum$cp,
  bic = all_combos_sum$bic
)

# Count education variables in each model
count_edu_vars <- function(pred_string) {
  sum(sapply(educ, function(v) grepl(v, pred_string)))
}
results$n_edu <- sapply(results$predictors, count_edu_vars)

# Filter: only models with 0, 1, or 2 education variables
BIC_ranking <- results[results$n_edu <= 2, ]

# Sort by BIC (lower is better)
BIC_ranking<- BIC_ranking[order(BIC_ranking$bic), ]

print(head(BIC_ranking, 10))
```

```
##    n_predictors
## 40            5
## 41            5
## 42            5
## 43            5
## 30            4
## 44            5
## 45            5
## 46            5
## 47            5
## 31            4
##                                                               predictors
## 40 PctPopUnderPov, PctLess9thGrade, PctNotHSGrad, racepctblack, PctImmig
## 41  PctPopUnderPov, PctLess9thGrade, PctBSorMore, racepctblack, PctImmig
## 42      PctPopUnderPov, PctBSorMore, racepctblack, agePct12t29, PctImmig
## 43        PctPopUnderPov, perCapInc, PctBSorMore, racepctblack, PctImmig
## 30                   PctPopUnderPov, PctBSorMore, racepctblack, PctImmig
## 44        PctPopUnderPov, PctEmploy, PctBSorMore, racepctblack, PctImmig
## 45     PctPopUnderPov, PctNotHSGrad, PctBSorMore, racepctblack, PctImmig
## 46        PctPopUnderPov, perCapInc, racepctblack, agePct12t29, PctImmig
## 47     PctPopUnderPov, PctNotHSGrad, racepctblack, agePct12t29, PctImmig
## 31                 PctLess9thGrade, PctNotHSGrad, racepctblack, PctImmig
##          rsq    adjrsq        cp       bic n_edu
```

```
## 40 0.5439255 0.5424904  54.34122 -1207.985     2
## 41 0.5316409 0.5301671  98.44391 -1165.591     2
## 42 0.5312988 0.5298240  99.67186 -1164.427     1
## 43 0.5295864 0.5281062 105.81953 -1158.610     1
## 30 0.5260723 0.5248801 116.43535 -1154.114     1
## 44 0.5263531 0.5248627 117.42728 -1147.685     1
## 45 0.5261590 0.5246680 118.12410 -1147.031     2
## 46 0.5253385 0.5238449 121.07004 -1144.272     0
## 47 0.5233031 0.5218031 128.37703 -1137.447     1
## 31 0.5198632 0.5186553 138.72673 -1133.353     2
```

We then run the multivariate regression equation

```r
# Print best model
best <- which.min(BIC_ranking$bic)
best_pred <- BIC_ranking$predictors[best]
cat("\nBest model:\n")
```

```
##
## Best model:
```

```r
cat("Predictors:", best_pred, "\n")
```

```
## Predictors: PctPopUnderPov, PctLess9thGrade, PctNotHSGrad, racepctblack, PctImmig
```

```r
cat("BIC:", round(BIC_ranking$bic[best], 2), "\n")
```

```
## BIC: -1207.99
```

```r
cat("Adjusted R²:", round(BIC_ranking$adjrsq[best], 4), "\n")
```

```
## Adjusted R²: 0.5425
```

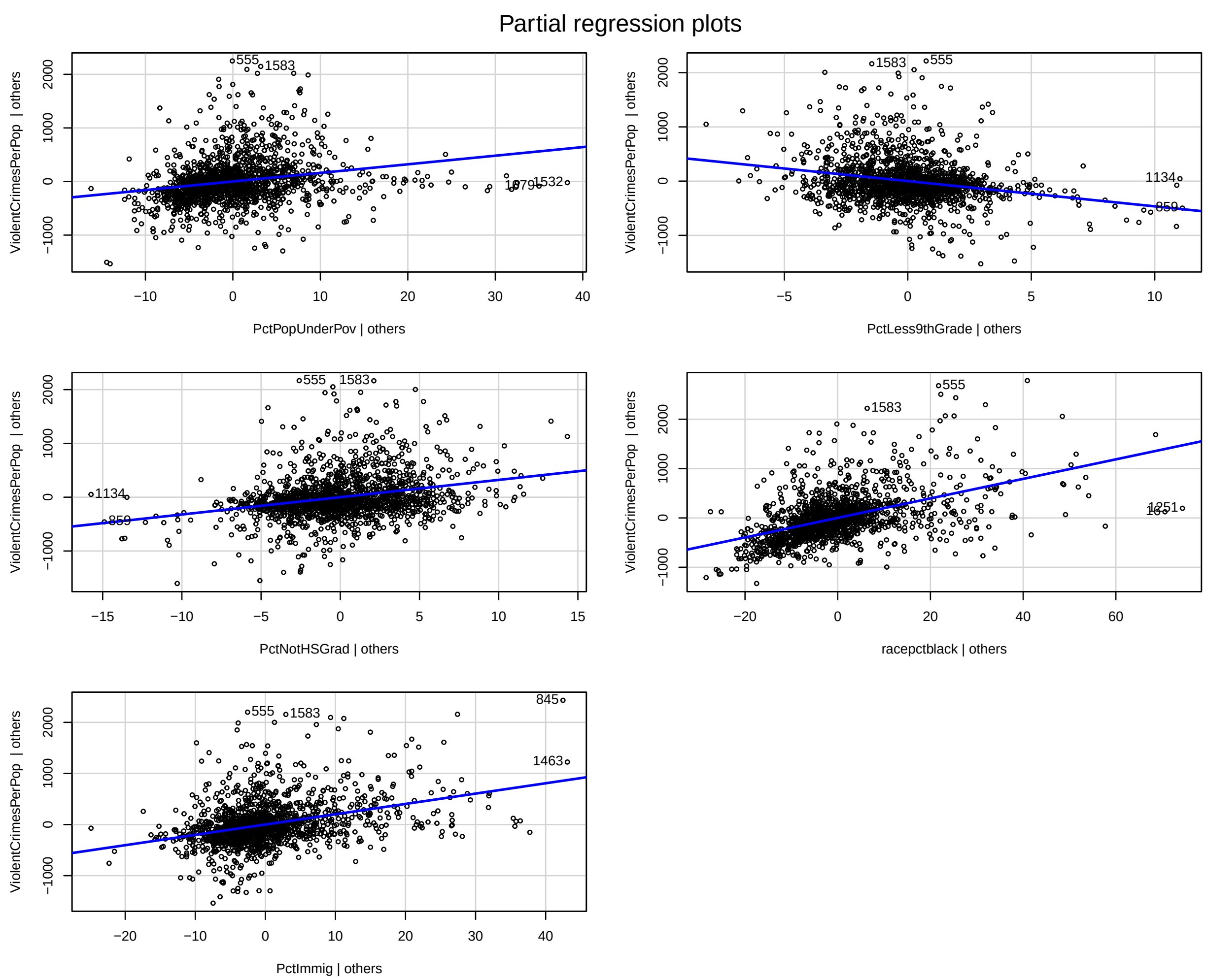
```r
# store pred
best_pred_sel <- strsplit(best_pred, ", ")[[1]]

