Analysis of Continuous Data project

Thomas Sertijn, Ilja Van Bever, Lieselot Van de Putte

2025-11-09

Research question

For this research the effect of socio-economic disadvantage to violent crime rates was investigated. More specifically we want to explore the association between poverty and violent crime rates in the USA. Becker (1968) stated that the decision to commit crime is a rational choice where people weigh the benefits and costs of committing crime against each other. It could be argued that the incentive to commit crime is higher for people who have a lower income, as they have more to gain and less to lose by committing crimes. Following this, we would then also expect that communities with higher poverty rates will also be associated with higher crime rates.

Design of the dataset

For this research, a dataset related to data gathered for the prediction of serious crime rates in the US is used. This 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.

Methods

For the purpose of our research question we selected a subset of variables present in in the dataset. Some economic variables: *PctPopUnderPov*, *perCapInc*, and *PctEmploy*, which give some information about economic status of the community, and a sociological subset, containing education levels in the communities, combined with *NummImmig*, *RacePctBlack*, *AgePct12t29*, all variables giving some info about the social composition and demographic structure of the communities.

1. To gain an initial understanding of the association between poverty rates (*PctPopUnderPov*) and violent crime rates (*ViolentCrimesPerPop*), a univariate regression is performed.
2. To analyze if the relationship is confounded by other socio-economic variables, whether there are relevant interaction effects at play and to see if the inclusion of these other socio-economic variables improves the performance of our model substantially, a multivariate regression is performed.

Before building the models, some descriptive analysis of the dataset and variables is performed. Next, the dataset is randomly split into a training dataset (80% of the data) and a holdout set (20% of the data).

For the multivariate model, an All-variable procedure is emplored to test every combination of possible variables as a model. The best model is chosen based on the Bayesian Information Criterion (BIC). Next, the assumptions of this best model are checked and partial regression plots are generated to check if each of the added variables are in the correct functional form. Afterwards, the most appropriate interaction term with the main predictor *PctPopUnderPov* is selected based on the same BIC procedure. Finally, multicollinearity is assessed using the Variance Inflation Factor (VIF) to obtain a more interpretable model. After determining the final model, some model diagnostics are calculated to determine outliers and influential points. Finally the model is validated by the holdout dataset.

Data preparation

The variable *NumImmig* is converted to a percentage of the total population, since the outcome variable *ViolentCrimesPerPop* (total number of violent crimes per 100K population) is expressed relative to the population size. We call this converted variable *PctImmig*. 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.

By examining the missing data, there were 221 (of the 2215) observations identified as NA values for the outcome variable *ViolentCrimesPerPop*. No imputations were used in order to have a more transparent model building procedure and to avoid overestimating the quality of the model fit. Therefore these observations were not taken into account for the model building. It can be noted that the variables *countyCode* and *communityCode* are also frequently unknown.

Descriptive analysis

Univariate descriptive analysis is performed, for both the missing values and the non-missing values. Neither summary statistics nor visualisations of the predictor distributions (like boxplots and histograms) indicate that the missing data have characteristics that differ substantially from the non-missing data.

To understand how the variables in the dataset relate to one another, correlation coefficients were calculated and scatter plots were examined to visualize the relationships between the variables, the outcome variable (*ViolentCrimesPerPop*), and the main predictor (*PctPopUnderPov*).

It is notable that the variable *racepctblack* shows the strongest correlation with the outcome variable ($r = 0.63$), even surpassing *PctPopUnderPov*, the main predictor selected for this study. Given some strong correlations between predictor variables (for example between *PctNotHSGrad* and *PctLess9thGrade* ($r = 0.93$), between *perCapInc* and *PctBSorMore* ($r = 0.77$), and between *PctNotHSGrad* and *PctBSorMore* ($r = -0.75$)) and the linear relationships some variables have with *PctPopUnderPov*, it is recommended to assess multicollinearity during the model-building stage.

The scatter plots show that most variables display a roughly linear relationship with *ViolentCrimesPerPop*, although that trend is often distorted in the extreme regions of the x-axis. *agePct12t29* has a very low correlation coefficient with *ViolentCrimesPerPop* (0.11). The variable *perCapInc* shows a higher correlation (-0.32), but there is clearly no linear trend present.

