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:

—- load data —-

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

—- remotely get variable names —-

```
library(dplyr)
library(stringr)
library(rvest)
url <- "https://archive.ics.uci.edu/dataset/211/communities+and+crime+unnormalized"

# Read the HTML page
page <- read_html(url)

# Extract all text from the page
text <- page %>% html_text()

# Split into lines
lines <- str_split(text, "\n")[[1]]

start <- grep("Additional Variable Information", lines, ignore.case = TRUE)
end   <- grep("Summary Statistics:", lines, ignore.case = TRUE)

# only retain the lines starting with --
```

```r
var_lines <- lines[str_starts(str_trim(lines), "--")]
# remove the --
var_names <- sapply(strsplit(var_lines, "--"), function(x) str_trim(x[2]))
# only retain the variable names by cutting everything after the :
var_names <- str_extract(var_names, "^[^:]+")
```

#remove in final product

```r
# variable_names <- list("communityname","state","countyCode", "communityCode", "fold", "population", "
#                        "racepctblack", "racePctWhite", "racePctAsian", "racePctHisp", "agePct12t21",
#                        "agePct65up", "numbUrban", "pctUrban", "medIncome", "pctWWage", "pctWFarmSelf"
#                        "pctWPubAsst", "pctWRetire", "medFamInc", "perCapInc", "whitePerCap", "blackPe
#                        "OtherPerCap", "HispPerCap", "NumUnderPov", "PctPopUnderPov", "PctLess9thGrade
#                        "PctUnemployed", "PctEmploy", "PctEmplManu", "PctEmplProfServ", "PctOccupManu"
#                        "MalePctNevMarr", "FemalePctDiv", "TotalPctDiv", "PersPerFam", "PctFam2Par", "
#                        "PctTeen2Par", "PctWorkMomYoungKids", "PctWorkMom", "NumKidsBornNeverMar", "Pc
#                        "PctImmigRecent", "PctImmigRec5", "PctImmigRec8", "PctImmigRec10", "PctRecentI
#                        "PctRecImmig10", "PctSpeakEnglOnly", "PctNotSpeakEnglWell", "PctLargHouseFam",
#                        "PersPerOwnOccHous", "PersPerRentOccHous", "PctPersOwnOccup", "PctPersDenseHou
#                        "HousVacant", "PctHousOccup", "PctHousOwnOcc", "PctVacantBoarded", "PctVacMore
#                        "PctHousNoPhone", "PctWOFullPlumb", "OwnOccLowQuart", "OwnOccMedVal", "OwnOccH
#                        "RentMedian", "RentHighQ", "RentQrange", "MedRent", "MedRentPctHousInc", "MedO
#                        "NumInShelters", "NumStreet", "PctForeignBorn", "PctBornSameState", "PctSameHo
#                        "PctSameState85", "LemasSwornFT", "LemasSwFTPerPop", "LemasSwFTFieldOps", "Lem
#                        "LemasTotReqPerPop", "PolicReqPerOffic", "PolicPerPop", "RacialMatchCommPol",
#                        "PctPolicHisp", "PctPolicAsian", "PctPolicMinor", "OfficAssgnDrugUnits", "NumK
#                        "LandArea", "PopDens", "PctUsePubTrans", "PolicCars", "PolicOperBudg", "LemasP
#                        "LemasPctOfficDrugUn", "PolicBudgPerPop", "murders", "murdPerPop", "rapes", "r
#                        "assaults", "assaultPerPop", "burglaries", "burglPerPop", "larcenies", "larcPe
#                        "arsons", "arsonsPerPop", "ViolentCrimesPerPop", "nonViolPerPop")
```

end of removable part

```r
colnames(violent_crimes_table) <- var_names
```

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

Design

The dataset combines 1990 U.S. Census socio-economic data, 1990 law enforcement data from the Law Enforcement Management and Admin Stats (LEMAS) survey, and 1995 FBI crime data, thereby creating two cohorts. For the FBI crime data, it is mentioned that states with a lower amount of visitors have a lower per capita crime rate and vice verse. The LEMAS survey covers all communities with police departments of at least 100 officers and a random sample of smaller departments. If communities were absent from either the

crime or census datasets (e.g., those with very small departments), then they were removed. All demographic data is from 1990, but per-capita crime rates use 1995 population counts. Finally, rape counts, a component of violent crime, are missing in some states due to inconsistent reporting, which resulted in missing total violent crime values for those states. We will investigate whether this missingness has probably a large effect on the model and if necessary use imputations.

Data preparation

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%). It's important to mention that this is not an exact transformation, because all demographic data is from 1990, but per-capita crime rates use 1995 population counts.

```
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
```

The only variable for which NA values are found is the outcome variable *ViolentCrimesPerPop*. Imputations can not help a lot to handle this. As already mentioned in the section on the study design, many of these NA values are due to the fact that rape counts, a component of violent crime, were not included in the statistics for several states. 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.

```
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
```

```
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
```

Univariate descriptives

After removing NA values from the database univariate descriptives are calculated, both the missing values and the non-missing values.

```
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 highsch
##   .. ..- 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 degr
##   .. ..- 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>
```

```
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
```

```
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
```

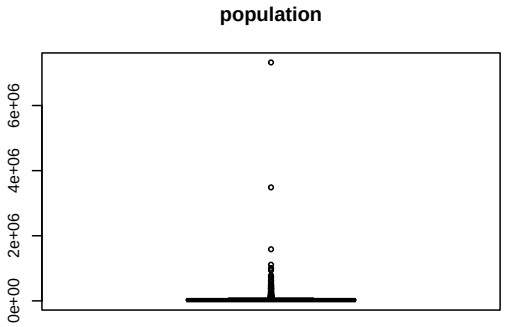
```
na_subset %>%
  pivot_longer(cols = where(is.numeric), names_to = "variable", values_to = "value") %>%
  group_by(variable) %>%
  summarise(
    min = min(value, na.rm = TRUE),
    q25 = quantile(value, 0.25, na.rm = TRUE),
    mean = mean(value, na.rm = TRUE),
    sd = sd(value, na.rm = TRUE),
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

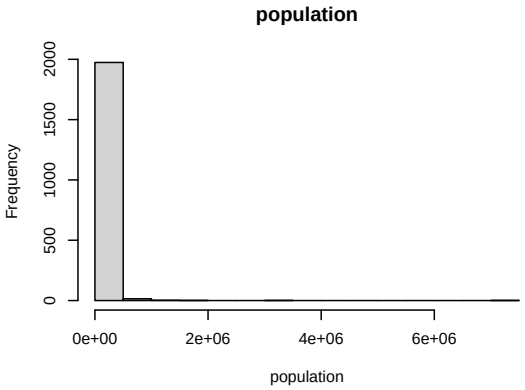
```
## # A tibble: 10 x 9
##    variable              min      q25     mean      sd    q75    max     n   NAs
##    <chr>               <dbl>    <dbl>    <dbl>   <dbl>  <dbl>  <dbl> <int> <int>
##  1 PctBSorMore          3.27     14.2     23.6   1.42e1 3.09e1 7.92e1   221     0
##  2 PctEmploy           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
```

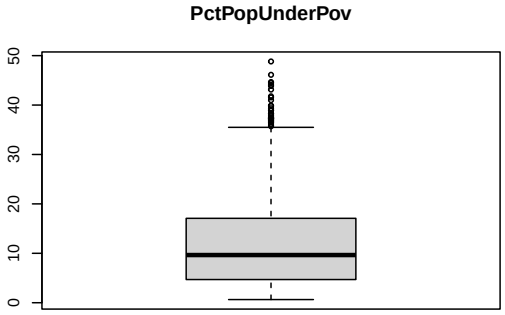
To gain insight into the univariate distributions, boxplots and histograms are generated.

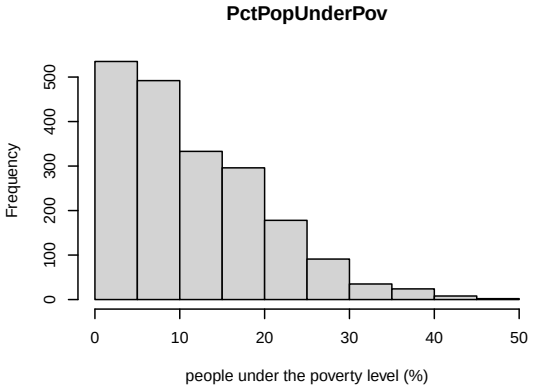
```
numeric_cols <- sapply(crimes_table_subset, is.numeric)
crimes_table_subset_num <- crimes_table_subset[, ..numeric_cols]
par(mfrow = c(4,2))
for(columnname in names(crimes_table_subset_num)){
  column <- crimes_table_subset_num[[columnname]]
  boxplot(column,
          main = columnname
        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

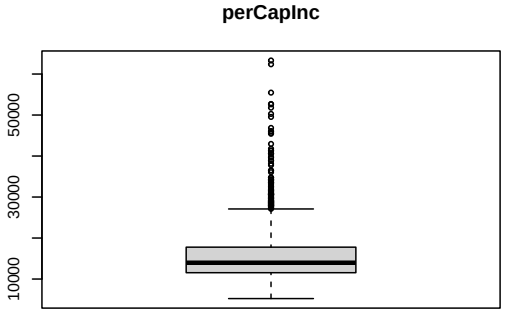
```
        main = columnname,
        xlab = get_label(column)
           )
}
```

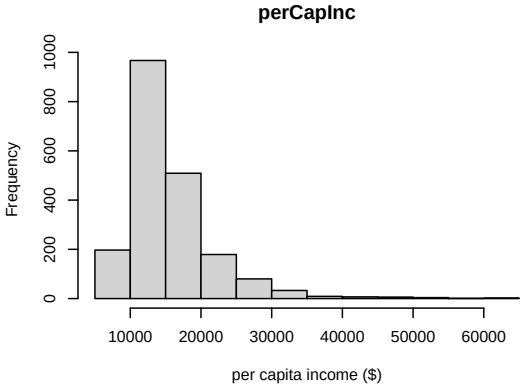

population
0e+00
2e+06
4e+06
6e+06

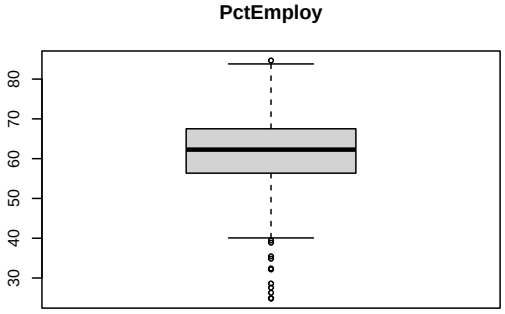

population
Frequency
0
500
1000
1500
2000
0e+00
2e+06
4e+06
6e+06
population

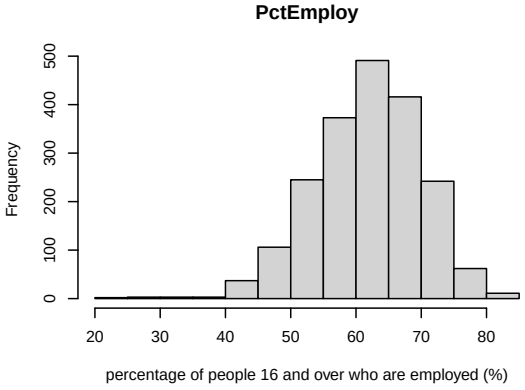

PctPopUnderPov
0
10
20
30
40
50

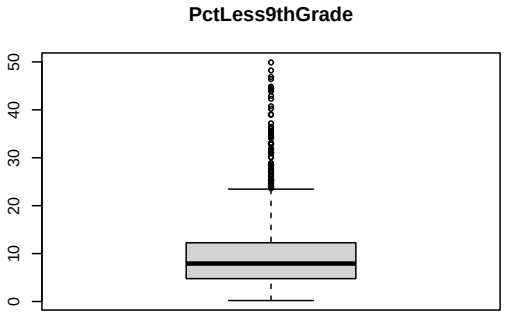

PctPopUnderPov
Frequency
0
100
200
300
400
500
0
10
20
30
40
50
people under the poverty level (%)

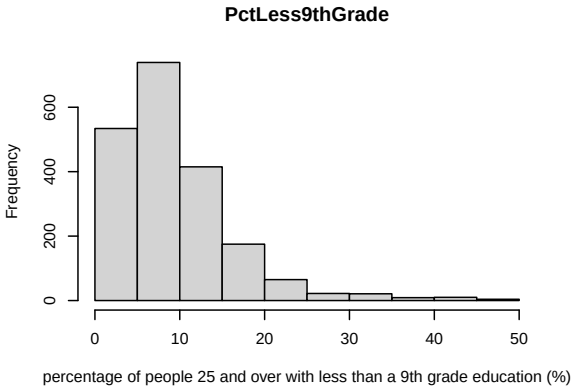

perCapInc
10000
30000
50000

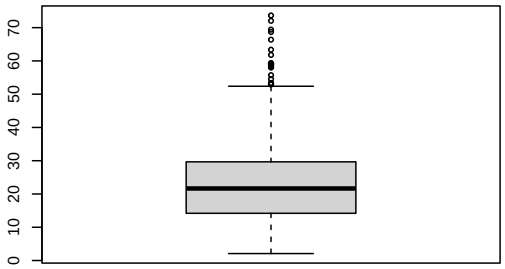

perCapInc
Frequency
0
200
400
600
800
1000
10000
20000
30000
40000
50000
60000
per capita income ($)

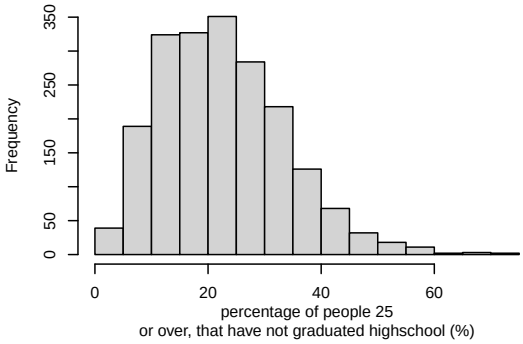

PctEmploy
30
40
50
60
70
80

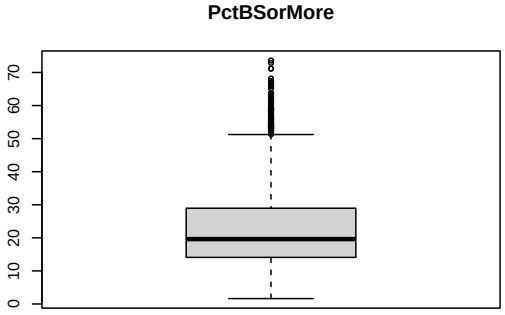

PctEmploy
Frequency
0
100
200
300
400
500
20
30
40
50
60
70
80
percentage of people 16 and over who are employed (%)

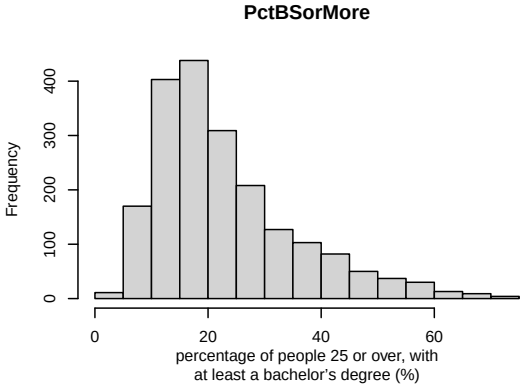

PctLess9thGrade
0
10
20
30
40
50

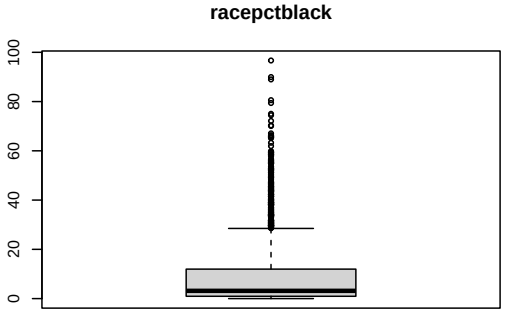

PctLess9thGrade
Frequency
0
200
400
600
0
10
20
30
40
50
percentage of people 25 and over with less than a 9th grade education (%)

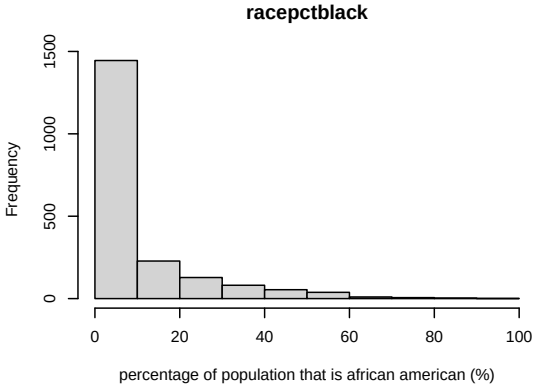

PctNotHSGrad
0
10
20
30
40
50
60
70

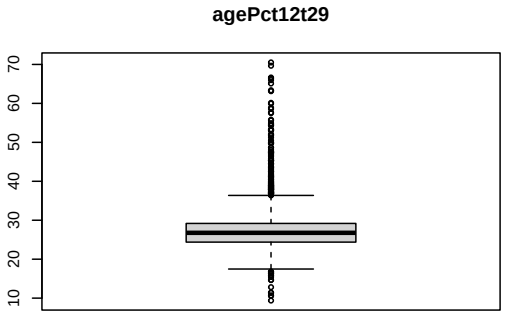

PctNotHSGrad
Frequency
0
50
150
250
350
0
20
40
60
percentage of people 25
or over, that have not graduated highschool (%)

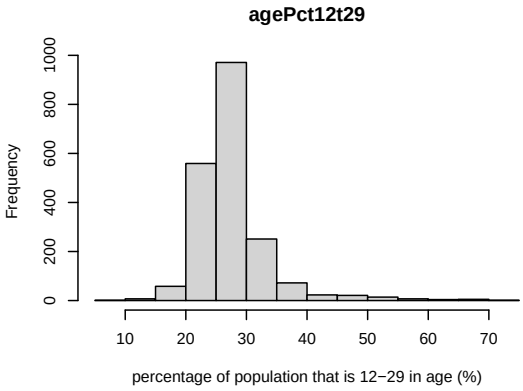

PctBSorMore
0
10
20
30
40
50
60
70

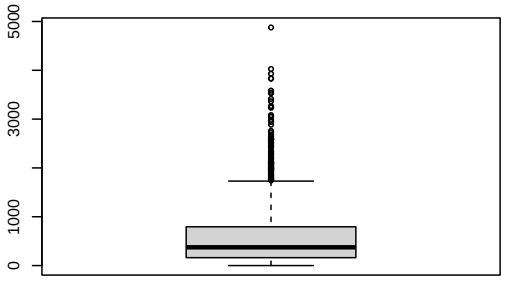

PctBSorMore
Frequency
0
100
200
300
400
0
20
40
60
percentage of people 25 or over, with
at least a bachelor's degree (%)

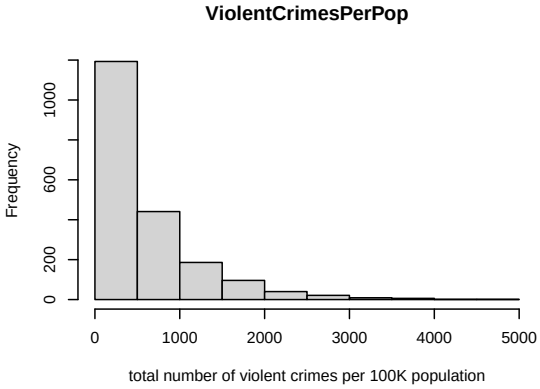

racepctblack
0
20
40
60
80
100

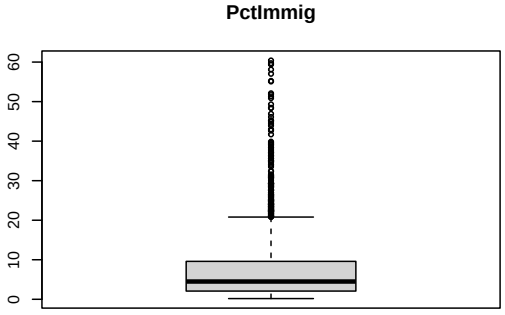

racepctblack
Frequency
0
500
1000
1500
0
20
40
60
80
100
percentage of population that is african american (%)

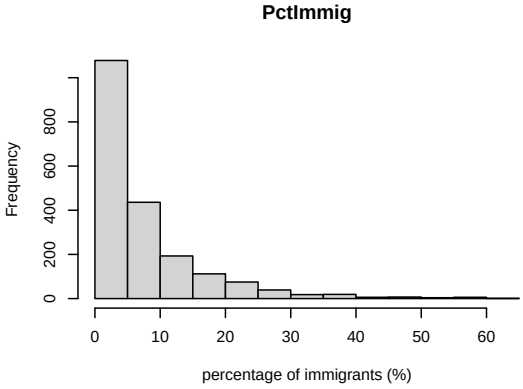

agePct12t29
10
20
30
40
50
60
70


agePct12t29
Frequency
0
200
400
600
800
1000
10
20
30
40
50
60
70
percentage of population that is 12−29 in age (%)


ViolentCrimesPerPop
0
1000
3000
5000

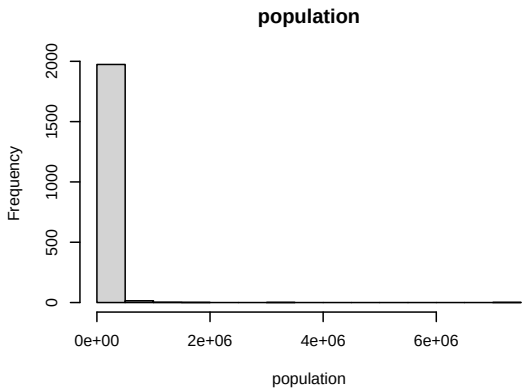

ViolentCrimesPerPop
Frequency
0
200
600
1000
0
1000
2000
3000
4000
5000
total number of violent crimes per 100K population