# multivariate regression with these predictors
formula_multi <- as.formula(paste("ViolentCrimesPerPop ~",
                                  paste(best_pred_sel, collapse = " + ")))
fit_multi <- lm(formula_multi, data = train_data)
summary(fit_multi)
```

```
##
## Call:
## lm(formula = formula_multi, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1387.36  -214.80   -46.72   138.35  2250.30
##
```

```
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -239.5160    30.8735  -7.758 1.53e-14 ***
## PctPopUnderPov    16.0549     1.7535   9.156  < 2e-16 ***
## PctLess9thGrade  -46.5431     4.8312  -9.634  < 2e-16 ***
## PctNotHSGrad      32.1548     2.8440  11.306  < 2e-16 ***
## racepctblack      19.7877     0.8764  22.578  < 2e-16 ***
## PctImmig          20.1995     1.3387  15.089  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 410.3 on 1589 degrees of freedom
## Multiple R-squared:  0.5439, Adjusted R-squared:  0.5425
## F-statistic:   379 on 5 and 1589 DF,  p-value: < 2.2e-16
```

Partial Regression Plots

```r
library(car)
avPlots(fit_multi, main = "Partial regression plots")
```



Interaction Terms Selection

Following the protocol, we add all interaction terms of the predictor variables with our main predictor variable *PctPopUnderPov* and evaluate their significance. We choose the best one.

```r
# store all other pred except for main pred
other <- setdiff(best_pred_sel, "PctPopUnderPov")

  # Create interaction terms
  interaction_terms <- paste("PctPopUnderPov", other, sep = ":")

  # Evaluate each interaction term individually
  interaction_results <- data.frame(
    interaction = interaction_terms,
    t_value = NA,
    p_value = NA,
    delta_adjrsq = NA
  )

  for(i in seq_along(interaction_terms)) {
    formula_single_int <- as.formula(paste("ViolentCrimesPerPop ~",
                                           paste(best_pred_sel, collapse = " + "), "+",
                                           interaction_terms[i]))
    fit_single_int <- lm(formula_single_int, data = train_data)
    coef_summary <- summary(fit_single_int)$coefficients
    int_row <- nrow(coef_summary)
    interaction_results$t_value[i] <- coef_summary[int_row, "t value"]
    interaction_results$p_value[i] <- coef_summary[int_row, "Pr(>|t|)"]
    interaction_results$delta_adjrsq[i] <- summary(fit_single_int)$adj.r.squared -
                                           summary(fit_multi)$adj.r.squared
  }

  interaction_results <- interaction_results[order(interaction_results$p_value), ]
  cat("Interaction terms ranked by p-value:\n")
```

```
## Interaction terms ranked by p-value:
```

```r
  print(interaction_results)
```

```
##                       interaction    t_value      p_value  delta_adjrsq
## 4        PctPopUnderPov:PctImmig  5.4414573 6.111972e-08  0.0080916330
## 2    PctPopUnderPov:PctNotHSGrad  1.9211681 5.488936e-02  0.0007734583
## 3    PctPopUnderPov:racepctblack  0.4527597 6.507836e-01 -0.0002290158
## 1 PctPopUnderPov:PctLess9thGrade -0.1743757 8.615924e-01 -0.0002793386
```

```r
  # Select best interaction
  best_interaction <- interaction_results$interaction[1]
  cat("\nSelected interaction term:", best_interaction, "\n")
```

```
##
## Selected interaction term: PctPopUnderPov:PctImmig
```

Multivariate Model

Based on the model selection procedure, we fit the multivariate model including the selected interaction term.