Two extreme values for the *ViolentCrimesPerPop* variable appear. Chestercity has the highest number of violent crimes (4877 violent crimes per 100K population). Spencercity, on the other hand, reports zero violent crimes. Given that Spencercity is a small community (11,066 inhabitants), it is difficult to assess the accuracy of the reported value. In the model diagnostics section, we will examine whether these two observations are outliers with respect to our linear model.

It was also investigated whether or not small communities have a higher probability to have more extreme values of the response and predictor variables. Scatter plots show that communities with a very small population indeed show a very large spread for all variables.

Model Building

Univariate linear regression

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

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

Here a coefficient of 38.87 could be found which represents the expected increase in violent crimes per 100K population if poverty rate increases by one percentage point. The R-squared value of 0.2701 represents the proportionate reduction of total variation in *ViolentCrimesPerPop* by the poverty percentage, in this univariate model this R-squared is rather small and can be taken to mean that poverty by itself is insufficient to explain the variation in violent crime rates. For the simple linear model the assumptions of linearity are violated. Larger outcome values tend to be underestimated, the variance for larger outcome values is larger and the QQ-plot shows violation of the normality assumption. It could also be seen that larger populations tended to have larger residuals, meaning that the model tends to underestimate the total number of crimes for large populations.

Figures to get a visual illustration of whether outliers are more common in small pop. We see larger residuals for larger populations. This means that the model tends te underestimate the total number of crimes for large populations.

Multivariate model

The first all-variable procedure resulted as model in:

$$ViolentCrimesPerPop_i = \beta_0 + \beta_1 \cdot PctPopUnderPov_i + \beta_2 \cdot PctLess9thGrade_i + \beta_3 \cdot PctNotHSGrad_i + \beta_4 \cdot PctImmig_i + \epsilon_i$$

However for this model, the assumptions of normality and equal error variances of the error terms are again violated.

Applying the log transformation offered a substantial improvement of our model showing approximate constant variances and residuals reasonably close to normal. The reduction in R-squared suggests that the model-fit was inflated due to heteroscedasticity.

Using the log transform Y variable to select a new model resulted in a similar model as before but with the *PctBSorMore* variable instead of the *PctNotHSGrad*. In this model, it could be observed from the partial regression plots that *PctRaceBlack* was not in the correct functional form. For this reason, a logit transformation was attempted which resulted in a more linear relation on the partial regression plots. Fitting the model with this logit transformed *racepctblack* seemed to also stabilize the residual vs fitted plot. This was followed by selecting a interaction term for the model, which showed that the best interaction given the BIC values would be between the main predictor *PctPopUnderPov* and *PctLess9thGrade*. However, this interaction introduced a high Vif value for both predictors and the interaction term possibly due to the presence of multiple education variables. Opting us to only take one education variable going forward. This resulted in a similar model with the *PctLess9thGrade* changed for *agePct12t29*. This model showed to have as best interaction term *PctPopUnderPov:agePct12t29*. However, This interaction resulted in a High VIF for both variables and their interaction term. For that reason, the second best interaction term *PctPopUnderPov:PctBSorMore* was chosen in the final model. This resulted in a final model:

$$ViolentCrimesPerPop_i = \beta_0 + \beta_1 \cdot PctPopUnderPov_i + \beta_2 \cdot PctBSorMore + \beta_3 \cdot agePct12t29 + \beta_4 \cdot PctImmig_i + \beta_5 \cdot race_black_logi$$

Model diagnostics

After deciding on the model and the form of the variables, the effect of the outliers was determined. This was done based on the Deleted Studentized residuals, the Cook's distance, the leverage, DFFITS, and DFBetas. These values were plotted, and values crossing a certain threshold were visualized and seen as influential. If certain datapoints were seen in multiple plots, then these were most likely more influential and possibly problematic outliers.

Using these plots, many influential outliers could be discovered. In order to exclude any data points that were influential due to observation errors, the variable values of these points were extracted and manually checked to see whether these are observations errors, rare cases, or valid but extreme cases.

By looking at all 318 communities in the dataset that were flagged as problematic, meaning they crossed the threshold of one of the diagnostic tests, no evidence could be found of errors or anomalies that would necessitate removal. Most of these communities were small and exhibited extreme demographic or crime-related characteristics. For example, two communities, Martinsvillecity and Vidorcity, both flagged by all 5 diagnostic tests, were reported to not have any people of black color in their community. As these are both small communities, we have no reason to suspect this observation to be wrongly annotated. Another example is Spencercity, which was reported to have a *ViolentCrimesPerPop* of 0, while this is not common in our dataset, we lack evidence to say that this observation is erroneous. Other observations showed either a high or low value in one or more variables compared to the majority of communities, reflecting genuine heterogeneity rather than data quality issues.