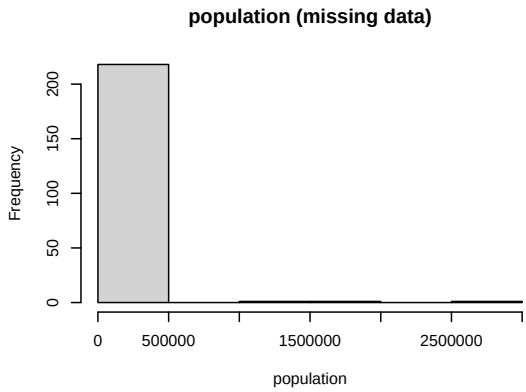

PctImmig
0
10
20
30
40
50
60

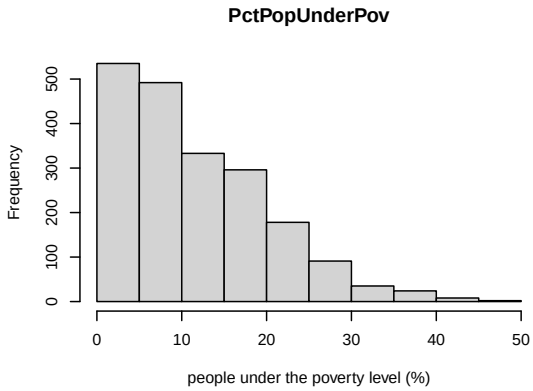

PctImmig
Frequency
0
200
400
600
800
0
10
20
30
40
50
60
percentage of immigrants (%)

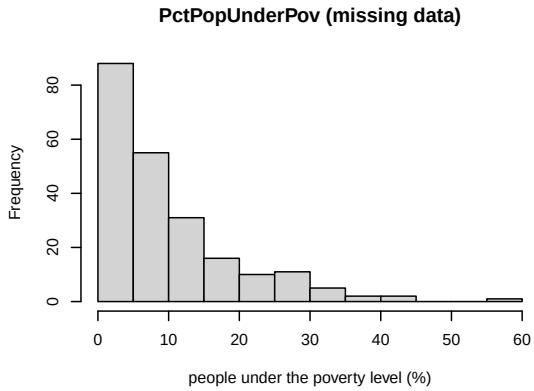
To investigate the impact of the missing values, the distribution of the other variables is compared in the missing data vs. the non-missing data. This is done using the histograms. Neither the summary statistics nor the scatter plots indicate that the missing data have characteristics that differ substantially from the non-missing data

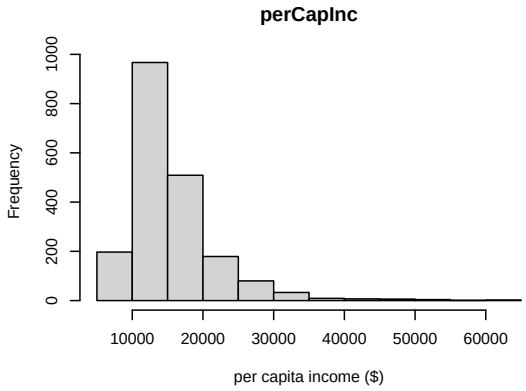
```
numeric_cols_na <- sapply(na_subset, is.numeric)
crimes_table_subset_num_na <- na_subset[, ..numeric_cols_na]
par(mfrow = c(4,2))
for(columnname in names(crimes_table_subset_num_na)){
  column <- crimes_table_subset_num[[columnname]]
  column_na <- crimes_table_subset_num_na[[columnname]]
  hist(column,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

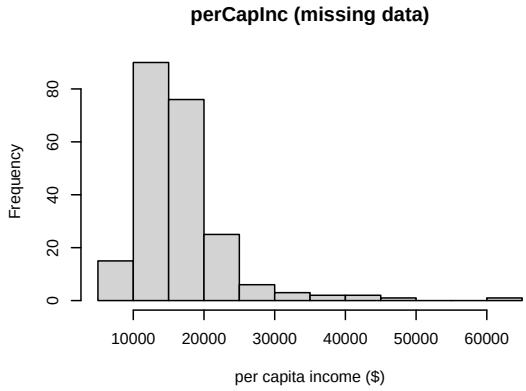

population
Frequency
0
500
1000
1500
2000
0e+00
2e+06
4e+06
6e+06
population

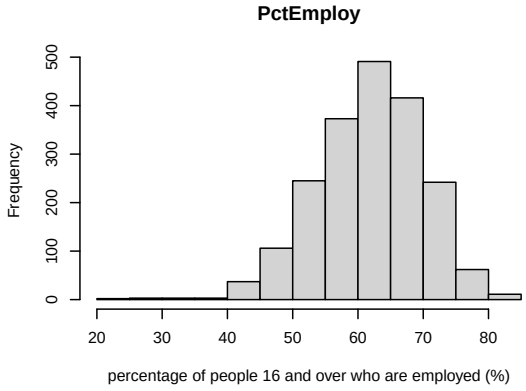

population (missing data)
Frequency
0
50
100
150
200
0
500000
1500000
2500000
population

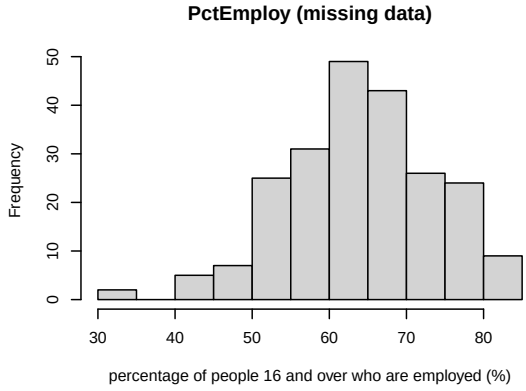

PctPopUnderPov
Frequency
0
100
200
300
400
500
0
10
20
30
40
50
people under the poverty level (%)

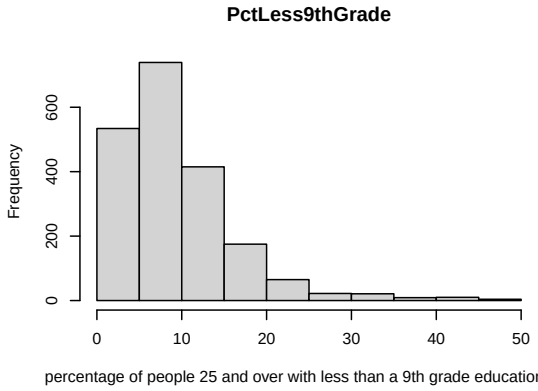

PctPopUnderPov (missing data)
Frequency
0
20
40
60
80
0
10
20
30
40
50
60
people under the poverty level (%)

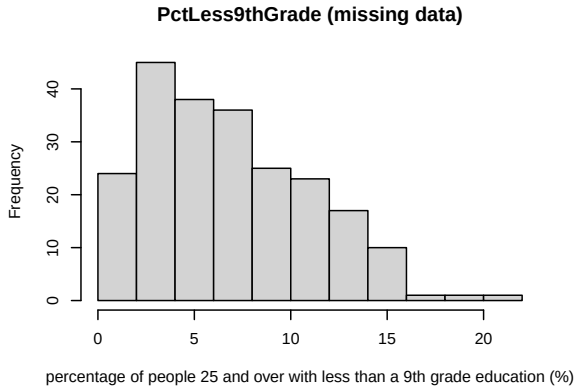

perCapInc
Frequency
0
200
400
600
800
1000
10000
20000
30000
40000
50000
60000
per capita income ($)

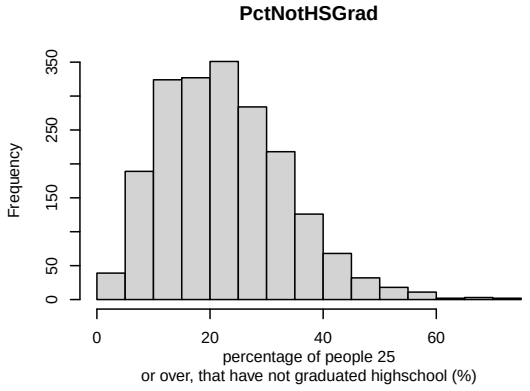

perCapInc (missing data)
Frequency
0
20
40
60
80
10000
20000
30000
40000
50000
60000
per capita income ($)

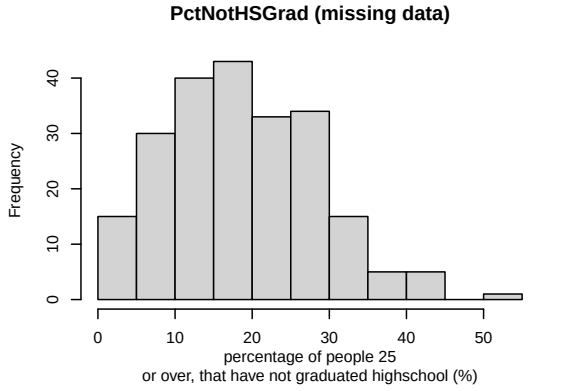

PctEmploy
Frequency
0
100
200
300
400
500
20
30
40
50
60
70
80
percentage of people 16 and over who are employed (%)

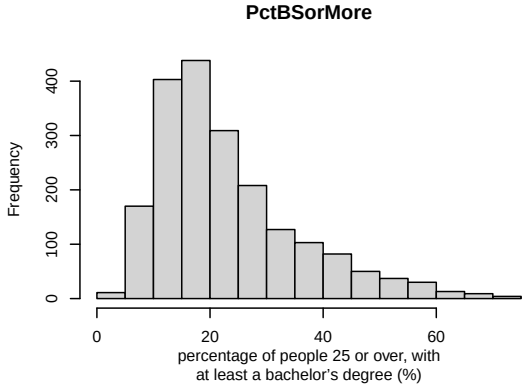

PctEmploy (missing data)
Frequency
0
10
20
30
40
50
30
40
50
60
70
80
percentage of people 16 and over who are employed (%)

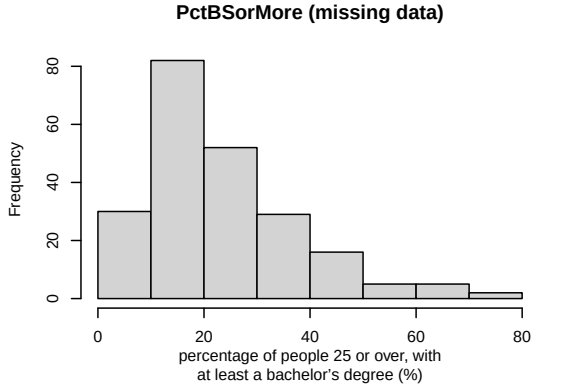

PctLess9thGrade
Frequency
0
200
400
600
0
10
20
30
40
50
percentage of people 25 and over with less than a 9th grade education (%)

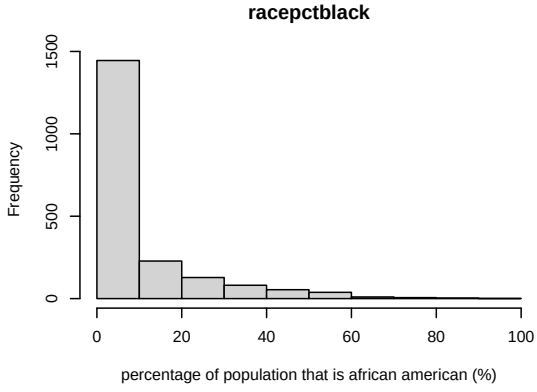

PctLess9thGrade (missing data)
Frequency
0
10
20
30
40
0
5
10
15
20
percentage of people 25 and over with less than a 9th grade education (%)

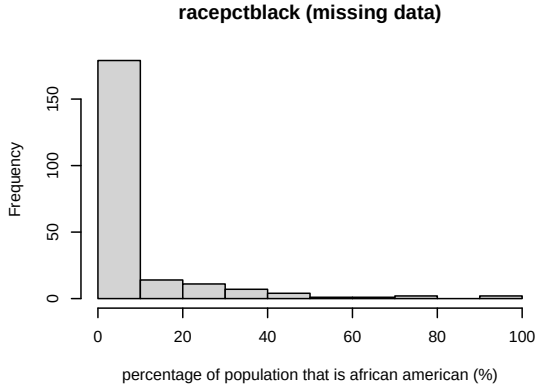

PctNotHSGrad
Frequency
0
50
150
250
350
0
20
40
60
percentage of people 25
or over, that have not graduated highschool (%)

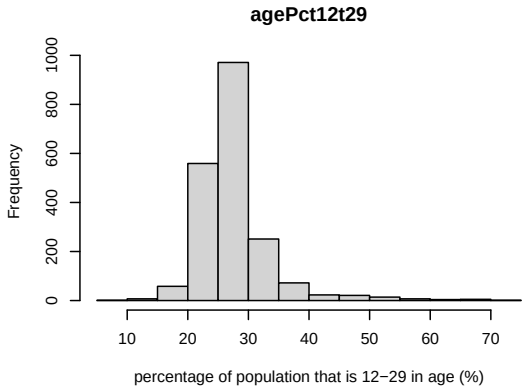

PctNotHSGrad (missing data)
Frequency
0
10
20
30
40
0
10
20
30
40
50
percentage of people 25
or over, that have not graduated highschool (%)

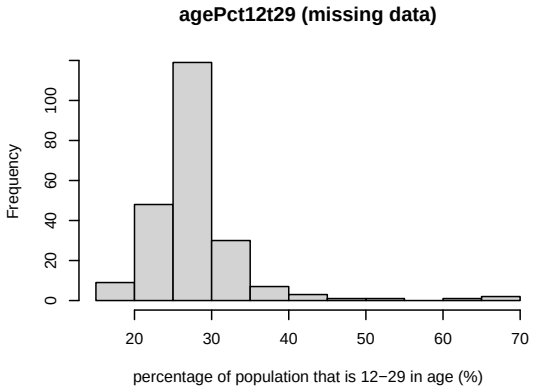

PctBSorMore
Frequency
0
100
200
300
400
0
20
40
60
percentage of people 25 or over, with
at least a bachelor's degree (%)

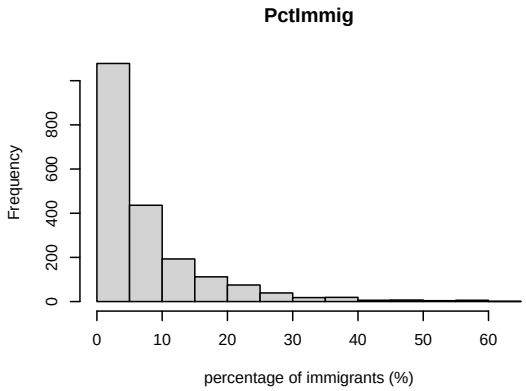

PctBSorMore (missing data)
Frequency
0
20
40
60
80
0
20
40
60
80
percentage of people 25 or over, with
at least a bachelor's degree (%)

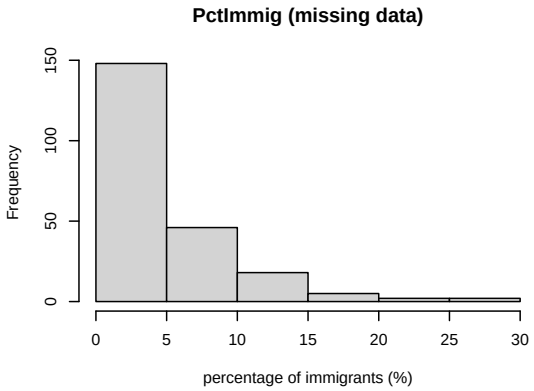

racepctblack
Frequency
0
500
1000
1500
0
20
40
60
80
100
percentage of population that is african american (%)


racepctblack (missing data)
Frequency
0
50
100
150
0
20
40
60
80
100
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 (%)

Multivariate descriptives

After investigating the univariate descriptives, multivariate descriptives are calculated.

Multivariate descriptives

After investigating the univariate descriptives, multivariate descriptives are calculated.

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.

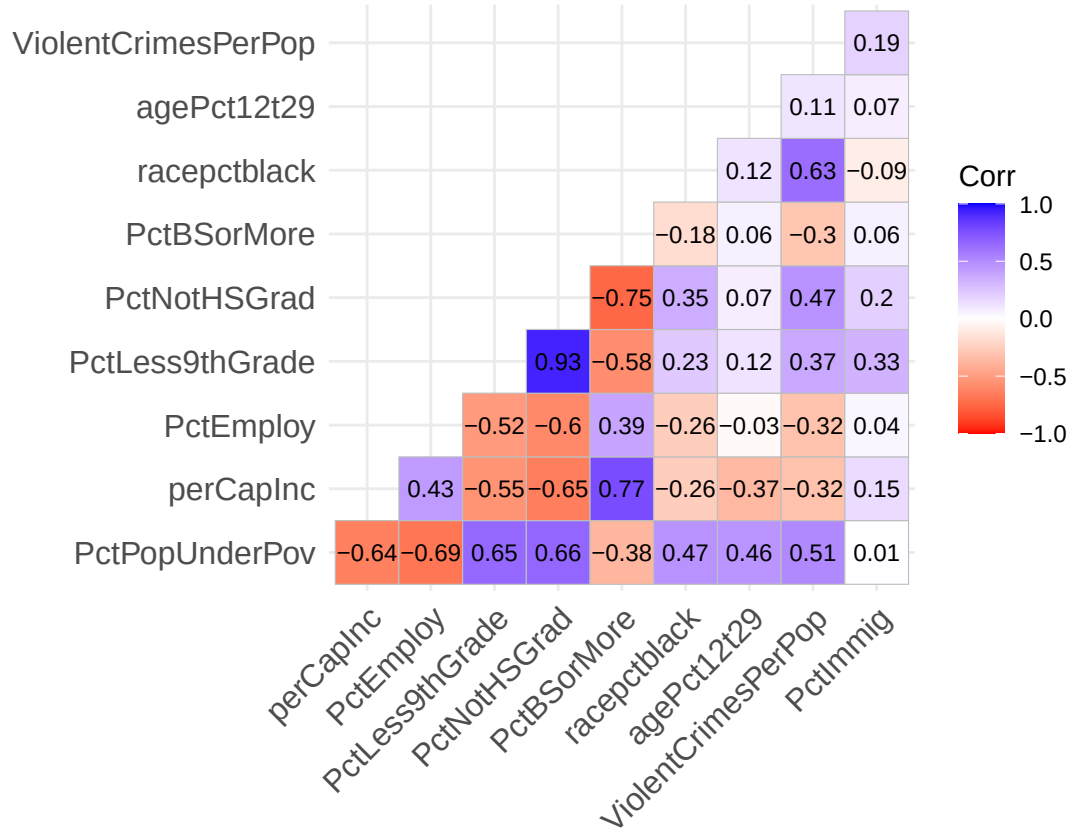
- *PctNotHSGrad* and *PctLess9thGrade* ($r = 0.93$)
- *perCapInc* and *PctBSorMore* ($r = 0.77$)
- *PctNotHSGrad* and *PctBSorMore* ($r = -0.75$)

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 that are really low.

```
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)

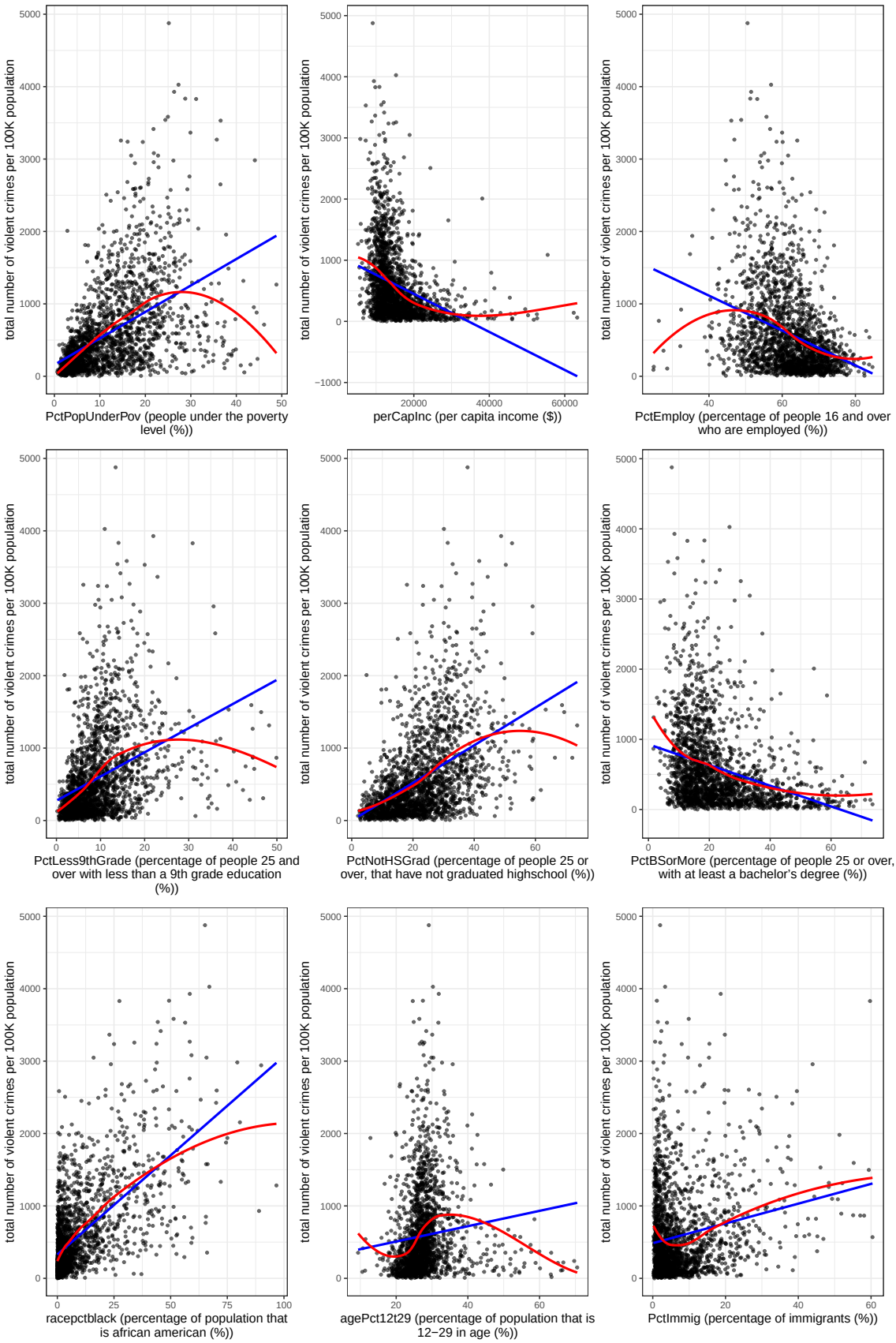
Other variables show a somewhat linear pattern, although this trend is often distorted in the extreme regions of the x-axis.