```r
# Estimate model with ineraction
formula_final <- as.formula(paste("ViolentCrimesPerPop ~",
                                  paste(best_pred_sel, collapse = " + "), "+",
                                  best_interaction))

fit_final <- lm(formula_final, data = train_data)
summary(fit_final)
```

```
##
## Call:
## lm(formula = formula_final, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1469.96  -207.36   -46.58   133.18  2216.43
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -156.7547    34.1708  -4.587 4.84e-06 ***
## PctPopUnderPov           10.6785     1.9991   5.342 1.05e-07 ***
## PctLess9thGrade         -53.5485     4.9583 -10.800  < 2e-16 ***
## PctNotHSGrad             34.3326     2.8470  12.059  < 2e-16 ***
## racepctblack             20.7291     0.8857  23.404  < 2e-16 ***
## PctImmig                  9.5361     2.3665   4.030 5.85e-05 ***
## PctPopUnderPov:PctImmig   0.7696     0.1414   5.441 6.11e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 406.6 on 1588 degrees of freedom
## Multiple R-squared:  0.5523, Adjusted R-squared:  0.5506
## F-statistic: 326.5 on 6 and 1588 DF,  p-value: < 2.2e-16
```

```r
# confint
print(confint(fit_final))
```

```
##                               2.5 %     97.5 %
## (Intercept)             -223.7792737 -89.730059
## PctPopUnderPov             6.7573820  14.599712
## PctLess9thGrade          -63.2741119 -43.822938
## PctNotHSGrad              28.7483479  39.916846
## racepctblack              18.9918724  22.466427
## PctImmig                   4.8942597  14.178011
## PctPopUnderPov:PctImmig    0.4921651   1.046971
```

Multicollinearity Check

Before checking model assumptions, we first assess multicollinearity using the Variance Inflation Factor (VIF) and remove a variable if necessary.

```r
library(car)
vif <- vif(fit_final)
print(vif)
```

```
##          PctPopUnderPov         PctLess9thGrade            PctNotHSGrad 
##                2.847684               10.651389                9.405196 
##            racepctblack                PctImmig PctPopUnderPov:PctImmig 
##                1.508038                4.034832                5.650188
```

There is multicollinearity, as PctNotHSGrad and PctLess9thGrade are highly correlated (also already derived before from Ilja's plots). Thus, we remove the variable with the highest VIF (PctLess9thGrade). We then do our model evaluation again, and find that the most relevant interaction term to include is now PctPopUnderPov*PctLess9thGrade

```r
# Highest vif variable
main_effects_vif <- vif[!grepl(":", names(vif))]
highest_vif_var <- names(which.max(main_effects_vif))

# remove
best_predictors_reduced <- setdiff(best_pred_sel, highest_vif_var)

# Re-evaluate interaction terms without the removed variable
other_predictors_reduced <- setdiff(best_predictors_reduced, "PctPopUnderPov")
interaction_terms_reduced <- paste("PctPopUnderPov", other_predictors_reduced, sep = ":")

interaction_results_reduced <- data.frame(
  interaction = interaction_terms_reduced,
  t_value = NA,
  p_value = NA
)

for(i in seq_along(interaction_terms_reduced)) {
  formula_int <- as.formula(paste("ViolentCrimesPerPop ~",
                                  paste(best_predictors_reduced, collapse = " + "), "+",
                                  interaction_terms_reduced[i]))
  fit_int <- lm(formula_int, data = train_data)
  coef_summary <- summary(fit_int)$coefficients
  int_row <- nrow(coef_summary)
  interaction_results_reduced$t_value[i] <- coef_summary[int_row, "t value"]
  interaction_results_reduced$p_value[i] <- coef_summary[int_row, "Pr(>|t|)"]
}

interaction_results_reduced <- interaction_results_reduced[order(interaction_results_reduced$p_value), ]
cat("Interaction terms ranked by p-value:\n")
```

```
## Interaction terms ranked by p-value:
```

```r
print(interaction_results_reduced)
```

```
##                    interaction    t_value     p_value
## 1 PctPopUnderPov:PctNotHSGrad -2.6951745 0.007109343
## 3     PctPopUnderPov:PctImmig  2.6367573 0.008451837
## 2 PctPopUnderPov:racepctblack  0.4092429 0.682416558
```

```r
best_interaction_reduced <- interaction_results_reduced$interaction[1]
cat("\nSelected interaction term:", best_interaction_reduced, "\n")
```

```
##
## Selected interaction term: PctPopUnderPov:PctNotHSGrad
```

```r
# fit reduced model
formula_final_reduced <- as.formula(paste("ViolentCrimesPerPop ~",
                                          paste(best_predictors_reduced, collapse = " + "), "+",
                                          best_interaction_reduced))

fit_final_reduced <- lm(formula_final_reduced, data = train_data)
summary(fit_final_reduced)
```