These findings show that these flagged communities likely represent valid but extreme cases present in our dataset whose demographic or crime-related profiles differ substantially from the average. Therefore, there is no reason to discard any of them. To further assess and account for the impact of these points, we fit a robust regression using Bisquare weights. When using a robust model, the coefficients revealed nearly identical to those from the OLS, indicating that the influential points are not the result of gross errors. The small differences in coefficients show that these are already quite stable and thus that maybe the outliers and influential points did not distort the OLS model significantly.

Interpretation of the final model

Our final model is: «».

Notes on interpretation The intercept on its own is not interpretable as it would require all predictor variables (and interaction terms) to be zero. This is not in the scope of our model. If it were, the intercept of NA could be interpreted as the expected log of (violent crime rate per 100k + 1) when all predictor variables are 0. There is no use in interpreting interaction-term coefficients on their own, as they indicate the effect of one predictor on the other, but have no direct effect on its own.

Interpretation of *PctPopUnderPov* The effect of *PctPopUnderPov* varies with *PctBsorMore* through the interaction terms. This means that the effect of *PctPopUnderPov* is not simply given by its coefficient, but instead we can look at the estimated effect of *PctPopUnderPov* on the expected value of the logarithm of (violent crime rate per population of 100k + 1) as $0.02584 + 8 \times 10^{-4}$ * PctBSorMore. If we take for example the mean of the *PctBSorMore* to calculate the effect of *PctPopUnderPov* when the interacting variable is held at its average and all variables are held constant, we can find 0.04424. The confidence interval of this effect is [0.03776, 0.05161] With a confidence coefficient of .95 we estimate that the true effect of *PctPopUnderPov* (including its interaction) on the expected value of the logarithm of (violent crime rate per population of 100k + 1) per unit increase in *PctPopUnderPov* (1 percent) is somewhere between 0.03776 and 0.05161, when the interacting variable is held at its average and all other variables are held constant. The interaction effect is reinforcing: with increasing percentages of people with higher education, the effect of *PctPopUnderPov* on the logarithm of (violent crime rate per population of 100k + 1) gets larger.

Interpretation of *PctBSorMore* The interpretation for *PctBSorMore* is similar to the one above because of their interaction. The conditional effect on of *PctBSorMore* on the logarithm of violent crime rate is given by: -0.02596 + 8 × 10^{-4}* PctPopUnderPov. Similarly to before we can take the mean of *PctPopUnderPov* and estimate the conditional effect of *PctBSorMore* on the expected value of the logarithm of (violent crime rate per population of 100k + 1) per unit increase of *PctBSorMore*, when keeping poverty at its mean and all other variables constant: -0.02596 + 0.00922 (95% CI: [-0.02068, -0.01239]). Contrary to above, we now see a interference effect: for higher levels of *PctPopUnderPov* the negative effect of *PctBSorMore* becomes less negative (the slope becomes less steep). Thus, under the conditions stated above, if a higher percentage of the population has a Bachelors degree or higher, the logarithm of the (violent crime rate per population of 100k + 1) is expected to decrease and this effect is less pronounced for higher levels of *PctPopUnderPov*.

Interpretation of *race_black_logit* *race_black_logit* has no interaction terms so we can interpret the main effect on its own. A one-unit increase in the log-odds of percentage African American is associated with a 0.27348 change in the expected value of the logarithm of (violent crime rate per population of 100k + 1) when keeping all other variables constant; where a one-unit increase in the log-odds has to be interpreted as a multiplication of the odds of being African American by e (≈ 2.72). The 95% confidence interval is [0.25066, 0.2963].This is a positive association: for increasing levels of the log-odds of percentage African Americans, there is an expected increase of the logarithm of (violent crime rate per population of 100k + 1).

Interpretation of *PctImmig* The coefficient 0.03008 represents the expected increase in the logarithm of (violent crime rate per population of 100k + 1) for an increase of one percentage point in *PctImmig*, keeping all other variables constant (95% CI:0.02565, 0.0345). An increase in percentage of immigrants is thus associated with an increase in the mean of log (violent crime rate per population of 100k + 1).