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:

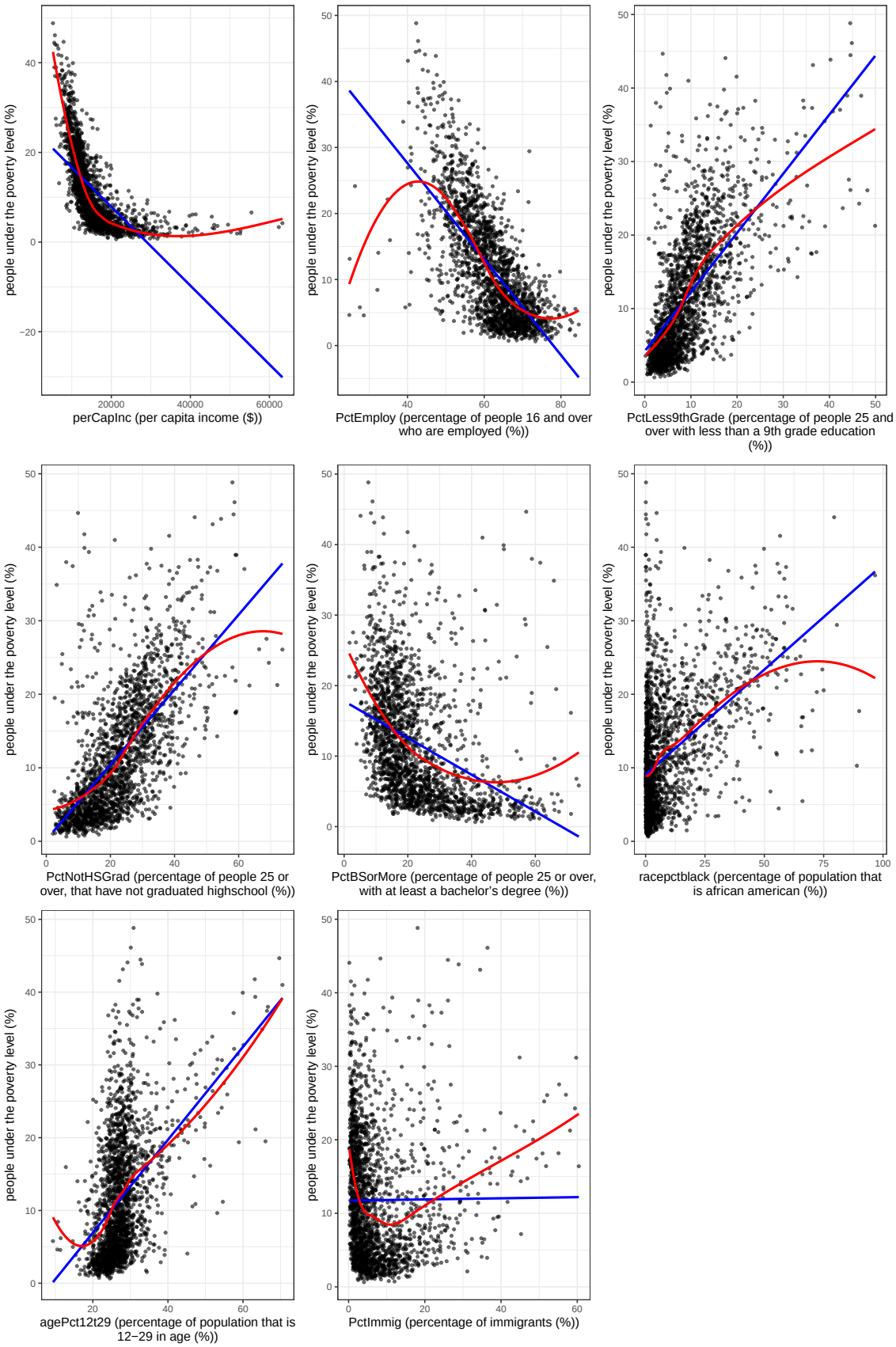
- *PctEmploy*
- *PctLess9thGrade*
- *PctNotHSGrad*

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_v

library(ggplot2)
library(patchwork)
df <- crimes_table_subset_num[,-c("population")]
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("population", "ViolentCrimesPerPop")]
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. For now the datapoint is kept for further analysis.

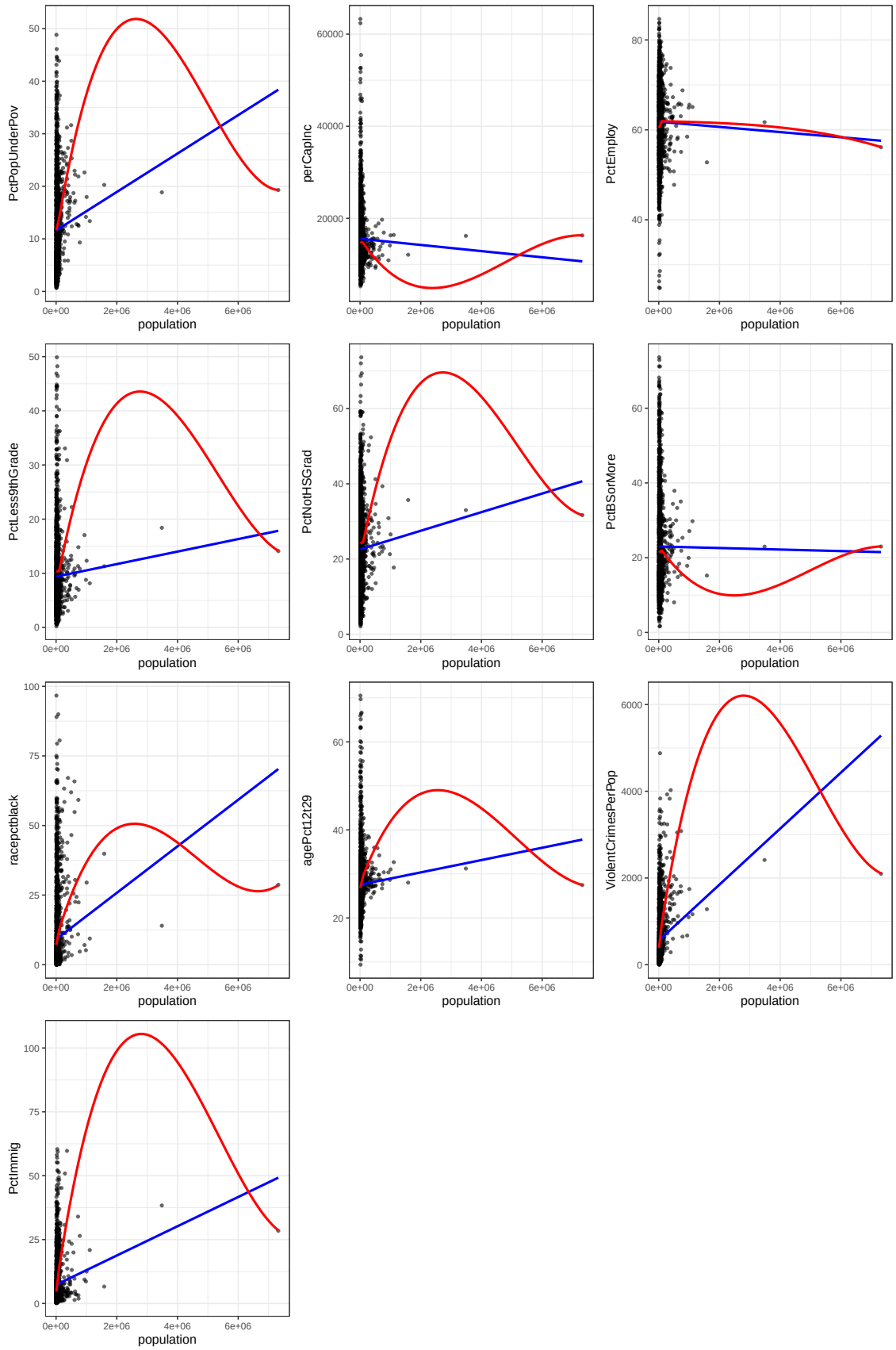
```
crimes_table_subset[order(crimes_table_subset_num$ViolentCrimesPerPop, decreasing=TRUE), , drop = FALSE]
```

```
##           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
```

In the protocol it's stated that small communities may have a higher probability to have more extreme values of the predictor and response variables. To investigate this scatter plots are created with *population* as x variable. It is clear that communities with a very small population show a very large spread for all variables.

```
df <- crimes_table_subset_num
x_var <- "population"
y_vars <- setdiff(colnames(df), x_var)
plots <- lapply(y_vars, function(y_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(y = str_wrap(y_var, width = 45))
}
)

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


PctPopUnderPov
perCapInc
PctEmploy
PctLess9thGrade
PctNotHSGrad
PctBSorMore
racepctblack
agePct12t29
ViolentCrimesPerPop
PctImmig
population
0e+00
2e+06
4e+06
6e+06

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.

```
set.seed(123)
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, ]
n_training <- nrow(train_data)

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

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

```
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:

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

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

```
##
## Call:
## lm(formula = ViolentCrimesPerPop ~ PctPopUnderPov, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1644.32  -249.82   -98.62   161.34  2801.61
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     150.612     21.665   6.952 5.24e-12 ***
## PctPopUnderPov   37.047      1.501  24.674  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 508.2 on 1593 degrees of freedom
## Multiple R-squared:  0.2765, Adjusted R-squared:  0.2761
## F-statistic: 608.8 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 = 150.61 + 37.05 * PctPopUnderPov
```

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

```
## R-squared: 0.2765
```

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

```
## Adjusted R-squared: 0.2761
```

```r
cat("BIC:", BIC(fit), "\n")
```

```
## BIC: 24423.2
```

```r
# MSE
sse_simple <- sum(fit$residuals^2)
mse_simple <- sse_simple/(n_training - 2)
cat("SSE:", round(sse_simple, 2), "\n")
```

```
## SSE: 411464269
```

```r
# Calculation of BIC-waarde in R: -2 * as.numeric(logLik(fit)) + attr(logLik(fit), "df") * log(n)
# Not the one from the course slides n*log(sse_simple) - n*log(n) + p*log(n), but ok?
cat("MSE:", round(mse_simple, 2), "\n")
```

```
## MSE: 258295.2
```

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

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

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

```
##                    2.5 %    97.5 %
## (Intercept)    108.11719 193.10635
## PctPopUnderPov  34.10175  39.99172
```

```r
# ANOVA
anova(fit)
```

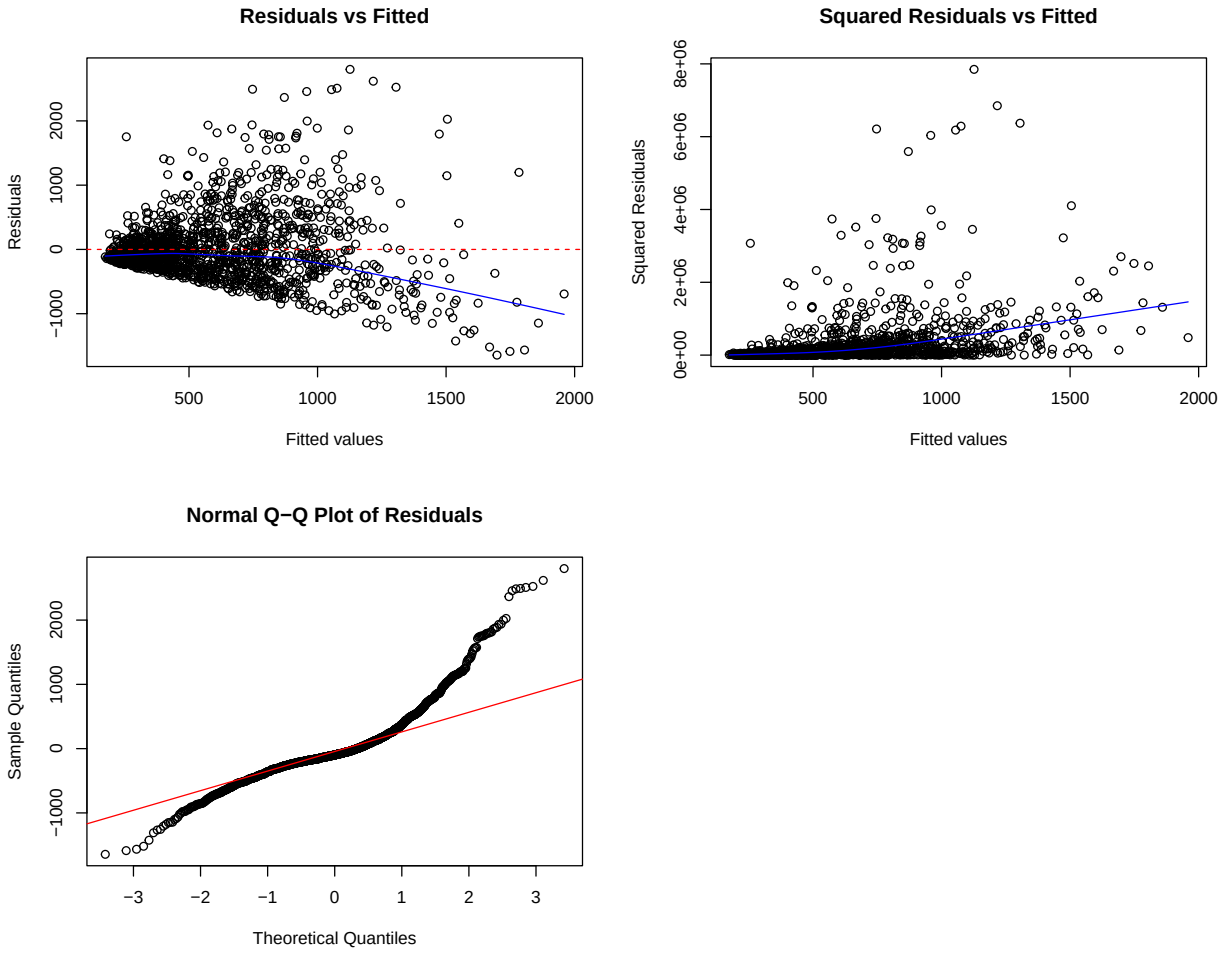
```
## Analysis of Variance Table
##
## Response: ViolentCrimesPerPop
##                   Df    Sum Sq   Mean Sq F value    Pr(>F)
## PctPopUnderPov     1 157255778 157255778  608.82 < 2.2e-16 ***
## Residuals       1593 411464269    258295
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

Assumption checks

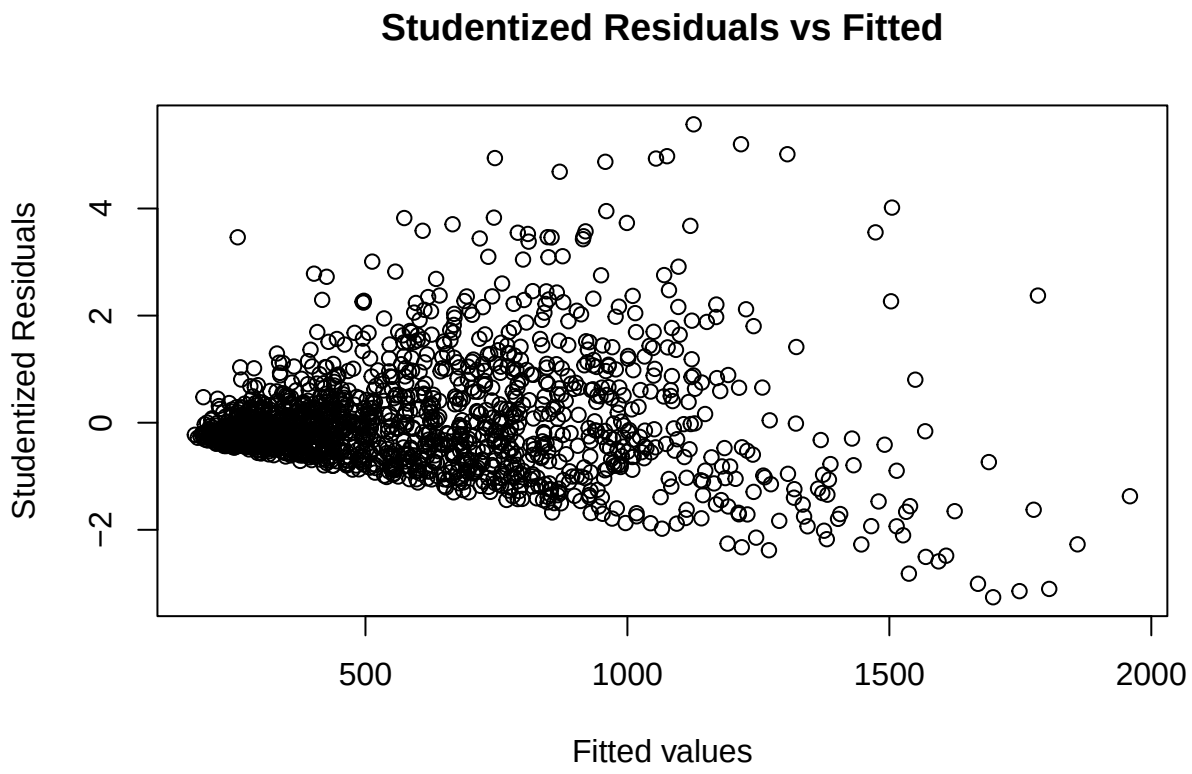
We check assumptions linearity, independence of errors, homoscedasticity, and normality of errors. It is clear these assumptions are violated. In the next step, we extend the model by adding relevant predictors and reassess the assumptions.

```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")

# 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 <dit zijn de deleted toch?> –> moet dit apart?

```
# 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?

  outlier_data <- train_data[outliers_simple, .(population, ViolentCrimesPerPop, PctPopUnderPov)]
  print(outlier_data)
```

```
##     population ViolentCrimesPerPop PctPopUnderPov
##          <int>               <num>          <num>
##  1:      19378             2008.66           2.85
##  2:      95706             2299.87          29.04
##  3:     109602             2078.85          16.47
##  4:      30996             1938.43          15.96
##  5:      11751             1581.25           7.20
##  6:      10014               47.11          28.09
##  7:     435146             2460.11          15.32
##  8:      22906             2193.23          25.55
##  9:      23451             2344.56          17.55
## 10:      45206             2678.55          20.67
## 11:      34311             2982.31          44.08
## 12:     219531             2978.69          26.17
## 13:      10404             1759.00          14.79
## 14:      86905             1651.73           9.35
```

```
## 15:      16799           2052.79         23.33
## 16:      98052           2109.96         21.17
## 17:      16491           1964.10         20.32
## 18:      43467           3414.57         21.79
## 19:      12001           2451.06         19.59
## 20:     265968           2466.68         24.82
## 21:     574283           1968.89         18.71
## 22:      36118           2507.51         11.43
## 23:      16027           1730.67         12.03
## 24:      26623           3540.57         24.40
## 25:      45549           2344.68         21.57
## 26:      15854            353.53         33.06
## 27:      25158           2605.96         19.02
## 28:      21080           2586.29         17.28
## 29:      41194           1700.70         14.00
## 30:      61018           2423.47         12.38
## 31:      50961           1846.45         13.25
## 32:      27334           1726.58         14.93
## 33:      23755           3268.26         35.71
## 34:      14903           2018.97         19.62
## 35:     723959           1810.07         12.66
## 36:     437319           1841.38         14.53
## 37:      13051             63.81         30.22
## 38:      37986           3583.48         24.97
## 39:      29925           2594.03         17.80
## 40:      28743            461.98         37.13
## 41:      30326           1884.24         18.65
## 42:     672971           1681.08         12.81
## 43:    3485398           2414.77         18.86
## 44:      10398            278.92         33.20
## 45:      12915            150.31         40.99
## 46:      61921           2082.86         22.49
## 47:     226505           1636.75         11.94
## 48:      21265            239.75         44.66
## 49:      71349           2541.38         13.92
## 50:    7322564           2097.71         19.29
## 51:      34590            112.41         37.43
## 52:      12822            161.69         43.13
## 53:     139739           2288.32         27.51
## 54:     280015           3235.45         19.44
## 55:      40949           1812.62          6.78
## 56:     164693           2304.49         15.76
## 57:      30705            302.47         38.30
## 58:      14302             53.73         41.77
## 59:      20807           2885.57         22.90
## 60:      27331           1996.15         13.07
## 61:      13024           2649.80         20.64
## 62:      49188           3834.10         28.78
## 63:      11874           1634.34          9.33
## 64:     741952           1682.47         12.47
## 65:      88675           2065.29         18.05
## 66:     228537           2017.65         18.88
## 67:      10690            295.76         34.98
## 68:      10588           1889.44         14.65
```