```
##
## Call:
## lm(formula = formula_final_reduced, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1476.20  -212.54   -54.97   134.18  2199.62
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -133.70670   38.56283  -3.467  0.00054 ***
## PctPopUnderPov                17.65331    2.94010   6.004 2.38e-09 ***
## PctNotHSGrad                  11.38134    1.89731   5.999 2.46e-09 ***
## racepctblack                  22.40867    0.85778  26.124  < 2e-16 ***
## PctImmig                      15.87911    1.28951  12.314  < 2e-16 ***
## PctPopUnderPov:PctNotHSGrad   -0.25780    0.09565  -2.695  0.00711 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 421.1 on 1589 degrees of freedom
## Multiple R-squared:  0.5195, Adjusted R-squared:  0.518
## F-statistic: 343.6 on 5 and 1589 DF,  p-value: < 2.2e-16
```

```r
cat("\n95% Confidence Intervals:\n")
```

```
##
## 95% Confidence Intervals:
```

```r
print(confint(fit_final_reduced))
```

```
##                                    2.5 %       97.5 %
## (Intercept)                 -209.3460701 -58.06733768
## PctPopUnderPov                11.8864291  23.42018896
## PctNotHSGrad                   7.6598528  15.10282500
## racepctblack                  20.7261604  24.09117320
## PctImmig                      13.3497821  18.40843639
## PctPopUnderPov:PctNotHSGrad   -0.4454155  -0.07018133
```

```r
# check vif again-->Correct
vif_adapted <- vif(fit_final_reduced)
print(vif_adapted)
```

```
##              PctPopUnderPov                PctNotHSGrad
##                    5.742771                    3.894478
##                racepctblack                    PctImmig
##                    1.318760                    1.116932
## PctPopUnderPov:PctNotHSGrad
##                   10.608979
```

We see that our model performs only a little less well, but this way we did account for multicollinearity and our estimates are correct.

```r
# Compare models
cat("Comparison of models:\n")
```

```
## Comparison of models:
```

```r
cat("Original model adjusted R²: ", round(summary(fit_multi)$adj.r.squared, 4), "\n")
```

```
## Original model adjusted R²:  0.5425
```

```r
cat("Reduced model adjusted R²:  ", round(summary(fit_final_reduced)$adj.r.squared, 4), "\n")
```

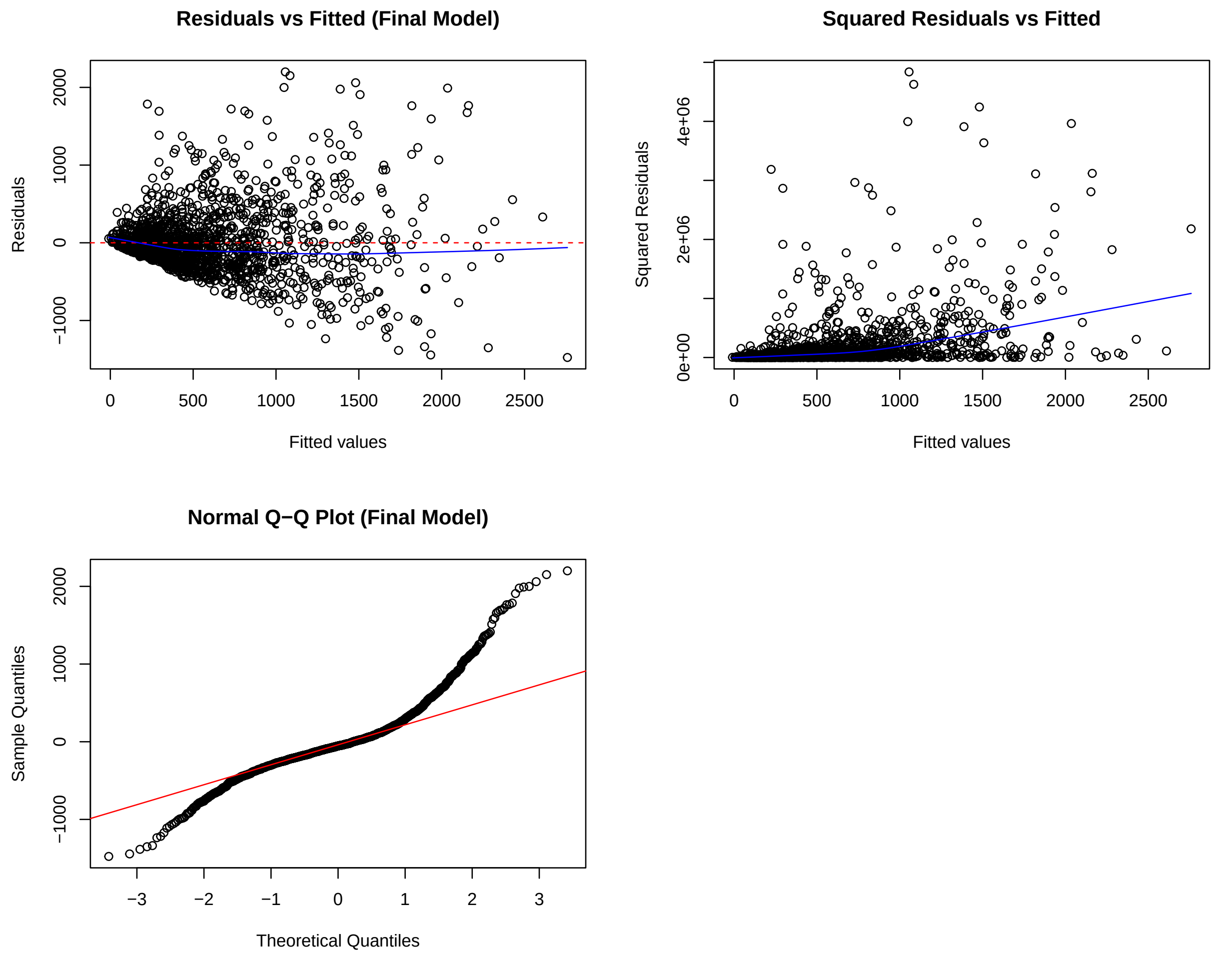
```
## Reduced model adjusted R²:   0.518
```