Interpretation of *agePct12t29* For the percentage of people between the age of 12 and 29, we find a negative association with the logarithm of (violent crime rate per population of 100k + 1). This is expressed by a coefficient of -0.0257 with 95% confidence interval [-0.03484, -0.01653]. With 95% confidence we estimate that the true coefficient of *agePct12t29* is between -0.03484 and -0.01653 when all other variables are fixed. Increasing percentage of *agePct12t29* is associated with a decrease in the mean logarithm of (violent crime rate per population of 100k + 1). This negative association is not in line with our prior assumption that violent crimes would increase for this age group.

Model validation

Refitting the model on the test data

When the final model is fitted to the test set or the full dataset, the estimated coefficients do not differ much from those obtained when fitting the model to the training set. The more data the model is fitted on, the more significant the p-values become.

Prediction of the test data

For the multivariate model 97.2431078 % of the observed values fall within the prediction interval. This means that the prediction intervals are a bit too strict. For the univariate model 91.7293233 % of the observed values fall within the prediction interval. This means that the prediction intervals are a bit too lenient. The variance of the residuals is proportional to the magnitude of the outcome variable. The univariate model incorrectly assumes homoscedasticity in the residuals, causing prediction intervals to be too wide for low outcome values and too narrow for high outcome values. As a result, the lower bound of the prediction interval is often negative for low values. In contrast, the multivariate model (which models the log-transformed outcome variable) produces prediction intervals whose width is proportional to the magnitude of the outcome variable, which aligns better with reality. Therefore, the prediction intervals from the multivariate model are more useful than those from the univariate model.

When point estimates are compared to observed values, both the univariate and the multivariate model tends to underestimate large outcome values and tends to overestimate small outcome values. This statement holds for both the prediction of the training data and the prediction of the test data. When the mean squared prediction error (for test data) is compared with the mean squared error (for training data), it can be noticed that the predictive power of the model is better for the test data than for the training data. This indicates that there is no overfitting. The same trend is visible when the values for R^2 are compared.

The R^2 of the multivariate model is higher than the R^2 of the univariate model. However, these values are difficult to compare, because the outcome variable differs between the two models ($ViolentCrimesPerPop$ vs. $\log(ViolentCrimesPerPop)$) and back-transforming the predicted values of $\log(ViolentCrimesPerPop)$ does not yield the expected value for $ViolentCrimesPerPop$ (,except under strict assumptions regarding the residuals).

Statistical discussion

There is a large portion of missing data in our dataset (221 of the 2215 observations). The main source of missing data is the *ViolentCrimesPerPop* variable, where 221 observations are missing assumably because of incomplete rape reporting in Midwestern States. This missingness could lead to biased estimates if these communities differ in crime-related or demographic characteristics. Though the characteristics of the communities with missing data are not much different from those of the non-missing data. We thus assume that this missingness does not have an important effect on our found results. However, if these communities are different in ways not captured by our measured variables, there could still be bias present. Moreover, the LEMAS survey includes all communities with police departments of more than 100 officers, but only a random sample of smaller departments. As a result, the representation of small communities in the dataset is less precise.

Smaller communities have greater variance in both predictor and outcome variables than larger communities. This can partly be explained by the smaller denominator: in a small community, small changes in the numerator may produce substantial changes in the rate compared to small changes in big communities.

Another caveat is that the FBI's data is from 1995, while the Census' populations data containing demographic variables are from 1990. If the demographic variables of certain communities changed drastically during this period, the predictor variables would not represent these communities well in 1995 (when the actual crimes happened).

Then, it should be noted that the data are from 1990-1995. Since then, the social and economic environment have changed, implying that our estimated relationships would not hold today as crime rates, reporting practices… have evolved. Thus, these results should be carefully interpreted in today's context as the mechanisms driving these relationships are different in comparison to when the data collection happened.

At last, it is mentioned in the description of the original dataset that many relevant factors are not included. Specifically, it is mentioned that per capita crime rates are calculated using resident populations, so communities with large numbers of visitors are expected to have higher crime rates, even if the actual risk is not different across communities. This, together with other unmeasured factors, implies that our results should be interpreted as correlations and not as causations.

Conclusion

In this report, we investigate how violent crime rates and socio-economic disadvantage, and more specifically the poverty rate, are associated in U.S. communities. For this purpose we used a merged dataset with detailed data on both crimes and demographic variables from 1990 and 1995.