```
## 69:      12200             1938.17          20.54
## 70:      12652              284.75          38.96
## 71:     358548             3829.21          31.17
## 72:     372242             2601.60          18.82
## 73:      87492             3530.78          36.56
## 74:      72411             2682.28          16.05
## 75:      13547               39.87          28.82
## 76:      10201             2650.26          36.51
## 77:      86835              157.99          29.56
## 78:     275221             3928.03          26.34
## 79:      49847             2037.32           9.78
## 80:     141686             1908.96          17.07
## 81:      20651              712.74          46.12
## 82:      75695             2089.28          18.75
## 83:      49998             1818.82          15.48
## 84:      47669             1964.89          17.60
## 85:      17363             2210.88          23.20
## 86:      22754             2523.46          17.84
## 87:      29541              352.88          39.35
## 88:     222103             1807.31           7.43
## 89:      15464             1640.60           9.28
## 90:      15023             1987.15          10.97
## 91:      87425             3239.20          16.10
## 92:      22122             2728.14          20.77
## 93:      10864             2333.62          25.07
## 94:      33892             2572.91          25.56
## 95:      61945             2956.98          21.84
##     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.

```
max_extra_predictors <- 4
# Define predictor variables for model selection
predictors <- c("perCapInc", "PctEmploy",
                "PctLess9thGrade", "PctNotHSGrad", "PctBSorMore",
                "racepctblack", "agePct12t29", "PctImmig")

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

formulas <- list()
for (i in 1:max_extra_predictors) {
  tmp <- combn(predictors, i)
  tmp <- apply(tmp, 2, paste, collapse=" + ")
  tmp <- paste0("ViolentCrimesPerPop~PctPopUnderPov + ", tmp)
  formulas[[i]] <- tmp
}
formulas <- unlist(formulas)
formulas <- sapply(formulas, as.formula)
```

```r
models <- lapply(formulas, lm, data=train_data)

bics <- sapply(models, BIC)
r_square <- sapply(models, function(m) summary(m)$r.squared)
adj_r_square <- sapply(models, function(m) summary(m)$adj.r.squared)
formula_vector <- vapply(formulas, function(f) paste(deparse(f), collapse = ""), character(1))

# build the frame
model_ranking <- data.frame(
  formula = formula_vector,
  r.square = r_square,
  adj.r.square = adj_r_square,
  BIC = bics
)
model_ranking <- model_ranking[order(model_ranking$BIC), ]
head(model_ranking, 10)
```

```
##
## ViolentCrimesPerPop~PctPopUnderPov + PctLess9thGrade + PctNotHSGrad + racepctblack + PctImmig Violen
## ViolentCrimesPerPop~PctPopUnderPov + PctLess9thGrade + PctBSorMore + racepctblack + PctImmig   Violer
## ViolentCrimesPerPop~PctPopUnderPov + PctBSorMore + racepctblack + agePct12t29 + PctImmig           V
## ViolentCrimesPerPop~PctPopUnderPov + perCapInc + PctBSorMore + racepctblack + PctImmig
## ViolentCrimesPerPop~PctPopUnderPov + PctBSorMore + racepctblack + PctImmig
## ViolentCrimesPerPop~PctPopUnderPov + perCapInc + racepctblack + agePct12t29 + PctImmig
## ViolentCrimesPerPop~PctPopUnderPov + PctNotHSGrad + PctBSorMore + racepctblack + PctImmig         Vi
## ViolentCrimesPerPop~PctPopUnderPov + PctEmploy + PctBSorMore + racepctblack + PctImmig
## ViolentCrimesPerPop~PctPopUnderPov + PctNotHSGrad + racepctblack + agePct12t29 + PctImmig         Vi
## ViolentCrimesPerPop~PctPopUnderPov + racepctblack + agePct12t29 + PctImmig
##                                                                                                r.squ
## ViolentCrimesPerPop~PctPopUnderPov + PctLess9thGrade + PctNotHSGrad + racepctblack + PctImmig 0.5555
## ViolentCrimesPerPop~PctPopUnderPov + PctLess9thGrade + PctBSorMore + racepctblack + PctImmig  0.5469
## ViolentCrimesPerPop~PctPopUnderPov + PctBSorMore + racepctblack + agePct12t29 + PctImmig      0.5461
## ViolentCrimesPerPop~PctPopUnderPov + perCapInc + PctBSorMore + racepctblack + PctImmig        0.5421
## ViolentCrimesPerPop~PctPopUnderPov + PctBSorMore + racepctblack + PctImmig                    0.5396
## ViolentCrimesPerPop~PctPopUnderPov + perCapInc + racepctblack + agePct12t29 + PctImmig        0.5413
## ViolentCrimesPerPop~PctPopUnderPov + PctNotHSGrad + PctBSorMore + racepctblack + PctImmig     0.5402
## ViolentCrimesPerPop~PctPopUnderPov + PctEmploy + PctBSorMore + racepctblack + PctImmig        0.5396
## ViolentCrimesPerPop~PctPopUnderPov + PctNotHSGrad + racepctblack + agePct12t29 + PctImmig     0.5369
## ViolentCrimesPerPop~PctPopUnderPov + racepctblack + agePct12t29 + PctImmig                    0.5346
##                                                                                               adj.r.s
## ViolentCrimesPerPop~PctPopUnderPov + PctLess9thGrade + PctNotHSGrad + racepctblack + PctImmig    0.55
## ViolentCrimesPerPop~PctPopUnderPov + PctLess9thGrade + PctBSorMore + racepctblack + PctImmig     0.54
## ViolentCrimesPerPop~PctPopUnderPov + PctBSorMore + racepctblack + agePct12t29 + PctImmig         0.54
## ViolentCrimesPerPop~PctPopUnderPov + perCapInc + PctBSorMore + racepctblack + PctImmig           0.54
## ViolentCrimesPerPop~PctPopUnderPov + PctBSorMore + racepctblack + PctImmig                       0.53
## ViolentCrimesPerPop~PctPopUnderPov + perCapInc + racepctblack + agePct12t29 + PctImmig           0.53
## ViolentCrimesPerPop~PctPopUnderPov + PctNotHSGrad + PctBSorMore + racepctblack + PctImmig        0.53
## ViolentCrimesPerPop~PctPopUnderPov + PctEmploy + PctBSorMore + racepctblack + PctImmig           0.53
## ViolentCrimesPerPop~PctPopUnderPov + PctNotHSGrad + racepctblack + agePct12t29 + PctImmig        0.53
## ViolentCrimesPerPop~PctPopUnderPov + racepctblack + agePct12t29 + PctImmig                       0.53
##                                                                                                    BI
## ViolentCrimesPerPop~PctPopUnderPov + PctLess9thGrade + PctNotHSGrad + racepctblack + PctImmig 23675.
## ViolentCrimesPerPop~PctPopUnderPov + PctLess9thGrade + PctBSorMore + racepctblack + PctImmig  23705.
```

```
## ViolentCrimesPerPop~PctPopUnderPov + PctBSorMore + racepctblack + agePct12t29 + PctImmig      23709.
## ViolentCrimesPerPop~PctPopUnderPov + perCapInc + PctBSorMore + racepctblack + PctImmig        23722.
## ViolentCrimesPerPop~PctPopUnderPov + PctBSorMore + racepctblack + PctImmig                    23724.
## ViolentCrimesPerPop~PctPopUnderPov + perCapInc + racepctblack + agePct12t29 + PctImmig        23725.
## ViolentCrimesPerPop~PctPopUnderPov + PctNotHSGrad + PctBSorMore + racepctblack + PctImmig     23729.
## ViolentCrimesPerPop~PctPopUnderPov + PctEmploy + PctBSorMore + racepctblack + PctImmig        23731.
## ViolentCrimesPerPop~PctPopUnderPov + PctNotHSGrad + racepctblack + agePct12t29 + PctImmig     23740.
## ViolentCrimesPerPop~PctPopUnderPov + racepctblack + agePct12t29 + PctImmig                    23741.
```

We then run the multivariate regression equation

```
# Print best model
best <- which.min(model_ranking$BIC)
best_pred <- model_ranking$formula[best]
cat("\nBest model:\n")
```

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

```
cat(best_pred, "\n")
```

```
## ViolentCrimesPerPop ~ PctPopUnderPov + PctLess9thGrade + PctNotHSGrad +     racepctblack + PctImmig
```

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

```
## BIC: 23675.65
```

```
cat("Adjusted R²:", round(model_ranking$adj.r.square[best], 4), "\n")
```

```
## Adjusted R²: 0.5541
```

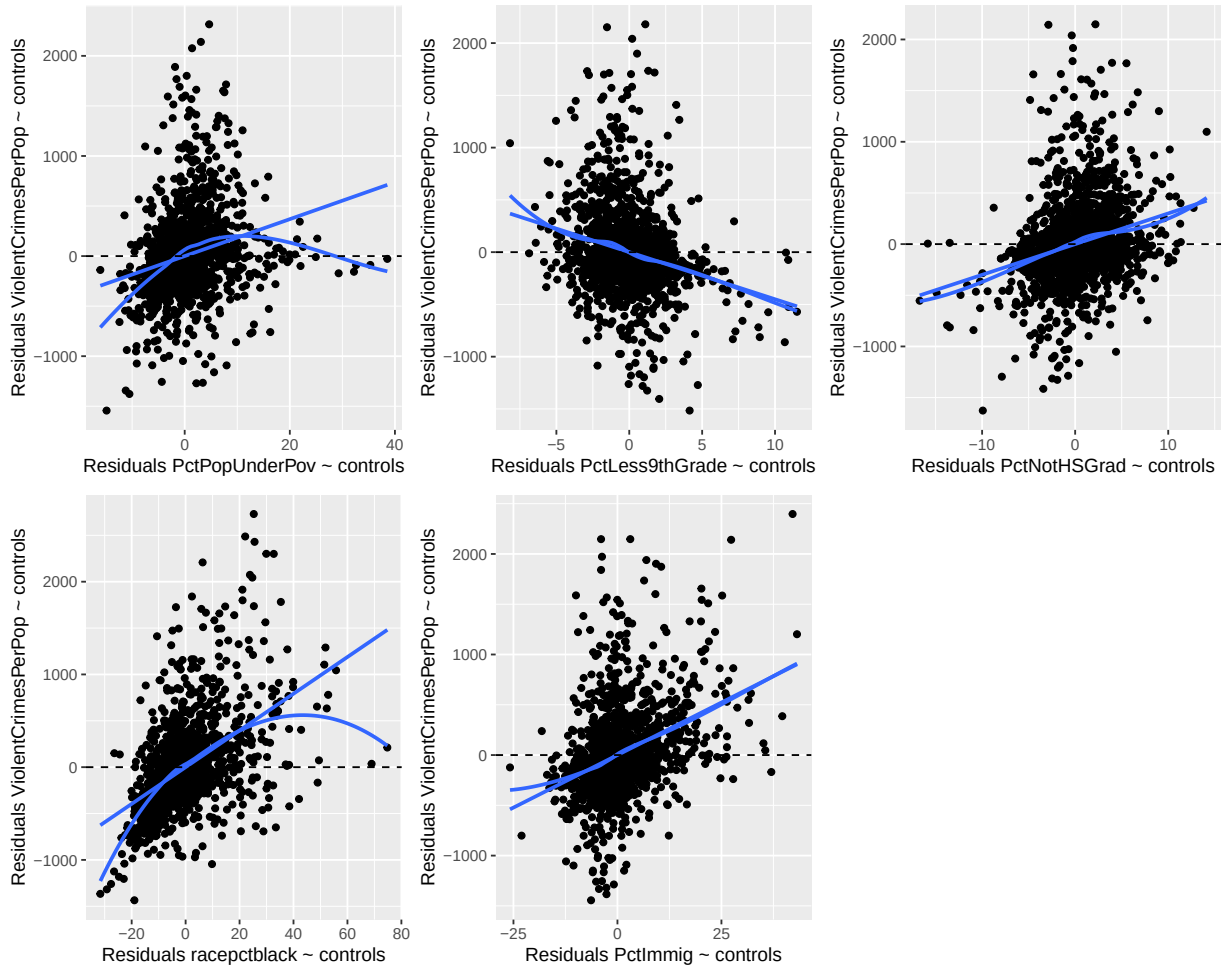
```
fit_multi <- lm(best_pred, data = train_data)
multi_var_summary <- summary(fit_multi)
multi_var_summary
```

```
##
## Call:
## lm(formula = best_pred, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1329.06  -203.10   -51.51   139.40  2229.43
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -232.6489    30.1281  -7.722 2.01e-14 ***
## PctPopUnderPov    18.4484     1.8037  10.228  < 2e-16 ***
## PctLess9thGrade  -45.0540     4.6966  -9.593  < 2e-16 ***
## PctNotHSGrad      29.9988     2.7693  10.832  < 2e-16 ***
## racepctblack      19.7995     0.8712  22.726  < 2e-16 ***
```

```
## PctImmig          20.9029     1.3097  15.961  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 398.9 on 1589 degrees of freedom
## Multiple R-squared:  0.5555, Adjusted R-squared:  0.5541
## F-statistic: 397.2 on 5 and 1589 DF,  p-value: < 2.2e-16
```

Partial Regression Plots

```
predictor_list <- strsplit(best_pred, split = " ")[[1]]
predictor_list <- grep(pattern = "\\w", predictor_list[3:length(predictor_list)], value = TRUE)

partial_regression_plot <- function(data, outcome, predictor, predictor_list) {
  temp <- train_data
  controls <- setdiff(predictor_list, predictor)

  formula_outcome  <- as.formula(paste(outcome, "~", paste(controls, collapse = " + ")))
  formula_pred <- as.formula(paste(predictor, "~", paste(controls, collapse = " + ")))

  lm_outcome <- lm(formula_outcome, data = temp)
  temp$resid_outcome <- resid(lm_outcome)
  lm_pred <- lm(formula_pred, data = temp)
  temp$resid_pred <- resid(lm_pred)

ggplot(temp, aes(x = resid_pred, y = resid_outcome)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  geom_smooth(method = "loess", se = FALSE) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  labs(x = paste("Residuals ", predictor, " ~ controls", sep = ""), y = paste("Residuals ", outcome, "
}

plots <- lapply(predictor_list, function(predictor) {
    partial_regression_plot(train_data, "ViolentCrimesPerPop", predictor, predictor_list)
}
)

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



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. -> in ons protocol staat 1 of meer en hier zijn meerdere significant?

```
selected_vars <- row.names(multi_var_summary$coefficients[-(1:2),])
interaction_terms <- paste("PctPopUnderPov", selected_vars, sep = ":")

formulas <- sapply(interaction_terms, function(i) {
  paste0(best_pred, " + ", i)
})

formulas <- lapply(formulas, as.formula)
models <- lapply(formulas, lm, data=train_data)

interaction_models_form <- vapply(formulas, function(f)
  paste(deparse(f), collapse = ""), character(1))
summary_val_extraction <- function(x, item) {
  tmp <- summary(x)$coefficients
  tmp[nrow(tmp), item]
```

```
}
p_vals_interaction <- sapply(models, function(m)
  summary_val_extraction(m, "Pr(>|t|)"))
t_vals_interaction <- sapply(models, function(m)
  summary_val_extraction(m, "t value"))

bics <- sapply(models, BIC)
r_square <- sapply(models, function(m) summary(m)$r.squared)
adj_r_square <- sapply(models, function(m) summary(m)$adj.r.squared)
delta_adj_r_square <- sapply(models, function(m)
  summary(m)$adj.r.squared - summary(fit_multi)$adj.r.squared)

interaction_table <- data.frame(
  p.vals = p_vals_interaction,
  t.vals = t_vals_interaction,
  r.square = r_square,
  adj.r.square = adj_r_square,
  delta.adj = delta_adj_r_square,
  BIC = bics
)
interaction_table <- interaction_table[order(interaction_table$p.vals), ]
interaction_table
```

```
##                                    p.vals    t.vals  r.square adj.r.square
## PctPopUnderPov:PctImmig        0.0001302069  3.835508 0.5595968    0.5579328
## PctPopUnderPov:PctNotHSGrad    0.3027525126  1.030883 0.5558142    0.5541359
## PctPopUnderPov:PctLess9thGrade 0.3126777270 -1.009940 0.5558023    0.5541239
## PctPopUnderPov:racepctblack    0.4470740197  0.760491 0.5556788    0.5540000
##                                    delta.adj      BIC
## PctPopUnderPov:PctImmig         3.814497e-03 23668.32
## PctPopUnderPov:PctNotHSGrad     1.759899e-05 23681.96
## PctPopUnderPov:PctLess9thGrade  5.606284e-06 23682.00
## PctPopUnderPov:racepctblack    -1.183495e-04 23682.45
```

```
best_interaction <- row.names(interaction_table[1,])
```

Multivariate Model

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

```
# Estimate model with interaction
formula_final <- as.formula(paste(best_pred, "+", 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
## -1388.66  -197.66   -51.16   133.84  2256.76
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -173.2794    33.7570  -5.133 3.20e-07 ***
## PctPopUnderPov            14.8240     2.0294   7.305 4.38e-13 ***
## PctLess9thGrade          -49.9935     4.8505 -10.307  < 2e-16 ***
## PctNotHSGrad              31.4252     2.7824  11.294  < 2e-16 ***
## racepctblack              20.4803     0.8855  23.130  < 2e-16 ***
## PctImmig                  13.4416     2.3420   5.739 1.14e-08 ***
## PctPopUnderPov:PctImmig    0.5299     0.1382   3.836  0.00013 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 397.1 on 1588 degrees of freedom
## Multiple R-squared:  0.5596, Adjusted R-squared:  0.5579
## F-statistic: 336.3 on 6 and 1588 DF,  p-value: < 2.2e-16
```

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

```
##                               2.5 %       97.5 %
## (Intercept)             -239.492305 -107.0665586
## PctPopUnderPov            10.843458   18.8045048
## PctLess9thGrade          -59.507631  -40.4793731
## PctNotHSGrad              25.967581   36.8828471
## racepctblack              18.743528   22.2170833
## PctImmig                   8.847921   18.0352651
## PctPopUnderPov:PctImmig    0.258924    0.8009256
```

Multicollinearity Check

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

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

```
##          PctPopUnderPov         PctLess9thGrade            PctNotHSGrad
##                2.991782               11.067405                9.562062
##            racepctblack                PctImmig PctPopUnderPov:PctImmig
##                1.515199                4.216785                5.974934
```

As already mentioned in the section on descriptives there is multicollinearity, as PctNotHSGrad and PctLess9thGrade are highly correlated. 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(predictor_list, 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 -3.7166306 0.0002088959
## 3     PctPopUnderPov:PctImmig  1.1039443 0.2697844394
## 2 PctPopUnderPov:racepctblack  0.7457968 0.4559004825
```

```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
## -1522.97  -201.89   -50.13   135.95  2188.07
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -159.5727    38.0697  -4.192 2.92e-05 ***
## PctPopUnderPov                22.8200     3.0693   7.435 1.70e-13 ***
## PctNotHSGrad                  11.2798     1.8493   6.100 1.33e-09 ***
## racepctblack                  22.2709     0.8533  26.100  < 2e-16 ***
## PctImmig                      16.7522     1.2525  13.375  < 2e-16 ***
## PctPopUnderPov:PctNotHSGrad   -0.3564     0.0959  -3.717 0.000209 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 408.5 on 1589 degrees of freedom
## Multiple R-squared:  0.5338, Adjusted R-squared:  0.5324
## F-statistic: 363.9 on 5 and 1589 DF,  p-value: < 2.2e-16
```

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

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

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

```
##                                   2.5 %      97.5 %
## (Intercept)                 -234.2448670 -84.9005221
## PctPopUnderPov                16.7997189  28.8402113
## PctNotHSGrad                   7.6526091  14.9070548
## racepctblack                  20.5972153  23.9446324
## PctImmig                      14.2955337  19.2088635
## PctPopUnderPov:PctNotHSGrad   -0.5445112  -0.1683157
```

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

```
##              PctPopUnderPov                PctNotHSGrad
##                    6.469284                    3.992725
##                racepctblack                    PctImmig
##                    1.330207                    1.140070
## PctPopUnderPov:PctNotHSGrad
##                   11.539465
```

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:  0.5541
```

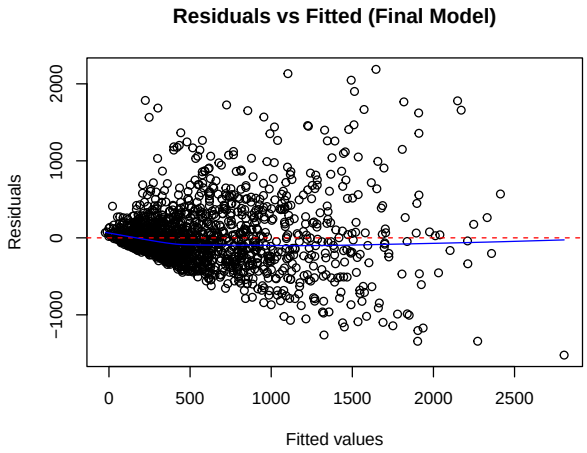
```r
cat("Reduced model adjusted R²:  ", round(summary(fit_final_reduced)$adj.r.squared, 4), "\n")
```

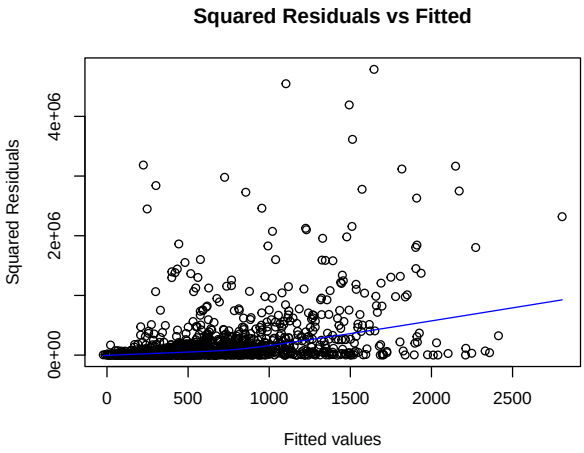
```
## Reduced model adjusted R²:   0.5324
```

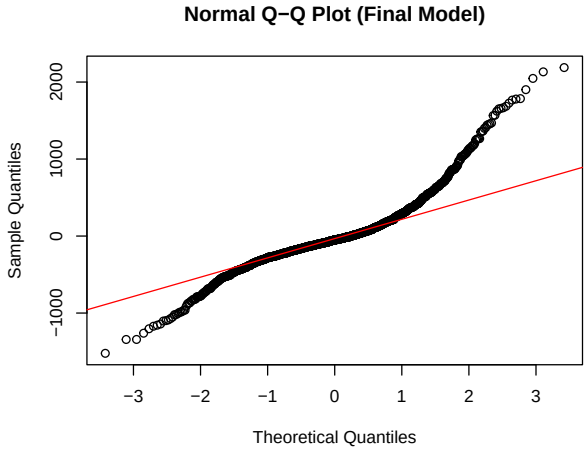
```r
# Update fit_final
fit_final <- fit_final_reduced
formula_final <- formula_final_reduced
best_predictors <- best_predictors_reduced
```

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")

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

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







Transformation of Y

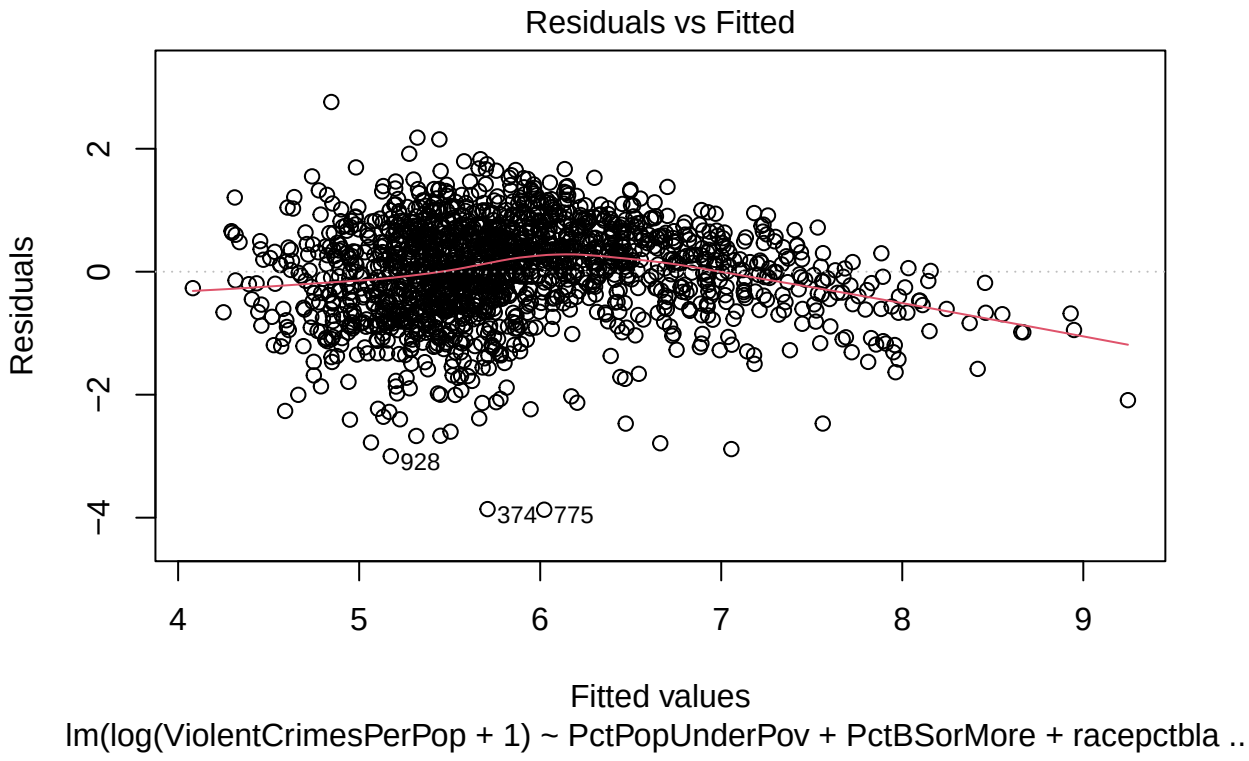
We see that the assumptions of normality and equal error variances of the error terms are again violated in the original multivariate model. A possible solution is to apply a log transformation of Y. The result shows approximate constant variances and residuals reasonably close to normal, applying the log transformation therefore offers a substantial improvement of our model. The reduction in R-squared suggests that the model-fit was inflated due to heteroscedasticity. We still see slightly heavy tails.

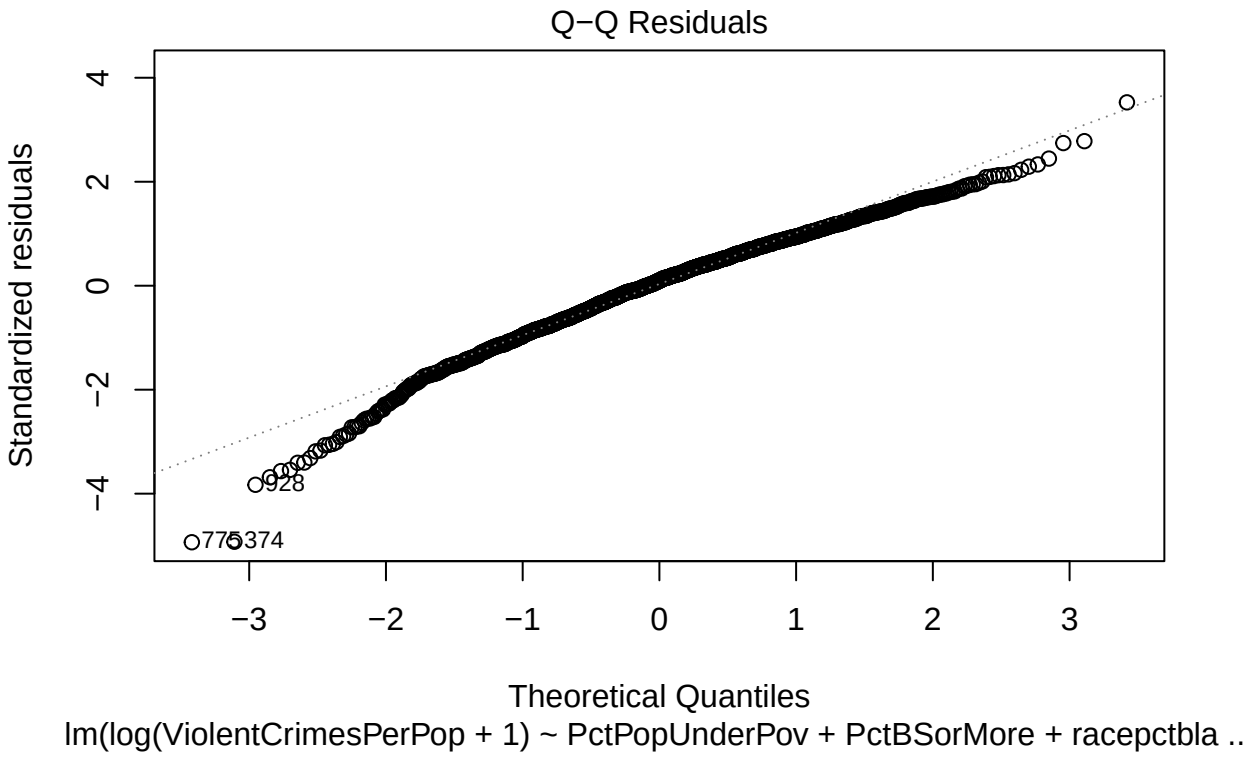
```
fit_log <- lm(log(ViolentCrimesPerPop + 1) ~
                PctPopUnderPov + PctBSorMore + racepctblack +
                PctImmig + PctPopUnderPov:PctBSorMore,
              data = train_data)