```r
# Update fit_final
fit_final <- fit_final_reduced
formula_final <- formula_final_reduced
best_predictors <- best_predictors_reduced
```

Assumption checks final model

```r
par(mfrow = c(2, 2))

#Residuals vs Fitted
plot(fit_final$fitted.values, fit_final$residuals,
     xlab = "Fitted values", ylab = "Residuals",
     main = "Residuals vs Fitted (Final Model)")
abline(h = 0, col = "red", lty = 2)
lines(lowess(fit_final$fitted.values, fit_final$residuals), col = "blue")

# Squared residuals vs Fitted
plot(fit_final$fitted.values, fit_final$residuals^2,
     xlab = "Fitted values", ylab = "Squared Residuals",
     main = "Squared Residuals vs Fitted")
lines(lowess(fit_final$fitted.values, fit_final$residuals^2), col = "blue")
```

```r
# QQ-plot
qqnorm(fit_final$residuals, main = "Normal Q-Q Plot (Final Model)")
qqline(fit_final$residuals, col = "red")

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



Robustness checks

For the final regression function, we included robust regression for outliers and heterosced-robust standard errors.

```r
library(sandwich)
library(lmtest)
library(MASS)

robust <- rlm(formula_final, data = train_data)
robust_se <- coeftest(robust, vcov = vcovHC(robust, type = "HC3"))
print(robust_se)
```

```
##
```

```
## z test of coefficients:
##
##                              Estimate Std. Error z value  Pr(>|z|)
## (Intercept)                 -85.38290   32.24328 -2.6481 0.0080949 **
## PctPopUnderPov               13.45909    3.74024  3.5985 0.0003201 ***
## PctNotHSGrad                  8.55936    1.69884  5.0384 4.696e-07 ***
## racepctblack                 22.06811    1.33874 16.4842 < 2.2e-16 ***
## PctImmig                     13.58370    1.61664  8.4024 < 2.2e-16 ***
## PctPopUnderPov:PctNotHSGrad  -0.15022    0.12426 -1.2089 0.2267177
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

#idk of die robustness nodig is of er een andere methode geprefereerd is, dit is comparison maar blijft er niet in

```r
# Compare OLS vs Robust coefficients
comparison <- data.frame(
  OLS = coef(fit_final),
  Robust = coef(robust),
  Difference = coef(fit_final) - coef(robust)
)
print(round(comparison, 4))
```

```
##                                   OLS   Robust Difference
## (Intercept)                 -133.7067 -85.3829   -48.3238
## PctPopUnderPov                17.6533  13.4591     4.1942
## PctNotHSGrad                  11.3813   8.5594     2.8220
## racepctblack                  22.4087  22.0681     0.3406
## PctImmig                      15.8791  13.5837     2.2954
## PctPopUnderPov:PctNotHSGrad   -0.2578  -0.1502    -0.1076
```

Outlier and Influence Diagnostics

We use several diagnostic measures to identify influential observations

```r
# Calculate diagnostics
stud_res_final <- rstudent(fit_final)
leverage <- hatvalues(fit_final)
p <- length(coef(fit_final))
n_train <- nrow(train_data)
leverage_threshold <- 2 * p / n_train
cooks_d <- cooks.distance(fit_final)
dffits_val <- dffits(fit_final)
dffits_threshold <- 2 * sqrt(p / n_train)
dfbetas_val <- dfbetas(fit_final)
dfbetas_threshold <- 2 / sqrt(n_train)

# dataframe
diagnostics <- data.frame(
  obs = 1:n_train,
  population = train_data$population,
  stud_residual = stud_res_final,
```

```r
  leverage = leverage,
  cooks_d = cooks_d,
  dffits = dffits_val
)

# Flag observations
diagnostics$outlier_residual <- abs(diagnostics$stud_residual) > 2
diagnostics$high_leverage <- diagnostics$leverage > leverage_threshold
diagnostics$high_cooks <- diagnostics$cooks_d > 4 / n_train
diagnostics$high_dffits <- abs(diagnostics$dffits) > dffits_threshold

# summ
cat("Outliers by studentized residuals (|r*| > 2):", sum(diagnostics$outlier_residual), "\n")
```

```
## Outliers by studentized residuals (|r*| > 2): 98
```

```r
cat("High leverage observations (h >", round(leverage_threshold, 4), "):",
    sum(diagnostics$high_leverage), "\n")
```

```
## High leverage observations (h > 0.0075 ): 151
```

```r
cat("High Cook's distance (D >", round(4/n_train, 4), "):",
    sum(diagnostics$high_cooks), "\n")
```