The univariate linear regression showed a significant positive relationship between the percentage poverty and the number of crimes per 10k population. For the multivariate model, that modelled the logarithm of *ViolentCrimesPerPop*, a subset of five additional (socio-economic) predictor variables was selected. The following predictor variables show a significant positive association with the number of crimes per 10k population (*ViolentCrimesPerPop*): *PctPopUnderPov*, *race_black_logit* and *PctImmig*. The following predictor variables showed a significant negative association with the number of crimes per 10k population: *PctBSorMore* and *agePct12t29*. The selected interaction term is *PctPopUnderPov:PctBSorMore* and has a significant positive association with the number of crimes per 100k population, which means that a higher percentage of inhabitants with a bachelor degree or more implies a stronger association between poverty and the number of crimes per 100k population. «We assume that this is because a high poverty rate in combination with a high educational attainment rate could imply there to be the a large level of inequality present in the community. This inequality then may be associated with higher violent crime rates (Kelly (2000), Fajnzylber et al. (2002) and Kang (2016)).» The multivariate model predicts the logarithm of the number of crimes per 100k population with an R^2 of 0.5328 on the training dataset and an R^2 of 0.5841 on the test dataset. The modelling of the logarithm is very useful to give good prediction intervals, because in this way the heteroskedasticity of *ViolentCrimesPerPop* is considered, but makes the interpretation of the model less convenient and makes it difficult to deliver good point estimates for *ViolentCrimesPerPop*.

Several limitations should be highlighted though. The dataset contains missing data, is not very recent and uses outdated demographic predictors. Keeping this context in mind, our results show that the number of violent crimes is associated with socio-economic disadvantage. It's important to mention that all results are interpreted as correlational and not as causal. The theoretical framework, which is essential for establishing causal relationships, is not further developed in this statistical study. Moreover, the number of predictors included in the analysis is very limited, which means confounding could play a significant role.

References

Becker GS (1968) Crime and Punishment: An Economic Approach. Journal of Political Economy 76: 169–217

Fajnzylber, P., Lederman, D., & Loayza, N. (2002). Inequality and violent crime. The journal of Law and Economics, 45(1), 1-39.

Kang, S. (2016). Inequality and crime revisited: effects of local inequality and economic segregation on crime. Journal of Population Economics, 29(2), 593-626.

Kelly, M. (2000). Inequality and crime. Review of economics and Statistics, 82(4), 530-539.

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)

Additional references

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

Source - https://stackoverflow.com/a

Posted by DonJ, modified by community. See post 'Timeline' for change history

Retrieved 2025-12-03, License - CC BY-SA 3.0

Appendix

Figures

Contributions

Code

```
knitr::opts_chunk$set(message = FALSE, warning = FALSE,
                      echo = FALSE, include = FALSE)
library(knitr)
library (glue)
library(dplyr)
library(data.table)
library(dplyr)
library(stringr)
library(rvest)
library(ggplot2)
library(gtsummary)
library(sjlabelled)
library(tidyr)
library(ggcorrplot)
library(patchwork)
library(MASS)
library(stdmod)
violent_crimes_table <- fread("curl https://archive.ics.uci.edu/static/public/211/communities+and+crime+

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 --
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, "^[^:]+")
colnames(violent_crimes_table) <- var_names
crimes_table_subset <- violent_crimes_table %>%
  dplyr::select(communityname,state,countyCode, communityCode, fold, population,
         PctPopUnderPov, perCapInc, PctEmploy, PctLess9thGrade, PctNotHSGrad, PctBSorMore,
         NumImmig, racepctblack, agePct12t29, ViolentCrimesPerPop
         )
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
crimes_table_subset %>%
  pivot_longer(cols = where(is.numeric), names_to = "variable", values_to = "value") %>%
  group_by(variable) %>%
  summarise(
    NAs = sum(is.na(value))
  )
na_subset <- crimes_table_subset %>%
  filter(is.na(ViolentCrimesPerPop)
  )
na_subset <- na_subset[,-'ViolentCrimesPerPop']
na_subset
crimes_table_subset = na.omit(crimes_table_subset)
colSums(crimes_table_subset == "?", na.rm = TRUE)
#str(crimes_table_subset)
#summary(crimes_table_subset)
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(),
```

```r
    NAs = sum(is.na(value))
  )
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))
  )
numeric_cols_na <- sapply(na_subset, is.numeric)
crimes_table_subset_num_na <- na_subset[, ..numeric_cols_na]
crimes_table_subset$logpopulation <- log10(crimes_table_subset$population)
numeric_cols <- sapply(crimes_table_subset, is.numeric)
crimes_table_subset_num <- crimes_table_subset[, ..numeric_cols]
par(mfrow = c(4,2))

plots <- lapply(names(crimes_table_subset_num_na), function(columnname) {
  column <- crimes_table_subset_num[[columnname]]
  boxplot(column,
          main = columnname
          )
  hist(column,
       main = columnname,
       xlab = get_label(column)
          )
}
)
numeric_cols_na <- sapply(na_subset, is.numeric)
crimes_table_subset_num_na <- na_subset[, ..numeric_cols_na]
par(mfrow = c(4,2))

plots <- lapply(names(crimes_table_subset_num_na), function(columnname) {
  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)
          )
}
)
cor_matrix <- cor(crimes_table_subset_num[,-c('population', 'logpopulation')])
cor_values <- as.data.frame(as.table(cor_matrix))
```