summary(fit_log)
```

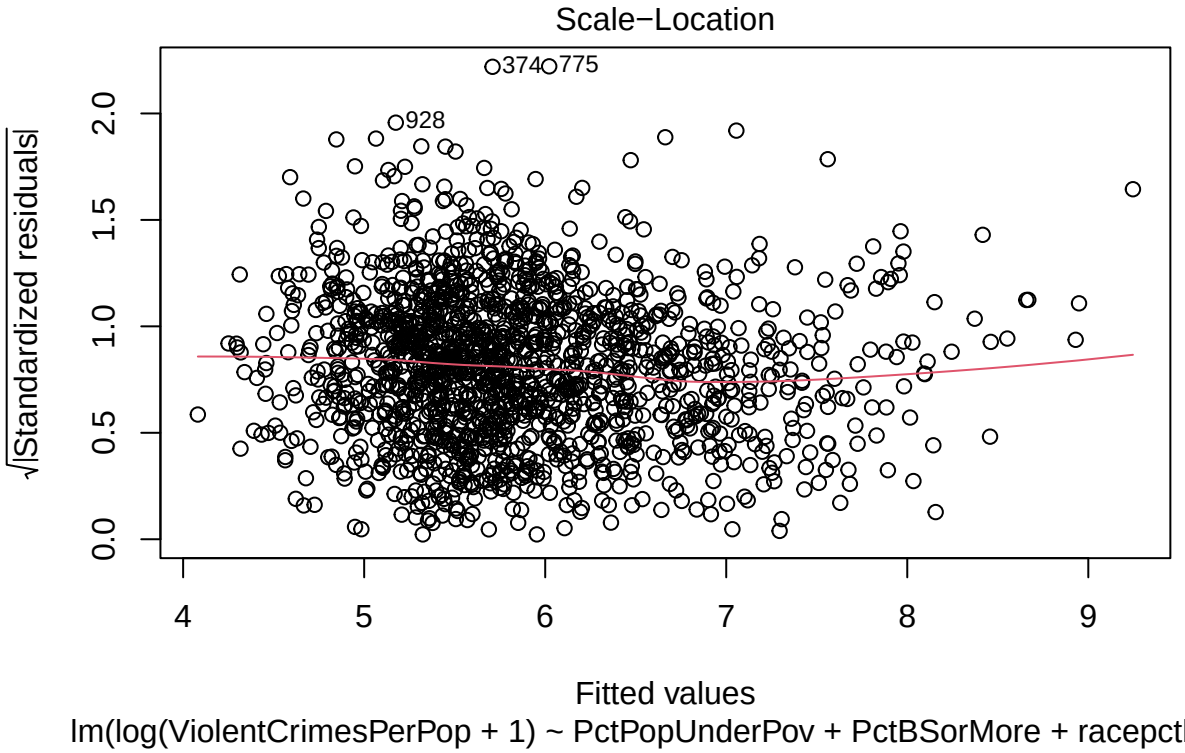
```
##
## Call:
## lm(formula = log(ViolentCrimesPerPop + 1) ~ PctPopUnderPov +
##     PctBSorMore + racepctblack + PctImmig + PctPopUnderPov:PctBSorMore,
##     data = train_data)
```

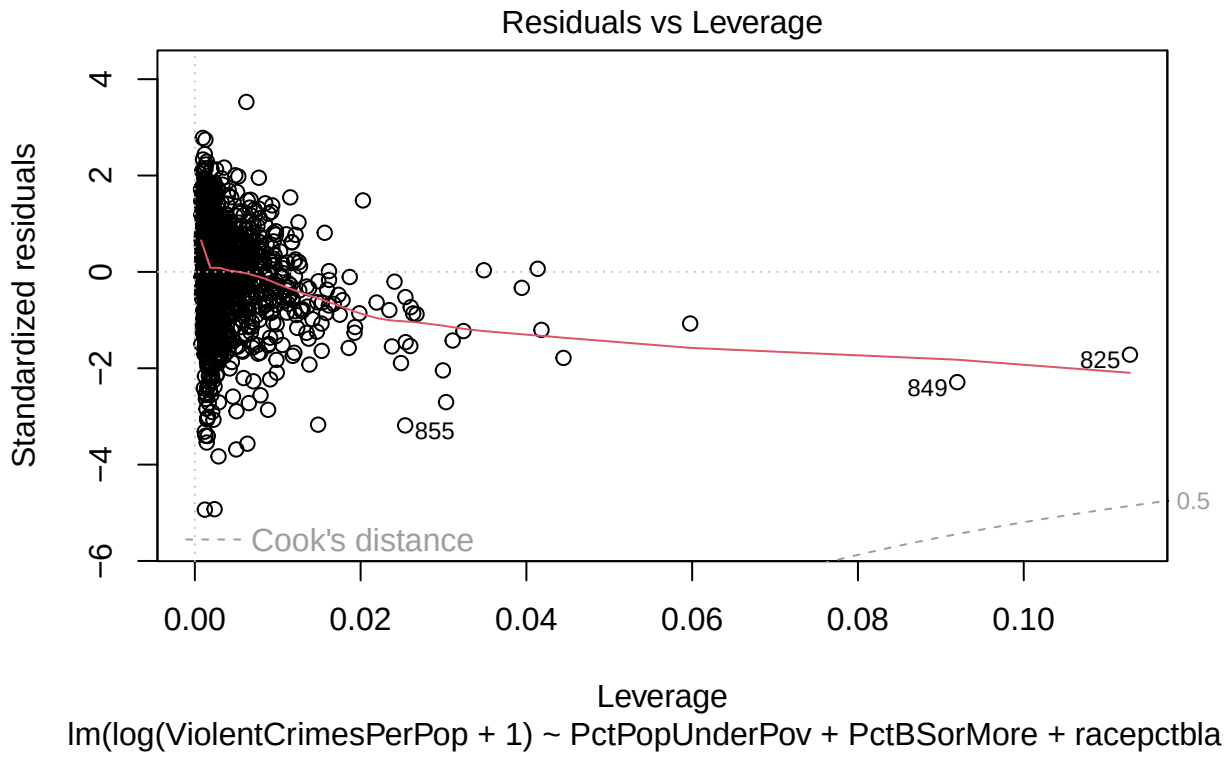
```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8703 -0.4927  0.0785  0.5465  2.7596
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 5.5072573  0.0751126  73.320  < 2e-16 ***
## PctPopUnderPov              0.0202729  0.0049867   4.065 5.03e-05 ***
## PctBSorMore                -0.0245700  0.0024184 -10.160  < 2e-16 ***
## racepctblack                0.0321281  0.0016452  19.528  < 2e-16 ***
## PctImmig                    0.0338266  0.0023043  14.680  < 2e-16 ***
## PctPopUnderPov:PctBSorMore  0.0005147  0.0001813   2.839  0.00458 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7849 on 1589 degrees of freedom
## Multiple R-squared:   0.49,  Adjusted R-squared:  0.4884
## F-statistic: 305.4 on 5 and 1589 DF,  p-value: < 2.2e-16
```

```
plot(fit_log)
```




Q−Q Residuals
Standardized residuals
4
2
0
−2
−4
928
775374
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(log(ViolentCrimesPerPop + 1) ~ PctPopUnderPov + PctBSorMore + racepctbla ..


Scale−Location
374
775
928
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
4
5
6
7
8
9
Fitted values
lm(log(ViolentCrimesPerPop + 1) ~ PctPopUnderPov + PctBSorMore + racepctbla ..




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

Rerun selection

We will repeat the model building process with the log-transformed ViolentCrimesPerPop.

```
train_data$logViolent <- log(train_data$ViolentCrimesPerPop + 1)
```

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.

```
max_extra_predictors <- 4
# Define predictor variables for model selection
predictors <- c("perCapInc", "PctEmploy",
                "PctLess9thGrade", "PctNotHSGrad", "PctBSorMore",
                "racepctblack", "agePct12t29", "PctImmig")

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

```r
formulas <- list()
for (i in 1:max_extra_predictors) {
  tmp <- combn(predictors, i)
  tmp <- apply(tmp, 2, paste, collapse=" + ")
  tmp <- paste0("logViolent~PctPopUnderPov + ", tmp)
  formulas[[i]] <- tmp
}
formulas <- unlist(formulas)
formulas <- sapply(formulas, as.formula)
models <- lapply(formulas, lm, data=train_data)

bics <- sapply(models, BIC)
r_square <- sapply(models, function(m) summary(m)$r.squared)
adj_r_square <- sapply(models, function(m) summary(m)$adj.r.squared)
formula_vector <- vapply(formulas, function(f) paste(deparse(f), collapse = ""), character(1))

# build the frame
model_ranking <- data.frame(
  formula = formula_vector,
  r.square = r_square,
  adj.r.square = adj_r_square,
  BIC = bics
)
model_ranking <- model_ranking[order(model_ranking$BIC), ]
head(model_ranking, 10)
```

```
##
## logViolent~PctPopUnderPov + PctLess9thGrade + PctBSorMore + racepctblack + PctImmig   logViolent ~ P
## logViolent~PctPopUnderPov + PctLess9thGrade + PctNotHSGrad + racepctblack + PctImmig logViolent ~ Pct
## logViolent~PctPopUnderPov + PctNotHSGrad + PctBSorMore + racepctblack + PctImmig         logViolent
## logViolent~PctPopUnderPov + PctBSorMore + racepctblack + PctImmig
## logViolent~PctPopUnderPov + PctBSorMore + racepctblack + agePct12t29 + PctImmig          logViolent
## logViolent~PctPopUnderPov + PctEmploy + PctBSorMore + racepctblack + PctImmig              logViole
## logViolent~PctPopUnderPov + perCapInc + PctBSorMore + racepctblack + PctImmig              logViole
## logViolent~PctPopUnderPov + perCapInc + racepctblack + agePct12t29 + PctImmig              logViole
## logViolent~PctPopUnderPov + perCapInc + PctNotHSGrad + racepctblack + PctImmig            logViolen
## logViolent~PctPopUnderPov + perCapInc + racepctblack + PctImmig
##                                                                                          r.square
## logViolent~PctPopUnderPov + PctLess9thGrade + PctBSorMore + racepctblack + PctImmig  0.5027349
## logViolent~PctPopUnderPov + PctLess9thGrade + PctNotHSGrad + racepctblack + PctImmig 0.4963986
## logViolent~PctPopUnderPov + PctNotHSGrad + PctBSorMore + racepctblack + PctImmig     0.4922422
## logViolent~PctPopUnderPov + PctBSorMore + racepctblack + PctImmig                    0.4874437
## logViolent~PctPopUnderPov + PctBSorMore + racepctblack + agePct12t29 + PctImmig      0.4886407
## logViolent~PctPopUnderPov + PctEmploy + PctBSorMore + racepctblack + PctImmig        0.4885967
## logViolent~PctPopUnderPov + perCapInc + PctBSorMore + racepctblack + PctImmig        0.4879832
## logViolent~PctPopUnderPov + perCapInc + racepctblack + agePct12t29 + PctImmig        0.4752934
## logViolent~PctPopUnderPov + perCapInc + PctNotHSGrad + racepctblack + PctImmig       0.4670193
## logViolent~PctPopUnderPov + perCapInc + racepctblack + PctImmig                      0.4641401
##                                                                                          adj.r.square
## logViolent~PctPopUnderPov + PctLess9thGrade + PctBSorMore + racepctblack + PctImmig     0.5011702
## logViolent~PctPopUnderPov + PctLess9thGrade + PctNotHSGrad + racepctblack + PctImmig    0.4948139
## logViolent~PctPopUnderPov + PctNotHSGrad + PctBSorMore + racepctblack + PctImmig        0.4906445
## logViolent~PctPopUnderPov + PctBSorMore + racepctblack + PctImmig                       0.4861542
```

```
## logViolent~PctPopUnderPov + PctBSorMore + racepctblack + agePct12t29 + PctImmig         0.4870316
## logViolent~PctPopUnderPov + PctEmploy + PctBSorMore + racepctblack + PctImmig           0.4869875
## logViolent~PctPopUnderPov + perCapInc + PctBSorMore + racepctblack + PctImmig           0.4863720
## logViolent~PctPopUnderPov + perCapInc + racepctblack + agePct12t29 + PctImmig           0.4736424
## logViolent~PctPopUnderPov + perCapInc + PctNotHSGrad + racepctblack + PctImmig          0.4653422
## logViolent~PctPopUnderPov + perCapInc + racepctblack + PctImmig                         0.4627920
##                                                                                           BIC
## logViolent~PctPopUnderPov + PctLess9thGrade + PctBSorMore + racepctblack + PctImmig  3759.066
## logViolent~PctPopUnderPov + PctLess9thGrade + PctNotHSGrad + racepctblack + PctImmig 3779.262
## logViolent~PctPopUnderPov + PctNotHSGrad + PctBSorMore + racepctblack + PctImmig     3792.372
## logViolent~PctPopUnderPov + PctBSorMore + racepctblack + PctImmig                    3800.000
## logViolent~PctPopUnderPov + PctBSorMore + racepctblack + agePct12t29 + PctImmig      3803.645
## logViolent~PctPopUnderPov + PctEmploy + PctBSorMore + racepctblack + PctImmig        3803.783
## logViolent~PctPopUnderPov + perCapInc + PctBSorMore + racepctblack + PctImmig        3805.695
## logViolent~PctPopUnderPov + perCapInc + racepctblack + agePct12t29 + PctImmig        3844.743
## logViolent~PctPopUnderPov + perCapInc + PctNotHSGrad + racepctblack + PctImmig       3869.699
## logViolent~PctPopUnderPov + perCapInc + racepctblack + PctImmig                      3870.917
```

We then run the multivariate regression equation

```
# Print best model
best <- which.min(model_ranking$BIC)
best_pred <- model_ranking$formula[best]
cat("\nBest model:\n")
```

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

```
cat(best_pred, "\n")
```

```
## logViolent ~ PctPopUnderPov + PctLess9thGrade + PctBSorMore +     racepctblack + PctImmig
```

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

```
## BIC: 3759.07
```

```
cat("Adjusted R²:", round(model_ranking$adj.r.square[best], 4), "\n")
```

```
## Adjusted R²: 0.5012
```

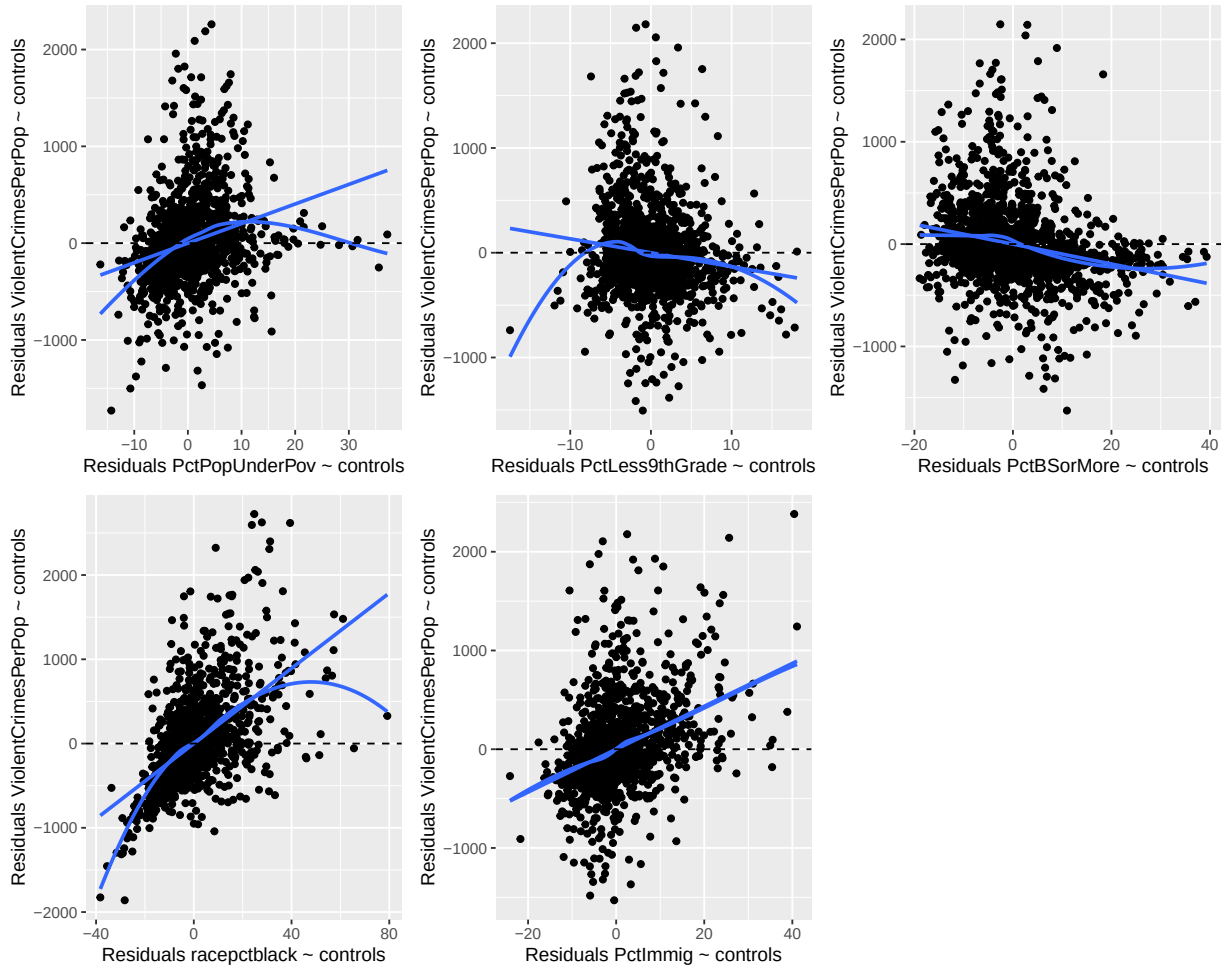
```
fit_multi <- lm(best_pred, data = train_data)
summary(fit_multi)
```

```
##
## Call:
## lm(formula = best_pred, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9823 -0.4742  0.0692  0.5361  2.6996
```

```
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.668443   0.073455   77.17  < 2e-16 ***
## PctPopUnderPov    0.046465   0.003516   13.22  < 2e-16 ***
## PctLess9thGrade -0.035244   0.005042   -6.99 4.03e-12 ***
## PctBSorMore     -0.027563   0.002025  -13.61  < 2e-16 ***
## racepctblack      0.030716   0.001623   18.93  < 2e-16 ***
## PctImmig          0.042330   0.002628   16.11  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.775 on 1589 degrees of freedom
## Multiple R-squared:  0.5027, Adjusted R-squared:  0.5012
## F-statistic: 321.3 on 5 and 1589 DF,  p-value: < 2.2e-16
```

Partial Regression Plots

```
predictor_list <- strsplit(best_pred, split = " ")[[1]]
predictor_list <- grep(pattern = "\\w", predictor_list[3:length(predictor_list)], value = TRUE)

plots <- lapply(predictor_list, function(predictor) {
    partial_regression_plot(train_data, "ViolentCrimesPerPop", predictor, predictor_list)
}
)

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



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. -> in ons protocol staat 1 of meer en hier zijn meerdere significant?