```
## High Cook's distance (D > 0.0025 ): 131
```

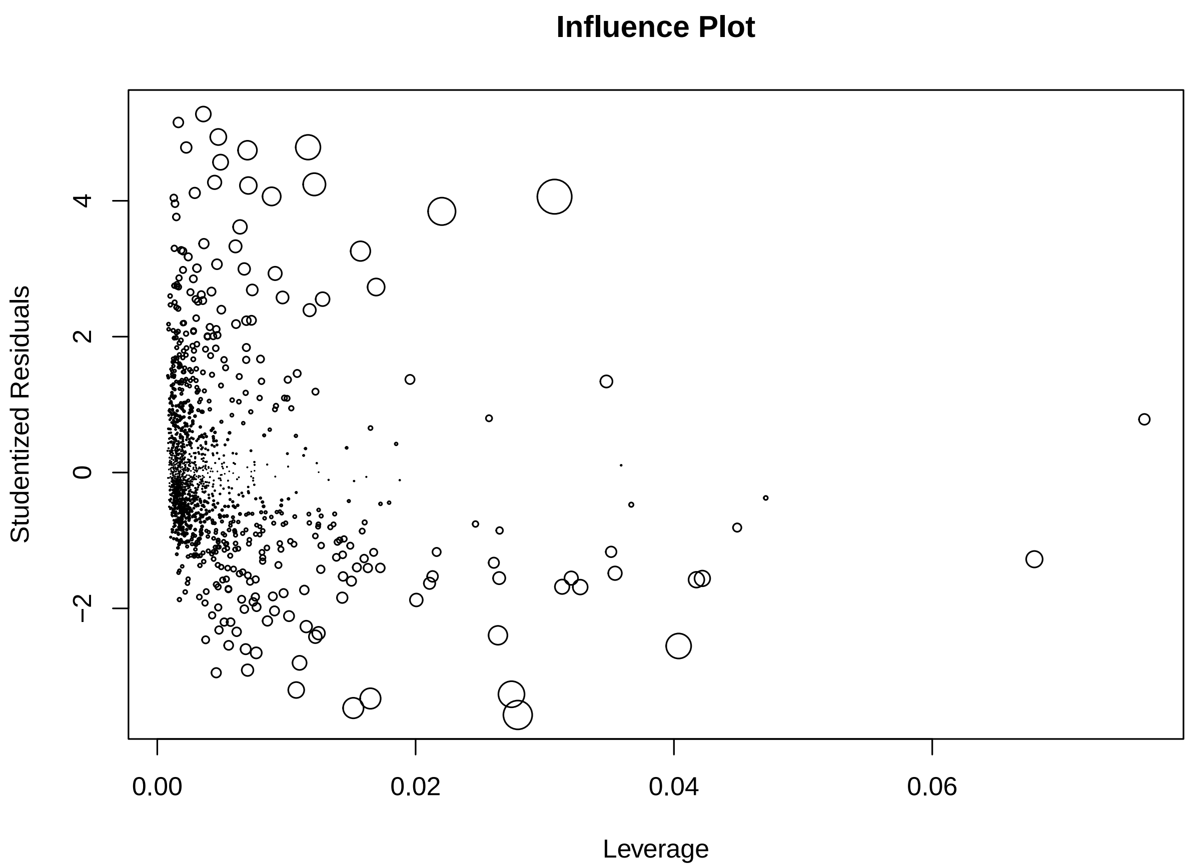
```r
cat("High DFFITS (|DFFITS| >", round(dffits_threshold, 4), "):",
    sum(diagnostics$high_dffits), "\n")
```