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

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


df <- crimes_table_subset_num[,-c("population", "logpopulation")]
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))

df <- crimes_table_subset_num[,-c("population", "ViolentCrimesPerPop", "logpopulation")]
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))
crimes_table_subset[order(crimes_table_subset_num$ViolentCrimesPerPop, decreasing=FALSE), , drop = FALS
df <- crimes_table_subset_num
x_var <- "logpopulation"
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(x = "log(population)", y = str_wrap(y_var, width = 45))
}
)

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

```r
set.seed(987654321)
n <- nrow(crimes_table_subset_num)
training <- sample(1:n, size = floor(0.8 * n))
train_data <- crimes_table_subset[training, ]
test_data <- crimes_table_subset[-training, ]
n_training <- nrow(train_data)

cat("Training set size:", n_training, "\n")
cat("Test set size:", nrow(test_data), "\n")
fit_simple <- lm(ViolentCrimesPerPop ~ PctPopUnderPov, data = train_data)
summary(fit_simple)

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

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

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

# Confidence intervals for coefficients
cat("\n95% Confidence Intervals:\n")
print(confint(fit_simple))

# ANOVA
anova(fit_simple)
par(mfrow = c(2, 2))

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

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

# QQ-plot of residuals (normality)
qqnorm(fit_simple$residuals, main = "Normal Q-Q Plot of Residuals")
qqline(fit_simple$residuals, col = "red")
```

```r
# Studentized residuals
stud_res <- rstudent(fit_simple)
plot(fit_simple$fitted.values, stud_res,
     xlab = "Fitted values", ylab = "Studentized Residuals",
     main = "Studentized Residuals vs Fitted")
outliers_simple <- which(abs(stud_res) > 2)
par(mfrow = c(1, 2))
# Residuals vs population
plot(train_data$logpopulation, fit_simple$residuals,
     xlab = "log(population)", ylab = "Residuals")
abline(h = 0, col = "red", lty = 2)
lines(lowess(train_data$logpopulation, fit_simple$residuals), col = "blue")

plot(train_data$logpopulation, stud_res,
     xlab = "log(population)", ylab = "Studentized Residuals")
abline(h = 0, col = "red", lty = 2)
lines(lowess(train_data$logpopulation, stud_res), col = "blue")
par(mfrow = c(1, 1))
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)
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), ]
# Print best model
```

```r
best <- which.min(model_ranking$BIC)
best_pred <- model_ranking$formula[best]
cat("\nBest model:\n")
cat(best_pred, "\n")
cat("BIC:", round(model_ranking$BIC[best], 2), "\n")
cat("Adjusted R²:", round(model_ranking$adj.r.square[best], 4), "\n")

fit_multi <- lm(best_pred, data = train_data)
multi_var_summary <- summary(fit_multi)
multi_var_summary
par(mfrow = c(2, 2))

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

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

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

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

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

multi_var_summary <- summary(fit_log)
multi_var_summary
plot(fit_log)
par(mfrow = c(1, 1))
train_data$logViolent <- log(train_data$ViolentCrimesPerPop + 1)
test_data$logViolent <- log(test_data$ViolentCrimesPerPop + 1)
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("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$BIC), ]
# Print best model
best <- which.min(model_ranking$BIC)
best_pred <- model_ranking$formula[best]
cat("\nBest model:\n")
cat(best_pred, "\n")
cat("BIC:", round(model_ranking$BIC[best], 2), "\n")
cat("Adjusted R²:", round(model_ranking$adj.r.square[best], 4), "\n")

fit_multi <- lm(best_pred, data = train_data)
multi_var_summary <- summary(fit_multi)
multi_var_summary
predictor_list <- row.names(multi_var_summary$coefficients[-1,])