```
# store all other pred except for main pred
other <- setdiff(predictor_list, "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("logViolent ~",
                                           paste(predictor_list, 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:
```

```
  print(interaction_results)
```

```
##                       interaction   t_value      p_value  delta_adjrsq
## 3    PctPopUnderPov:racepctblack -6.554624 7.519773e-11  0.0128344082
## 1 PctPopUnderPov:PctLess9thGrade -4.524553 6.502250e-06  0.0060386562
## 4        PctPopUnderPov:PctImmig -2.804443 5.101727e-03  0.0021458069
## 2     PctPopUnderPov:PctBSorMore  0.162906 8.706132e-01 -0.0003057831
```

```
  # Select best interaction
  best_interaction <- c(interaction_results$interaction[1:3])
  cat("\nSelected interaction term:", best_interaction, "\n")
```

```
##
## Selected interaction term: PctPopUnderPov:racepctblack PctPopUnderPov:PctLess9thGrade PctPopUnderPov
```

Multivariate Model

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

```
# Estimate model with interaction
formula_final <- as.formula(paste("logViolent ~",
                                  paste(predictor_list, collapse = " + "), "+",
                                  paste(best_interaction, collapse = " + ")))

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
## -4.0904 -0.4586  0.0805  0.5176  2.7475
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     5.2663990  0.0995459  52.904  < 2e-16 ***
## PctPopUnderPov                  0.0683878  0.0046818  14.607  < 2e-16 ***
## PctLess9thGrade                -0.0165471  0.0082758  -1.999   0.0457 *
## PctBSorMore                    -0.0247628  0.0021622 -11.453  < 2e-16 ***
## racepctblack                    0.0495189  0.0034818  14.222  < 2e-16 ***
## PctImmig                        0.0510966  0.0050203  10.178  < 2e-16 ***
## PctPopUnderPov:racepctblack    -0.0010236  0.0001576  -6.494 1.11e-10 ***
## PctPopUnderPov:PctLess9thGrade -0.0006663  0.0003181  -2.095   0.0364 *
## PctPopUnderPov:PctImmig        -0.0007679  0.0003025  -2.539   0.0112 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7606 on 1586 degrees of freedom
## Multiple R-squared:  0.522,  Adjusted R-squared:  0.5196
## F-statistic: 216.5 on 8 and 1586 DF,  p-value: < 2.2e-16
```

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

```
##                                      2.5 %       97.5 %
## (Intercept)                     5.071143689  5.4616543948
## PctPopUnderPov                  0.059204622  0.0775708816
## PctLess9thGrade                -0.032779705 -0.0003145522
## PctBSorMore                    -0.029003919 -0.0205217350
## racepctblack                    0.042689578  0.0563482197
## PctImmig                        0.041249413  0.0609437232
## PctPopUnderPov:racepctblack    -0.001332824 -0.0007144760
## PctPopUnderPov:PctLess9thGrade -0.001290321 -0.0000423758
## PctPopUnderPov:PctImmig        -0.001361238 -0.0001746585
```

Multicollinearity Check

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

```
library(car)
# practicum: car::vif(fit)
vif <- vif(fit_final)
print(vif)
```

```
##                  PctPopUnderPov                PctLess9thGrade
##                        4.341591                       8.784164
##                     PctBSorMore                   racepctblack
##                        2.054733                       6.387843
##                        PctImmig    PctPopUnderPov:racepctblack
##                        5.283246                       8.162681
## PctPopUnderPov:PctLess9thGrade        PctPopUnderPov:PctImmig
##                       12.538911                       7.808059
```

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

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

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

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

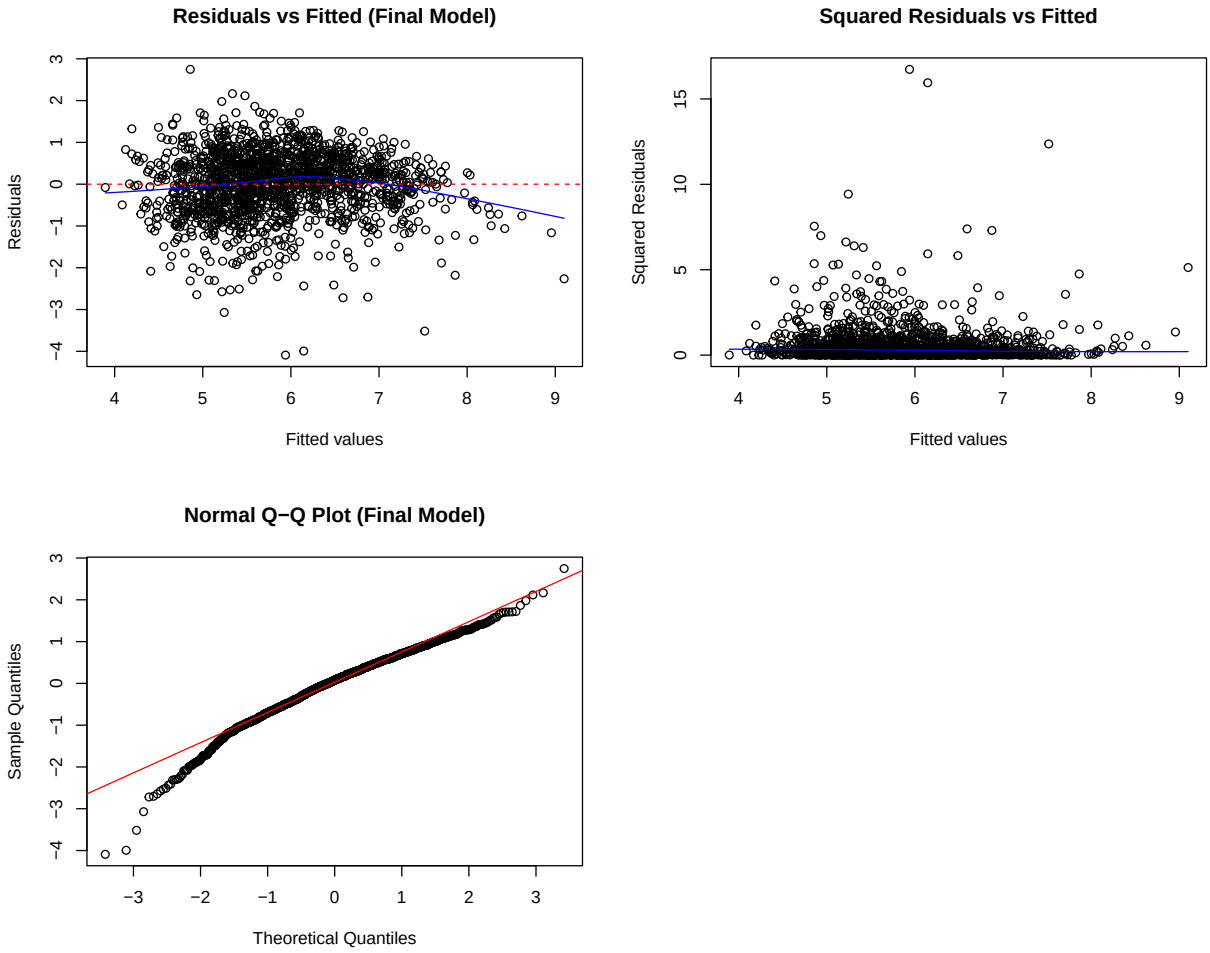
```
## Original model adjusted R²:  0.5012
```

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

```
## interaction term model adjusted R²:   0.5196
```

Assumption checks final model

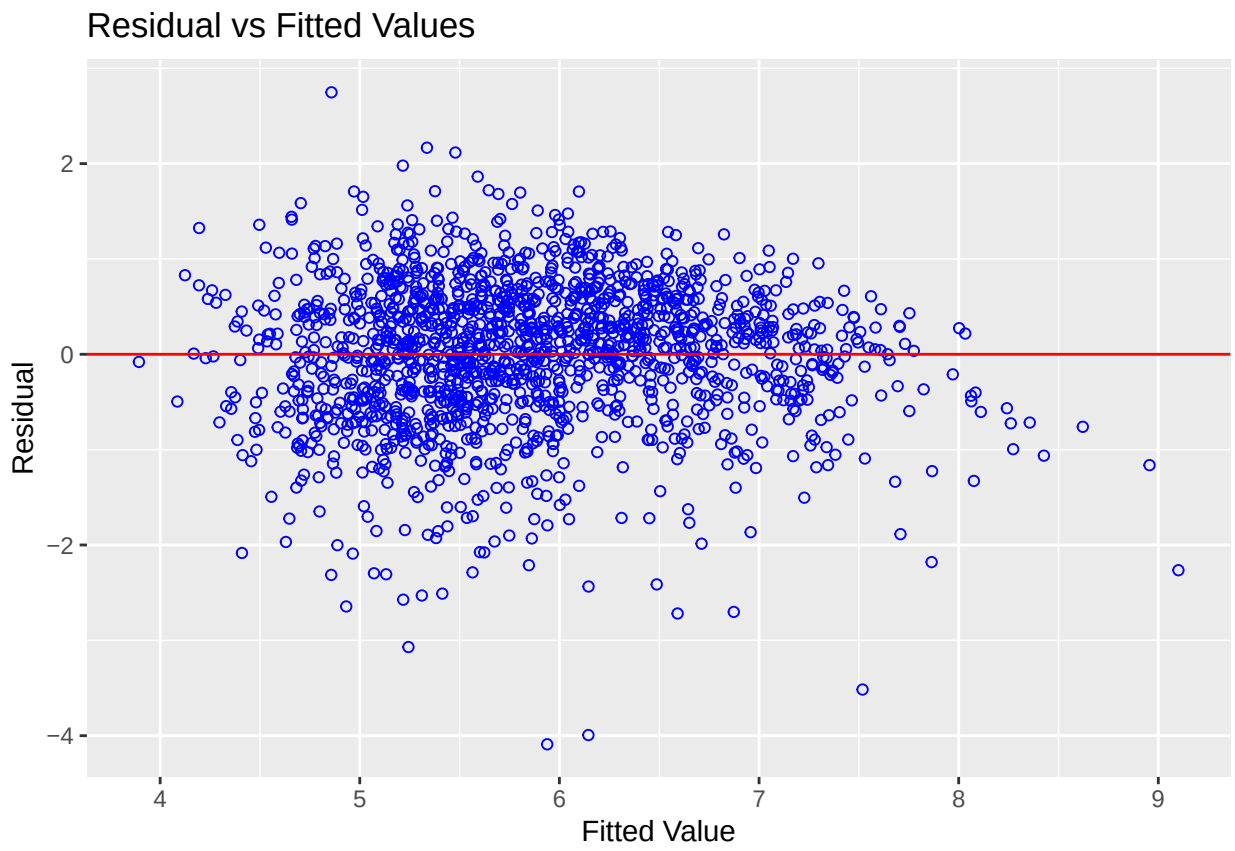
```
par(mfrow = c(2, 2))

#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")

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

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

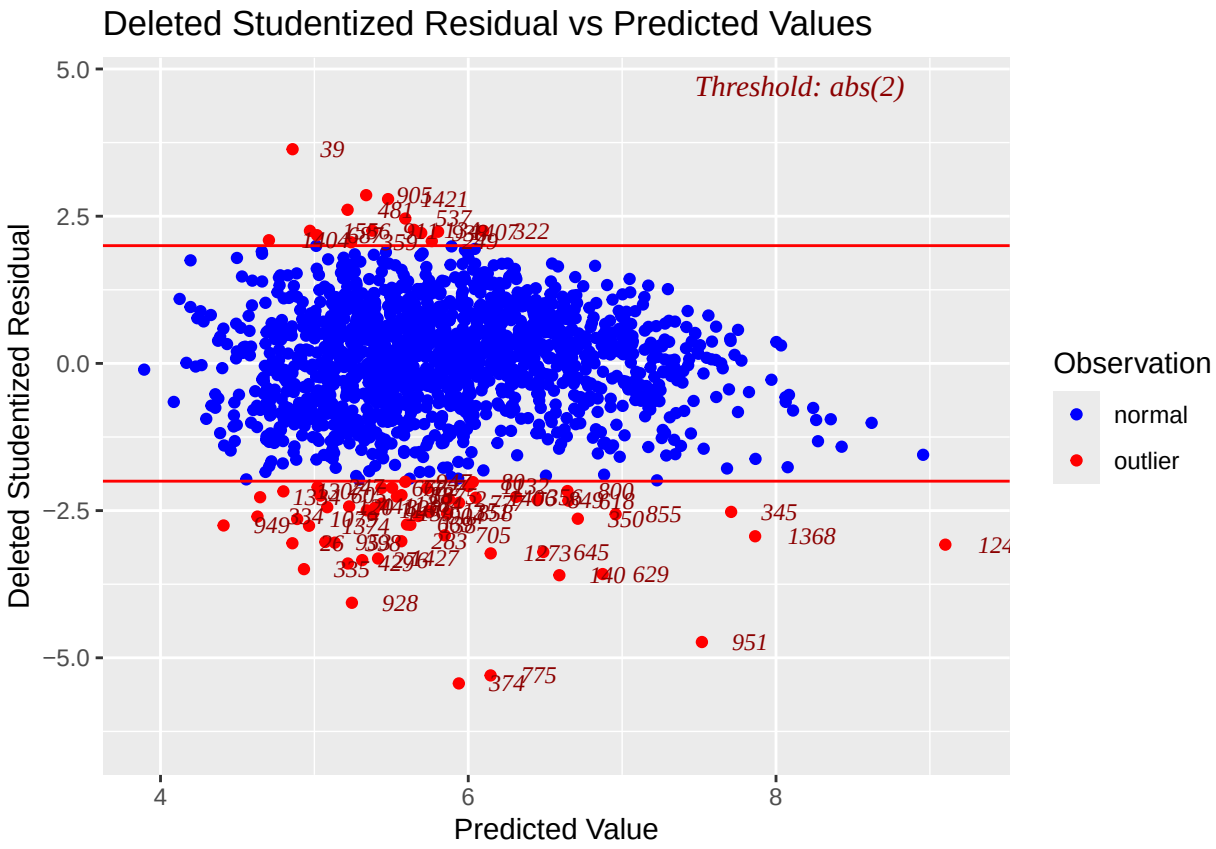


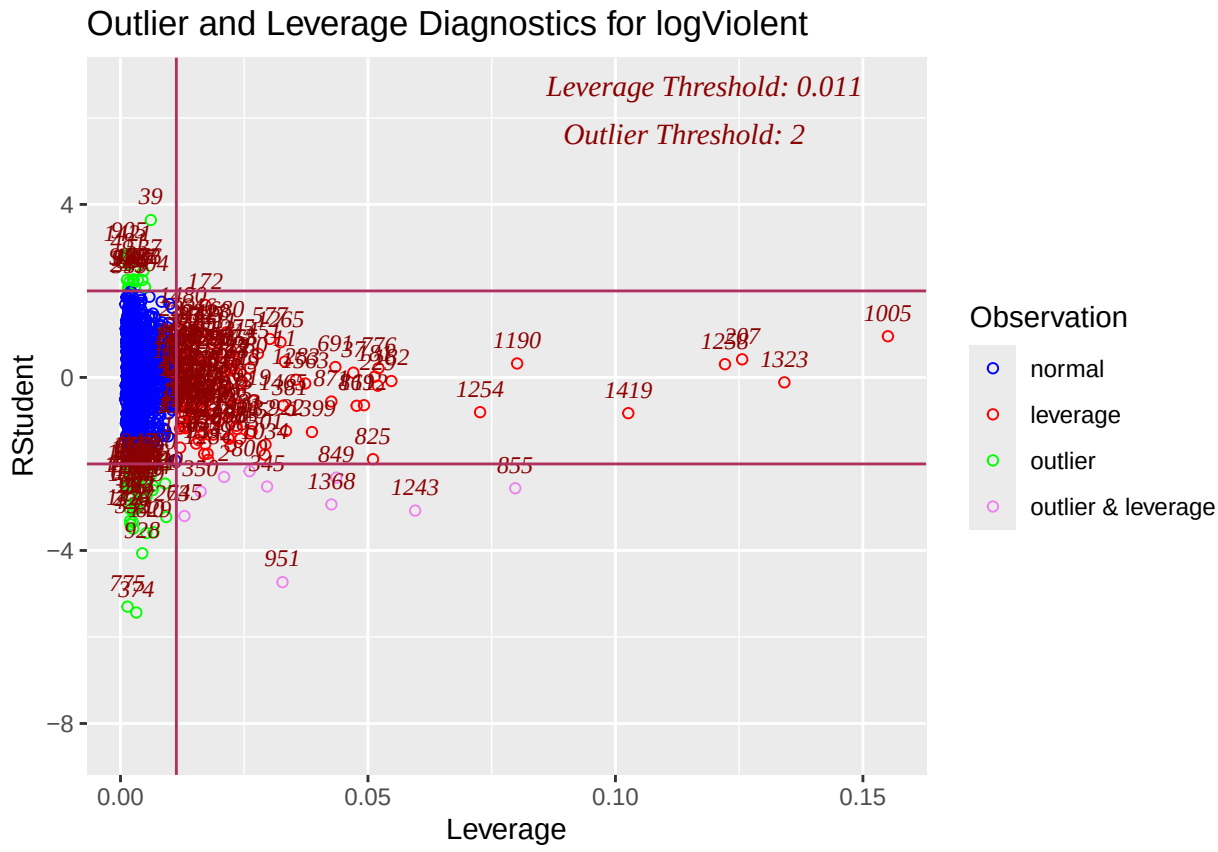
diagnostics

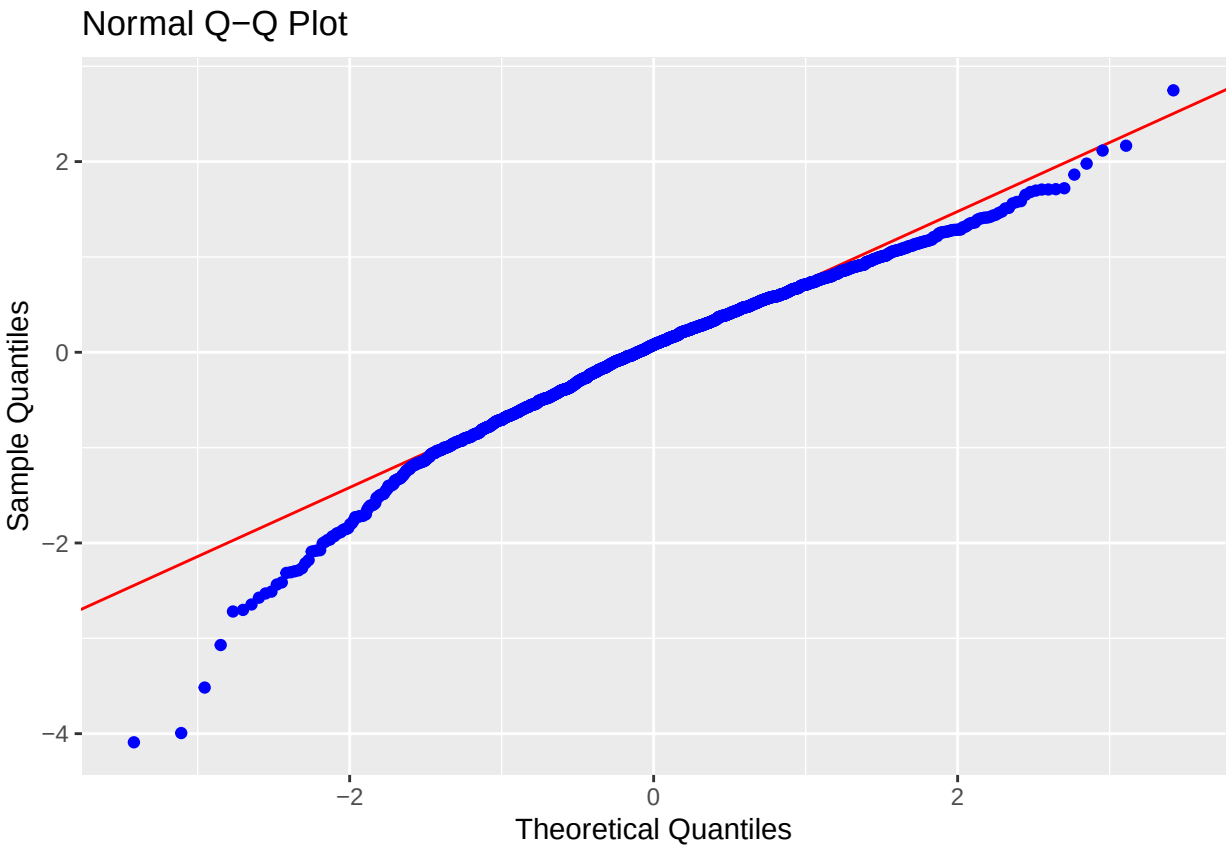
deze kan ook al eerder gebruikt worden, die geeft echt alles en moet je niet apart dingen gaan berekenen zoals hieronder/boven (uit een van zijn pc labs). hierin kan je goed zien dat er toch redelijk wat influential outliers zijn, dus dat robust wel is.