```
## High DFFITS (|DFFITS| > 0.1227 ): 131
```

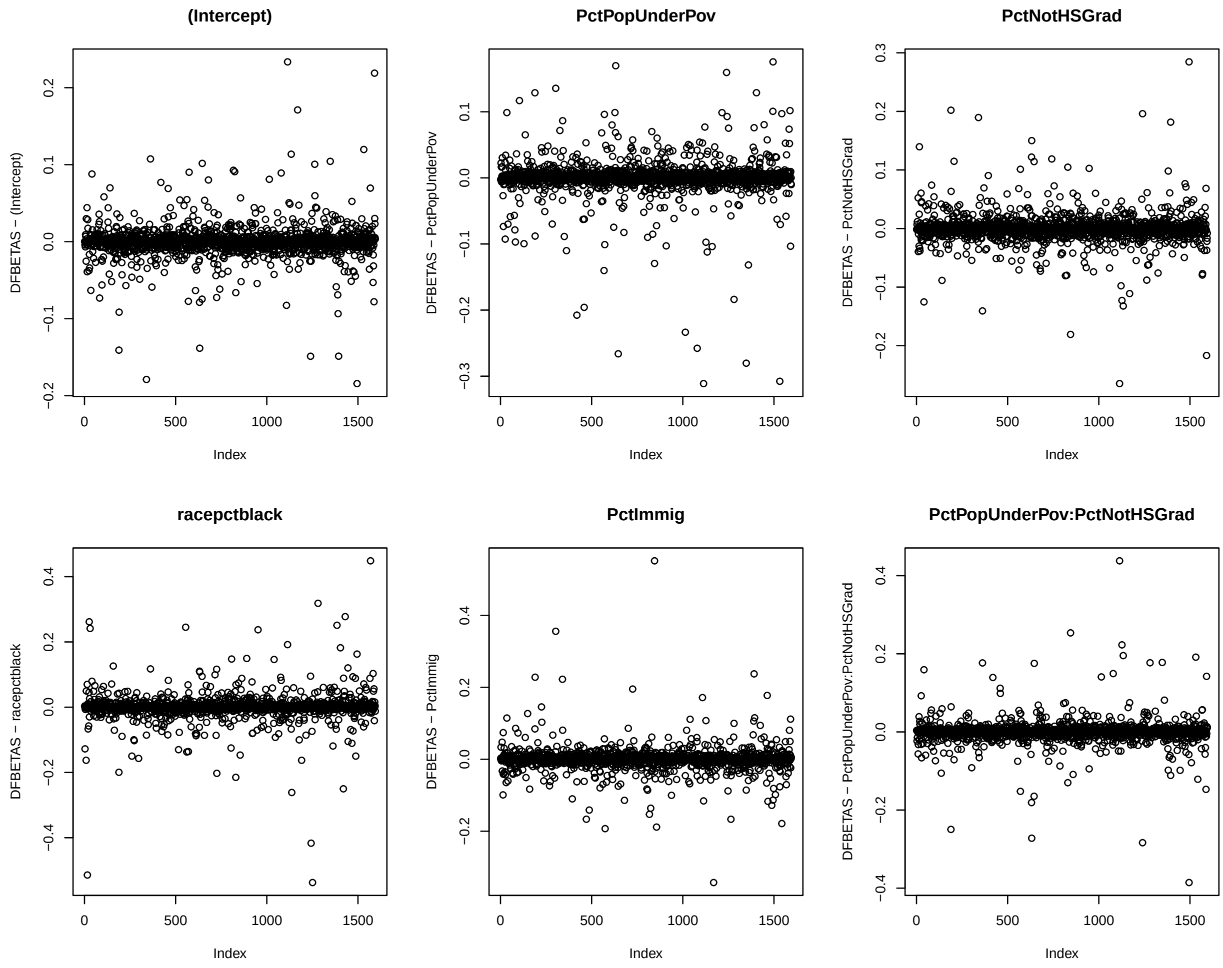
```r
# find influent obs
influential <- diagnostics[diagnostics$high_cooks | diagnostics$high_dffits, ]
influential <- influential[order(-influential$cooks_d), ]
print(head(influential[, c("obs", "population", "stud_residual", "leverage", "cooks_d", "dffits")], 10))
```

```
##       obs population stud_residual   leverage    cooks_d     dffits
## 845   845     358548      4.060766 0.03075769 0.08637199  0.7233836
## 16     16      12257     -3.568613 0.02791232 0.06049821 -0.6047065
## 1114 1114      87492      3.844158 0.02202982 0.05500311  0.5769576
## 1251 1251      15745     -3.263820 0.02742268 0.04975728 -0.5480486
## 1495 1495      12822     -2.553635 0.04036567 0.04555820 -0.5237356
## 1569 1569     394017      4.788167 0.01167515 0.04452441  0.5204160
## 1281 1281     275221      4.242108 0.01216030 0.03653000  0.4706637
## 1243 1243      11980     -3.326481 0.01650036 0.03074651 -0.4308685
## 189   189      18906     -3.466935 0.01517801 0.03066166 -0.4304019
## 340   340      12135      3.258989 0.01573267 0.02812434  0.4120289
```

```r
# Influence plot
plot(leverage, stud_res_final,
     xlab = "Leverage", ylab = "Studentized Residuals",
     main = "Influence Plot",
     cex = sqrt(cooks_d) * 10)
```




```r
# DFBETAS plots
par(mfrow = c(2, ceiling(ncol(dfbetas_val)/2)))
for(j in 1:ncol(dfbetas_val)) {
  plot(dfbetas_val[, j],
       ylab = paste("DFBETAS -", colnames(dfbetas_val)[j]),
       main = colnames(dfbetas_val)[j])
}
```




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

Summary

Dusja multivariate model stuk beter dan univariate model als je kijkt naar de tabel

```r
# summary

mse_final <- mean(fit_final$residuals^2)
summary_results <- data.frame(
  Model = c("Simple (PctPopUnderPov only)", "Final Multivariate"),
  R_squared = c(round(summary(fit)$r.squared, 4),
                round(summary(fit_final)$r.squared, 4)),
  Adj_R_squared = c(round(summary(fit)$adj.r.squared, 4),
                    round(summary(fit_final)$adj.r.squared, 4)),
  MSE = c(round(mse_simple, 2), round(mse_final, 2))
)

kable(summary_results, caption = "Comparison of Simple and Final Multivariate Models")
```

Table 2: Comparison of Simple and Final Multivariate Models

Model	R_squared	Adj_R_squared	MSE
Simple (PctPopUnderPov only)	0.2478	0.2473	276550.6
Final Multivariate	0.5195	0.5180	176662.0

Using the holdout set, we compute the Mean Squared Prediction Error (MSPR) and comparing it to the Mean Squared Error (MSE) from the training set.

```r
# Predictions on test set
pred <- predict(fit_final, newdata = test_data)

# MSPR
mspr <- mean((test_data$ViolentCrimesPerPop - pred)^2)

# MSE training
mse_final <- mean(fit_final$residuals^2)

cat("MSE (training set):", round(mse_final, 2), "\n")
```

```
## MSE (training set): 176662
```

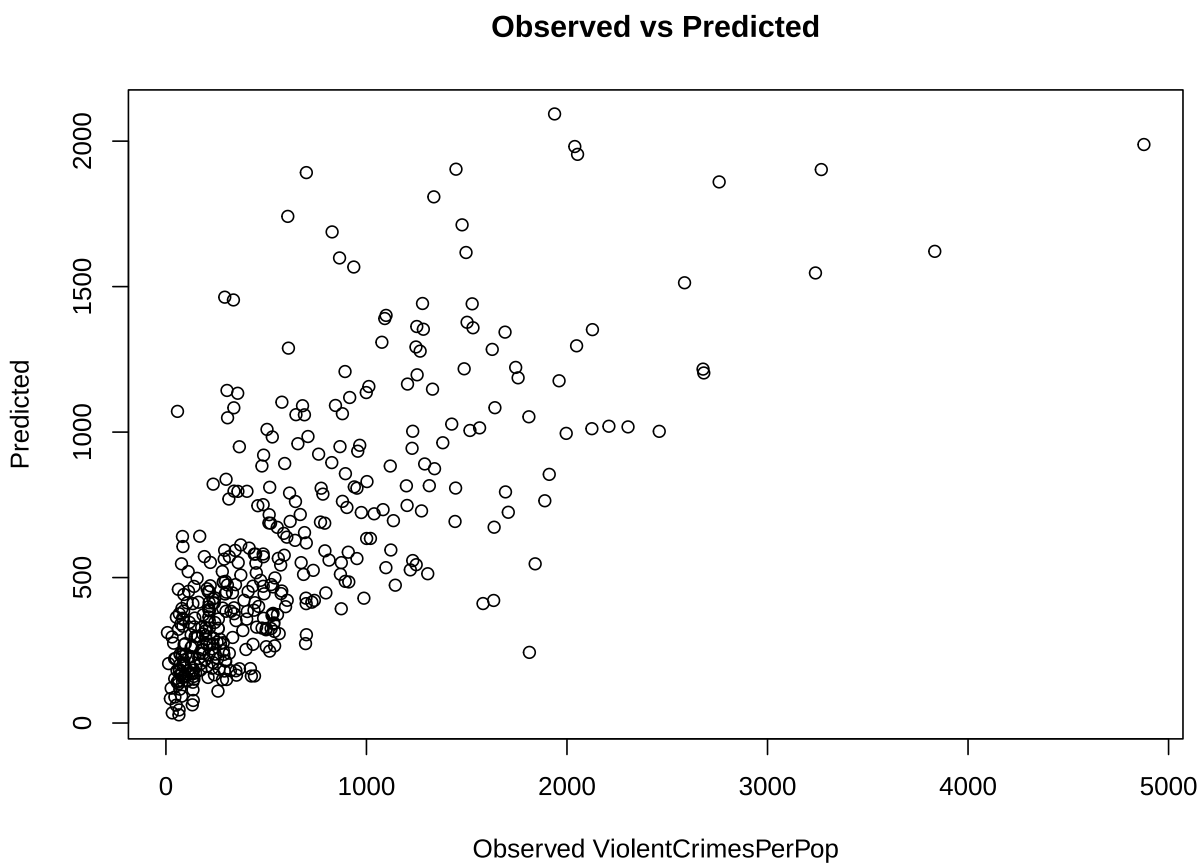
```r
cat("MSPR (test set):", round(mspr, 2), "\n")
```

```
## MSPR (test set): 200515.3
```

```r
cat("Ratio MSPR/MSE:", round(mspr/mse_final, 4), "\n")
```

```
## Ratio MSPR/MSE: 1.135
```

```r
# validation
plot(test_data$ViolentCrimesPerPop, pred,
     xlab = "Observed ViolentCrimesPerPop", ylab = "Predicted",
     main = "Observed vs Predicted")
```




```
# R-squared on test set
ss_res <- sum((test_data$ViolentCrimesPerPop - pred)^2)
ss_tot <- sum((test_data$ViolentCrimesPerPop - mean(test_data$ViolentCrimesPerPop))^2)
r2_test <- 1 - ss_res/ss_tot
cat("\nR² on test set:", round(r2_test, 4), "\n")
```

```
##
## R² on test set: 0.5183
```

```
cat("R² on training set:", round(summary(fit_final)$r.squared, 4), "\n")
```

```
## R² on training set: 0.5195
```

References

Becker GS (1968) Crime and Punishment: An Economic Approach. J Polit Econ 76: 169–217

References dataset

U. S. Department of Commerce, Bureau of the Census, Census Of Population And Housing 1990 United States: Summary Tape File 1a & 3a (Computer Files), U.S. Department Of Commerce, Bureau Of The Census Producer, Washington, DC and Inter-university Consortium for Political and Social Research Ann Arbor, Michigan. (1992)

U.S. Department of Justice, Bureau of Justice Statistics, Law Enforcement Management And Administrative Statistics (Computer File) U.S. Department Of Commerce, Bureau Of The Census Producer, Washington, DC and Inter-university Consortium for Political and Social Research Ann Arbor, Michigan. (1992)

U.S. Department of Justice, Federal Bureau of Investigation, Crime in the United States (Computer File) (1995)

Redmond, M. A. and A. Baveja: A Data-Driven Software Tool for Enabling Cooperative Information Sharing Among Police Departments. European Journal of Operational Research 141 (2002) 660-678.