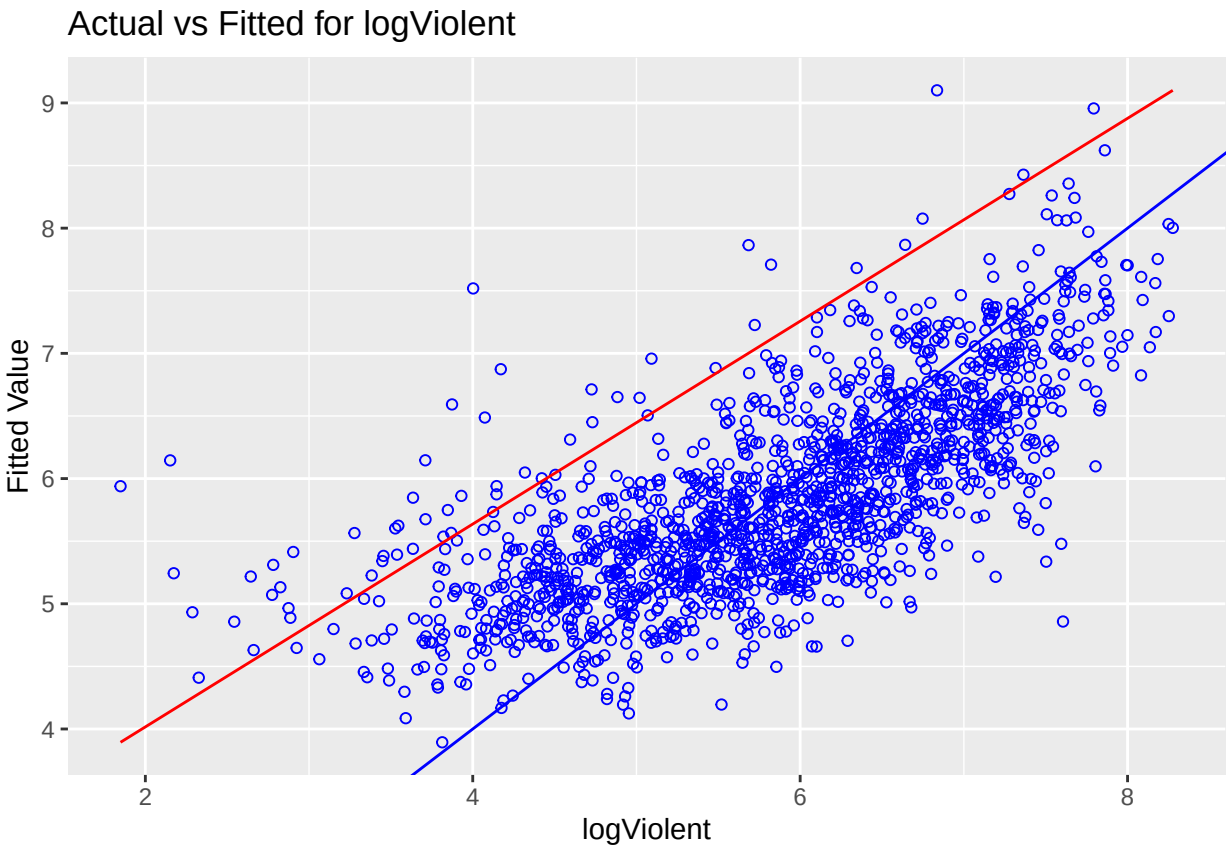
```
library(olsrr)
diagplots <- olsrr::ols_plot_diagnostics(fit_final, print_plot = TRUE)
```

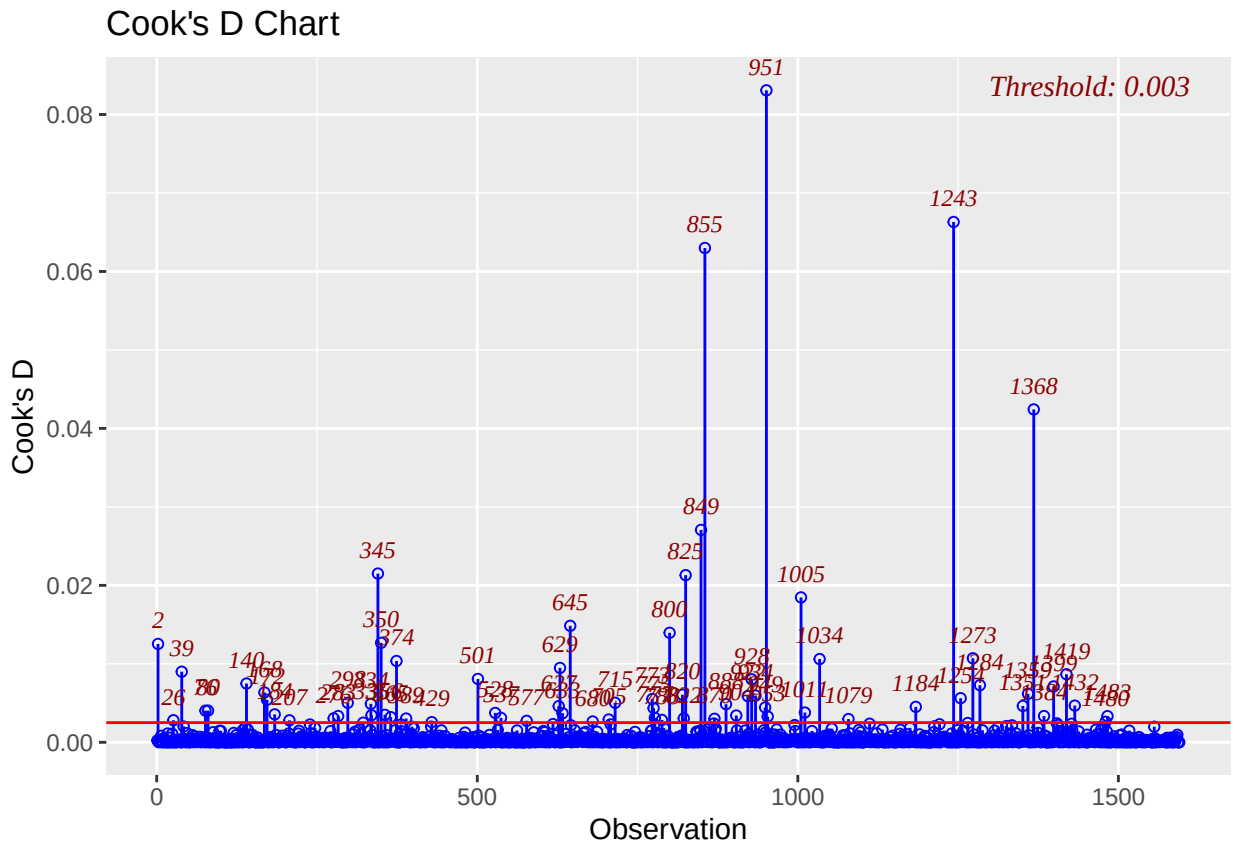

Residual vs Fitted Values
Residual
Fitted Value
2
0
−2
−4
4
5
6
7
8
9

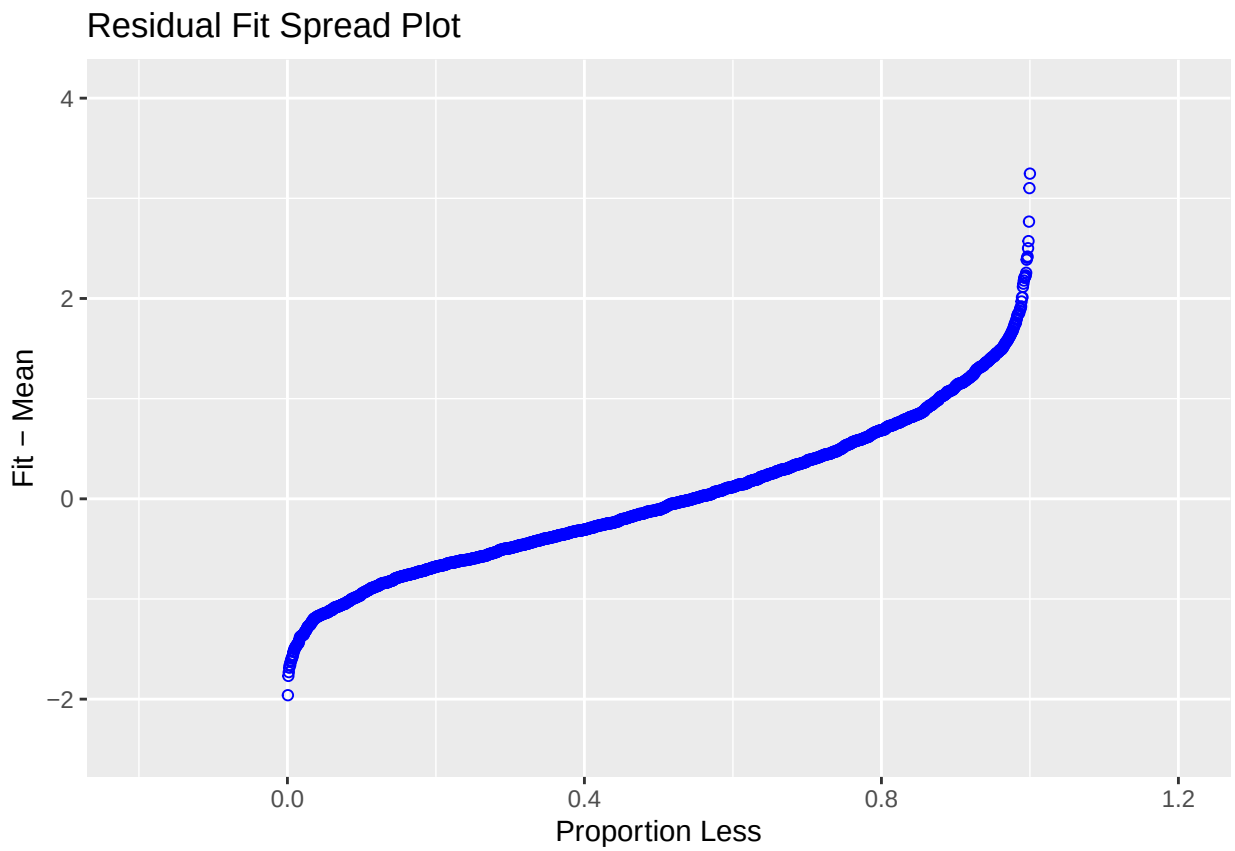

Deleted Studentized Residual vs Predicted Values
Threshold: abs(2)
Deleted Studentized Residual
Predicted Value
5.0
2.5
0.0
−2.5
−5.0
4
6
8
Observation
normal
outlier

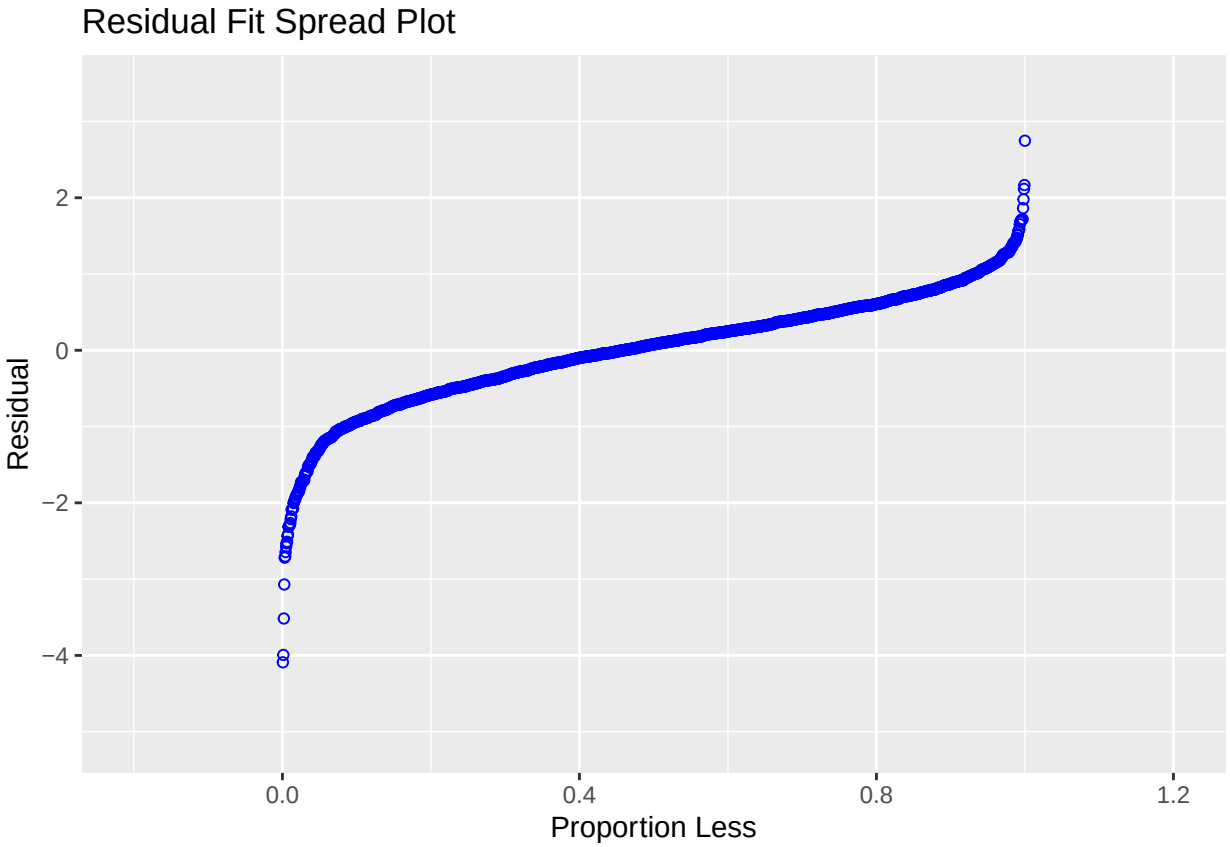

Outlier and Leverage Diagnostics for logViolent
Leverage Threshold: 0.011
Outlier Threshold: 2
RStudent
Leverage
4
0
−4
−8
0.00
0.05
0.10
0.15
Observation
normal
leverage
outlier
outlier & leverage

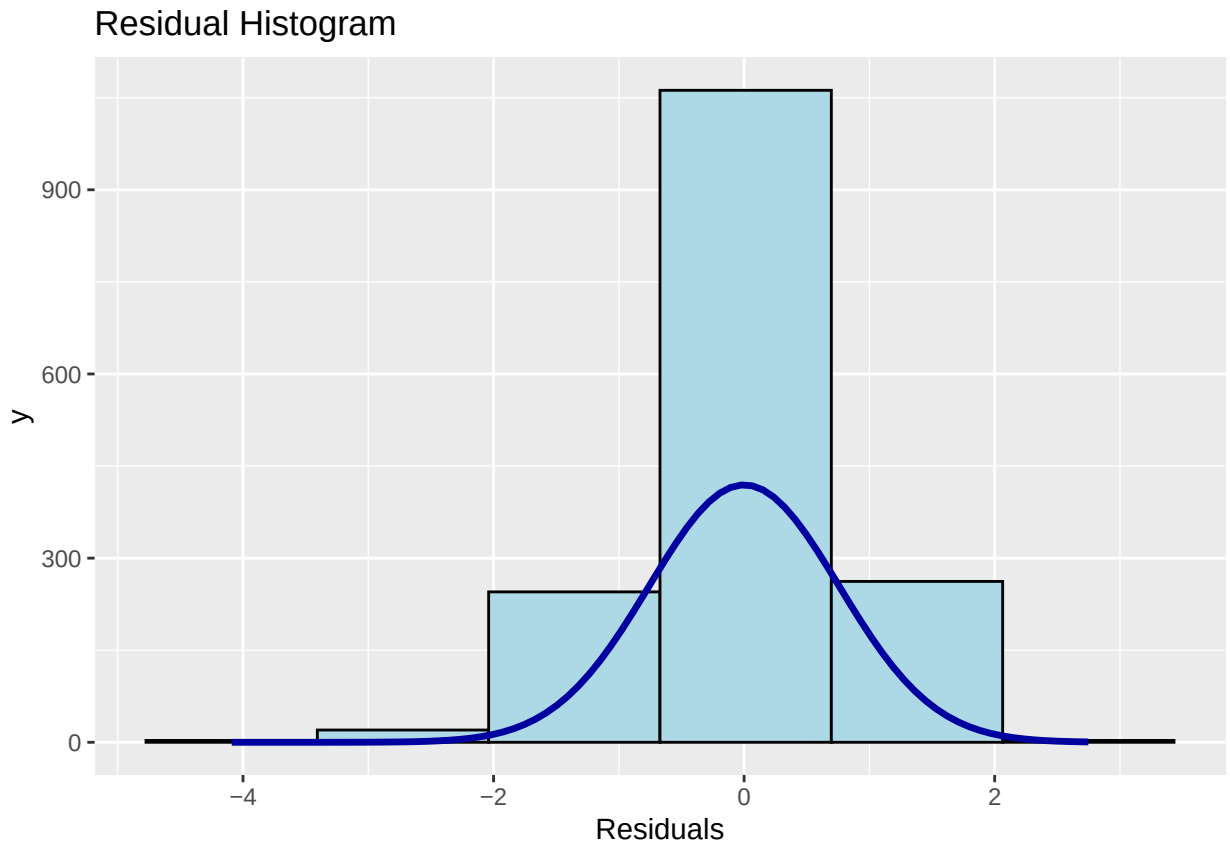

Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
2
0
−2
−4
−2
0
2

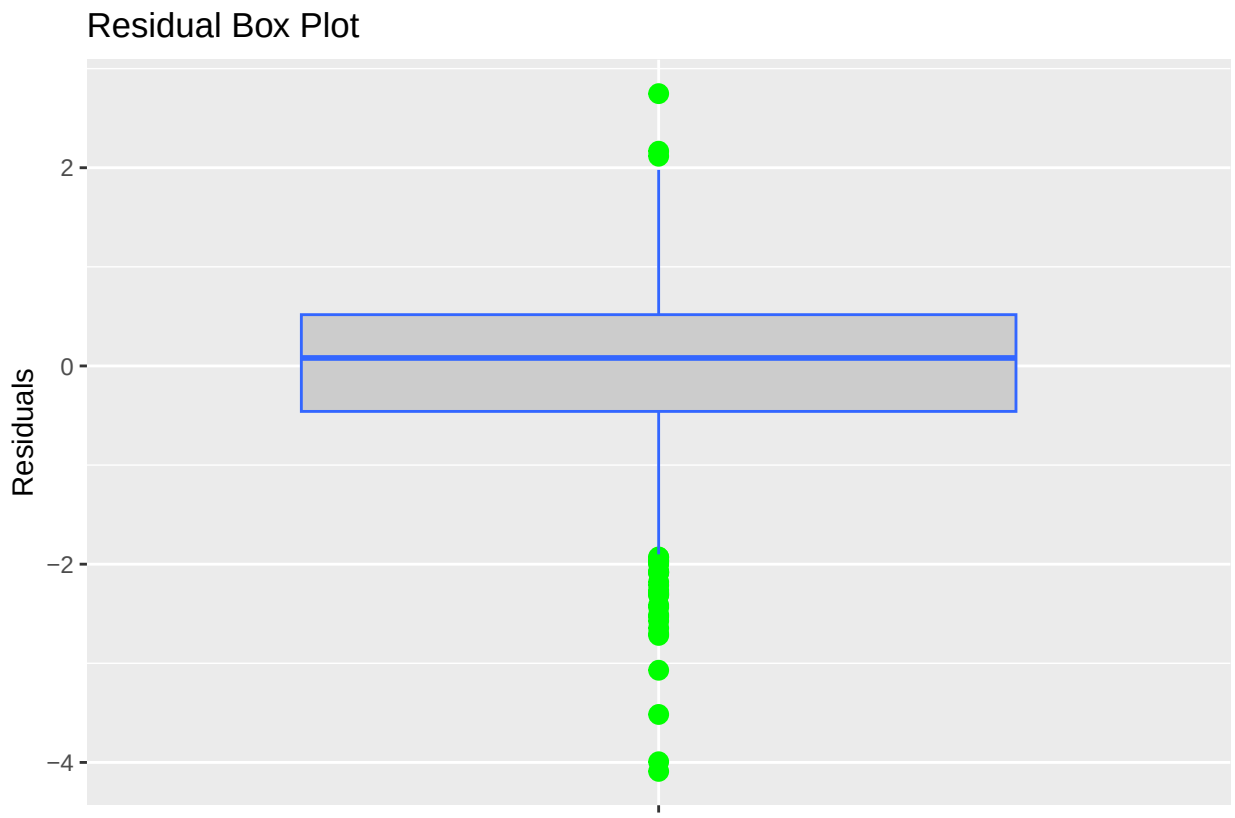

Actual vs Fitted for logViolent
Fitted Value
logViolent
9
8
7
6
5
4
2
4
6
8


Cook's D Chart
Threshold: 0.003
Cook's D
Observation
0.00
0.02
0.04
0.06
0.08
0
500
1000
1500
951
855
1243
1368
849
345
825
1005
645
800
1034
2
350
374
39
629
501
1273
1184
1079


Residual Fit Spread Plot
Fit − Mean
4
2
0
−2
0.0
0.4
0.8
1.2
Proportion Less


Residual Fit Spread Plot
Residual
2
0
−2
−4
0.0
0.4
0.8
1.2
Proportion Less


Residual Histogram
900
600
300
0
y
−4
−2
0
2
Residuals



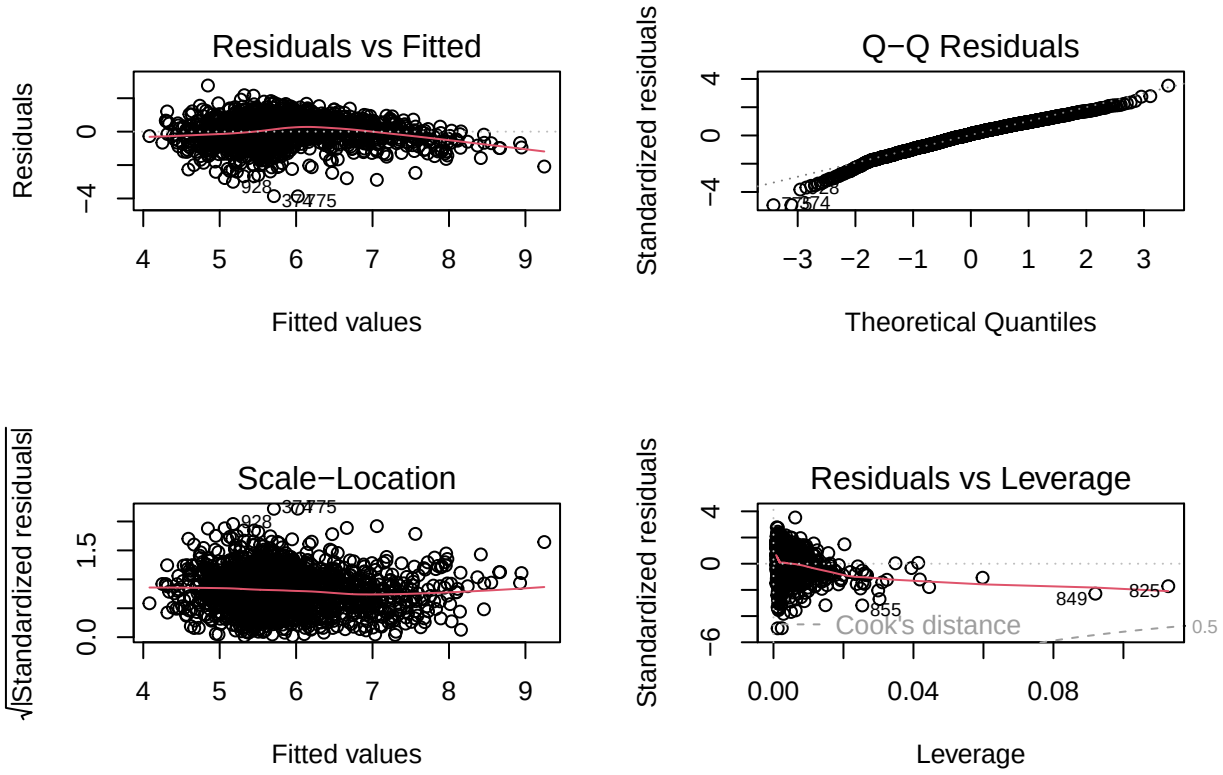
Diagnostic plots (figuur) reveal several potential outliers and high-leverage points. To assess the impact of these points, we fit a robust regression using Bisquare weights.

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

robust <- rlm(formula(fit_final), data = train_data, psi = psi.bisquare)
summary(robust, method = "XtX")
```

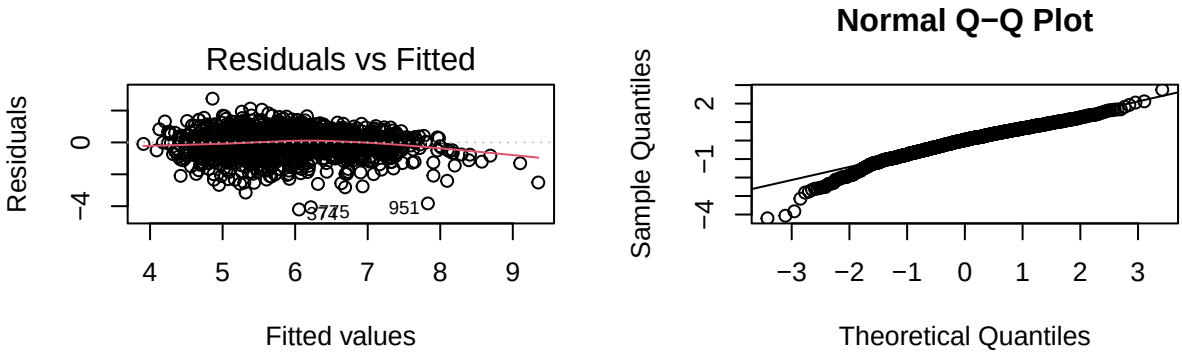
```
##
## Call: rlm(formula = formula(fit_final), data = train_data, psi = psi.bisquare)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.20460 -0.48682  0.03906  0.46842  2.74494
##
## Coefficients:
##                                Value    Std. Error t value
## (Intercept)                     5.2632   0.0945    55.6715
## PctPopUnderPov                  0.0766   0.0044    17.2271
## PctLess9thGrade                -0.0203   0.0079    -2.5767
## PctBSorMore                    -0.0247   0.0021   -12.0074
## racepctblack                    0.0546   0.0033    16.5034
## PctImmig                        0.0499   0.0048    10.4649
## PctPopUnderPov:racepctblack    -0.0013   0.0001    -8.7039
## PctPopUnderPov:PctLess9thGrade -0.0007   0.0003    -2.3283
```

```
## PctPopUnderPov:PctImmig          -0.0008    0.0003    -2.6411
##
## Residual standard error: 0.714 on 1586 degrees of freedom
```

```
par(mfrow=c(2,2))
plot(fit_log)
```




```
par(mfrow=c(2,2))
plot(robust, which = 1)

qqnorm(resid(robust))
qqline(resid(robust))
```



A comparison of the coefficients reveal no substantial differences, indicating that the influential points are not the result of gross errors. . The identified leverage-points can be called 'good leverages' as they exhibit high leverage but their residuals are still small, this even improves the precision of the regression coefficients. (zie referentie)

```
# 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)                     5.2664  5.2632     0.0032
## PctPopUnderPov                  0.0684  0.0766    -0.0082
## PctLess9thGrade                -0.0165 -0.0203     0.0037
## PctBSorMore                    -0.0248 -0.0247    -0.0001
## racepctblack                    0.0495  0.0546    -0.0051
## PctImmig                        0.0511  0.0499     0.0012
## PctPopUnderPov:racepctblack    -0.0010 -0.0013     0.0003
## PctPopUnderPov:PctLess9thGrade -0.0007 -0.0007     0.0000
## PctPopUnderPov:PctImmig        -0.0008 -0.0008     0.0000
```

Robustness checks (dit mag dan weg)

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

```
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)                    5.26256287  0.10602635  49.6345 < 2.2e-16 ***
## PctPopUnderPov                 0.07508904  0.00612368  12.2621 < 2.2e-16 ***
## PctLess9thGrade               -0.01901434  0.00791013  -2.4038  0.016226 *
## PctBSorMore                   -0.02472039  0.00232174 -10.6474 < 2.2e-16 ***
## racepctblack                   0.05344722  0.00499720  10.6954 < 2.2e-16 ***
## PctImmig                       0.05007152  0.00489164  10.2362 < 2.2e-16 ***
## PctPopUnderPov:racepctblack   -0.00124511  0.00018804  -6.6214 3.559e-11 ***
## PctPopUnderPov:PctLess9thGrade -0.00071184  0.00032706  -2.1765  0.029520 *
## PctPopUnderPov:PctImmig       -0.00075974  0.00026686  -2.8470  0.004414 **
## ---
## 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

```
# 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)                    5.2664  5.2626     0.0038
## PctPopUnderPov                 0.0684  0.0751    -0.0067
## PctLess9thGrade               -0.0165 -0.0190     0.0025
## PctBSorMore                   -0.0248 -0.0247     0.0000
## racepctblack                   0.0495  0.0534    -0.0039
## PctImmig                       0.0511  0.0501     0.0010
## PctPopUnderPov:racepctblack   -0.0010 -0.0012     0.0002
## PctPopUnderPov:PctLess9thGrade -0.0007 -0.0007     0.0000
## PctPopUnderPov:PctImmig       -0.0008 -0.0008     0.0000
```

Outlier and Influence Diagnostics (dit nog houden eventueel)

We use several diagnostic measures to identify influential observations

```
# 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,
  # zowel studentized deleted residuals als studentized residuals insteken?
  # DFBETAS ook insteken (per beta kan je dat bepalen) (chapter 10 Diagnostics slide 40)
  stud_residual = stud_res_final,
  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): 71
```

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

```
## High leverage observations (h > 0.0113 ): 145
```

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

```
## High Cook's distance (D > 0.0025 ): 70
```

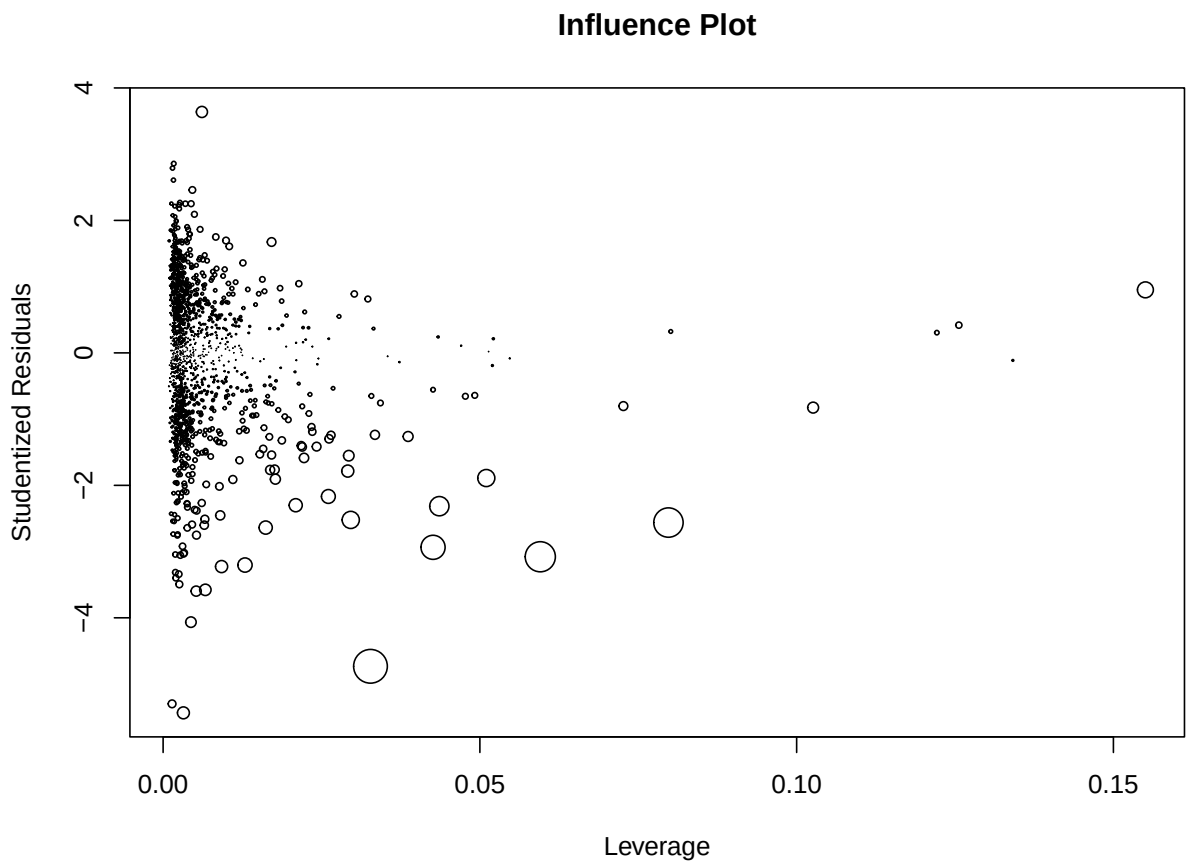
```
cat("High DFFITS (|DFFITS| >", round(dffits_threshold, 4), "):",
    sum(diagnostics$high_dffits), "\n")
```

```
## High DFFITS (|DFFITS| > 0.1502 ): 70
```

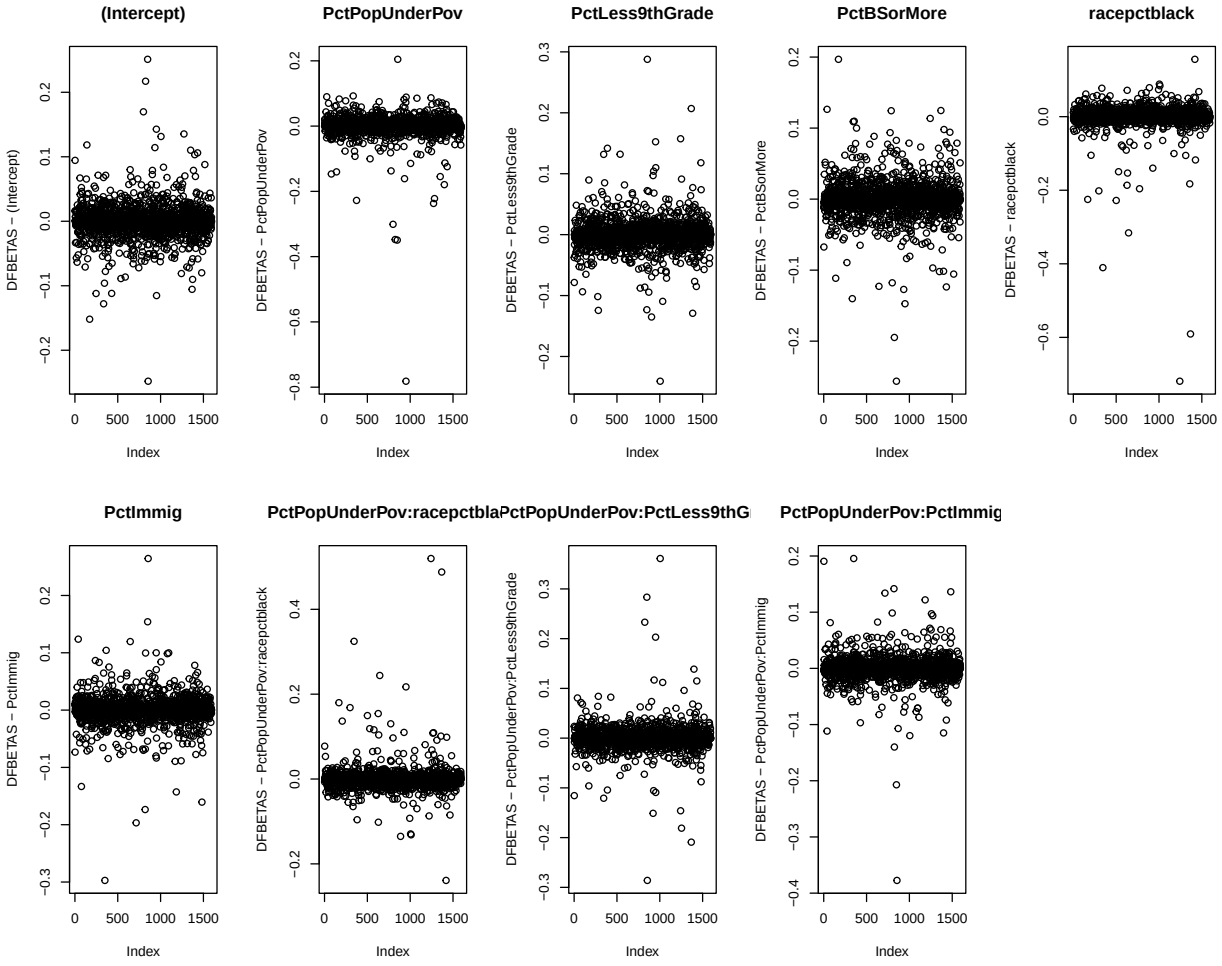
```
# find influent obs
influential <- diagnostics[diagnostics$high_cooks | diagnostics$high_dffits, ]
influential <- influential[order(-influential$cooks_d), ]
print(head(influential[, c("obs", "population", "stud_residual", "leverage", "cooks_d", "dffits")], 10))
```

```
##       obs population stud_residual   leverage    cooks_d     dffits
## 951   951      14302    -4.7321941 0.03273226 0.08307930 -0.8705169
## 1243 1243      15745    -3.0783821 0.05953685 0.06630277 -0.7745413
## 855   855      12822    -2.5618769 0.07976467 0.06298903 -0.7542479
## 1368 1368      36291    -2.9357492 0.04261123 0.04241797 -0.6193510
## 849   849      34590    -2.3147241 0.04360223 0.02706664 -0.4942359
## 345   345      12131    -2.5220282 0.02961827 0.02149854 -0.4406142
## 825   825      21265    -1.8898923 0.05101305 0.02129850 -0.4381750
## 1005 1005      12694     0.9516408 0.15505187 0.01846615  0.4076585
## 645   645      54052    -3.2032856 0.01292627 0.01484376 -0.3665705
## 800   800      12915    -2.1679362 0.02608589 0.01395480 -0.3548044
```

```
# 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])
}
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 4: Comparison of Simple and Final Multivariate Models

Model	R_squared	Adj_R_squared	MSE
Simple (PctPopUnderPov only)	0.2765	0.2761	258295.21
Final Multivariate	0.5220	0.5196	0.58

validation -> moet nog aangepast worden

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.

```
# 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): 0.58
```

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

```
## MSPR (test set): 828987.8
```

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

```
## Ratio MSPR/MSE: 1441205
```

```
# Predictions on test set
test_data$logViolent <- log(test_data$ViolentCrimesPerPop + 1)
pred <- predict(fit_final, newdata = test_data)

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

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

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

```
## MSE (training set): 0.58
```

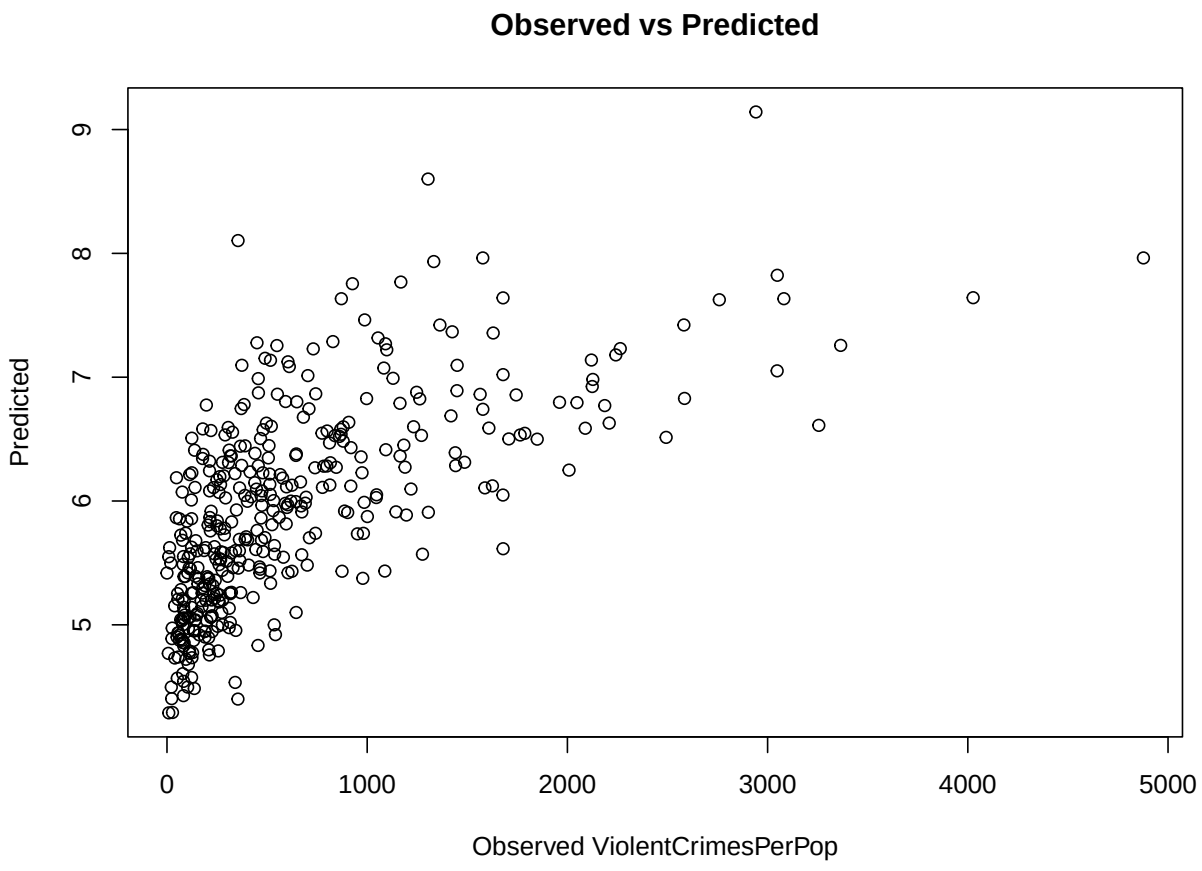
```
cat("MSPR (test set):", round(mspr, 2), "\n")
```

```
## MSPR (test set): 0.7
```

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

```
## Ratio MSPR/MSE: 1.2153
```

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



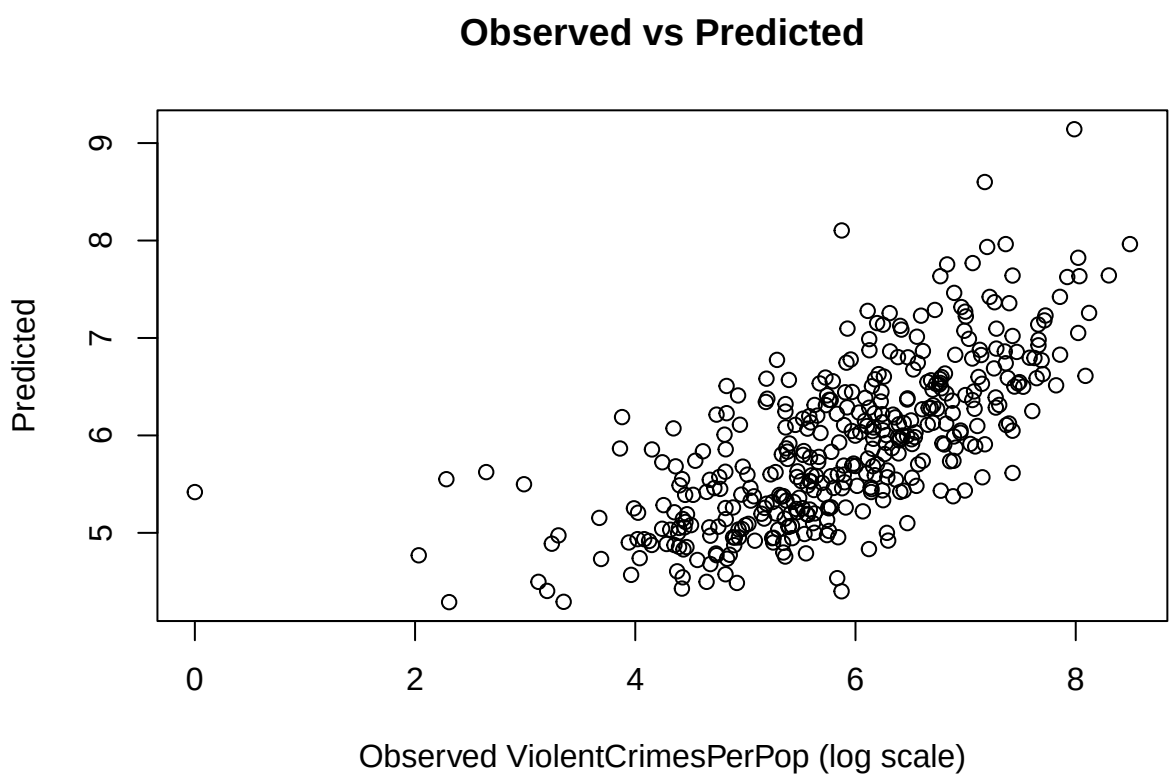

```
# R-squared on test set
ss_res <- sum((test_data$ViolentCrimesPerPop - pred)^2)
ss_tot <- sum((test_data$ViolentCrimesPerPop - mean(test_data$ViolentCrimesPerPop))^2)
r2_test <- 1 - ss_res/ss_tot
cat("\nR² on test set:", round(r2_test, 4), "\n")
```

```
##
## R² on test set: -0.7949
```

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

```
## R² on training set: 0.522
```

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




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

```
##
## R² on test set (log scale): 0.4728
```

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

```
## R² on training set (log scale): 0.522
```

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.

Rousseeuw, P. J., & van Zomeren, B. C. (1990). Unmasking Multivariate Outliers and Leverage Points. Journal of the American Statistical Association, 85(411), 633–639. https://doi.org/10.1080/01621459.1990.10474920