partial_regression_plot <- function(data, outcome, predictor, predictor_list) {
  temp <- 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)
```

```r
  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, " ~ controls", sep = ""))
}
plots <- lapply(predictor_list, function(predictor) {
    partial_regression_plot(train_data, "logViolent", predictor, predictor_list)
}
)

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

pct_to_logit <-  function(percentage, addition = 1e-6) {
  percentiles_in_range01 <- percentage / 100
  # add small value to avoid 0 and 1
  percentiles_in_range01 <- pmin(
    pmax(percentiles_in_range01, addition),
    1 - addition)
  logit <- log(percentiles_in_range01 /
                                     (1 - percentiles_in_range01))
  logit
}

train_data$race_black_logit <- pct_to_logit(train_data$racepctblack)
test_data$race_black_logit <- pct_to_logit(test_data$racepctblack)
fit_logit <- lm(logViolent ~
                PctPopUnderPov +
                PctLess9thGrade +
                PctBSorMore +
                race_black_logit +
                PctImmig,
              data = train_data)


multi_var_logit_summary <- summary(fit_logit)
multi_var_logit_summary
plot(fit_logit)
par(mfrow = c(1, 1))
predictor_list <- row.names(multi_var_logit_summary$coefficients[-1,])
plots <- lapply(predictor_list, function(predictor) {
    partial_regression_plot(train_data, "logViolent", predictor, predictor_list)
}
)

# Print 9 plots per pg
```

```r
print(wrap_plots(plots, ncol = 3))
selected_vars <- setdiff(
  row.names(multi_var_logit_summary$coefficients),
  c("(Intercept)", "PctPopUnderPov")
)

interaction_terms <- paste("PctPopUnderPov", selected_vars, sep = ":")

formulas_interaction <- sapply(interaction_terms, function(i) {
  paste("logViolent ~", paste(predictor_list, collapse = " + "), "+",
                                  i)
})

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

interaction_models_form <- vapply(formulas_interaction, 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

best_interaction <- row.names(interaction_table[1,])
# Estimate model with interaction
formula_final <- as.formula(paste("logViolent ~",
                                  paste(predictor_list, collapse = " + "), "+",
                                  "PctPopUnderPov:PctLess9thGrade"))
fit_final <- lm(formula_final, data = train_data)
summary(fit_final)
library(car)
# practicum: car::vif(fit)
```

```r
vif <- vif(fit_final)
print(vif)
cat("BIC: ", BIC(fit_final))

filtered_formulas <- Filter(function(f) {
  variables <- all.vars(f)
  number_education_variables <- sum(variables %in% educ)
  number_education_variables <= 1
}, formulas)
#change formulas to contain the logit based race variable
filtered_formulas <- lapply(filtered_formulas, function(f) {
  f_str <- deparse(f)                                # formula -> character vector
  f_str <- paste(f_str, collapse = " ")              # collapse into one string
  f_str <- gsub("racepctblack", "race_black_logit", f_str)  # replace variable name
  as.formula(f_str)                                  # back to formula
})



models <- lapply(filtered_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(filtered_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$BIC), ]
# Print best model
best <- which.min(model_ranking$BIC)
best_pred <- model_ranking$formula[best]
cat("\nBest model:\n")
cat(best_pred, "\n")
cat("BIC:", round(model_ranking$BIC[best], 2), "\n")
cat("Adjusted R²:", round(model_ranking$adj.r.square[best], 4), "\n")

fit_multi <- lm(best_pred, data = train_data)
multi_var_summary <- summary(fit_multi)
multi_var_summary
predictor_list <- row.names(multi_var_summary$coefficients[-1,])
selected_vars <- setdiff(
  row.names(multi_var_summary$coefficients),
  c("(Intercept)", "PctPopUnderPov")
)

interaction_terms <- paste("PctPopUnderPov", selected_vars, sep = ":")
```

```r
formulas_interaction <- sapply(interaction_terms, function(i) {
  paste("logViolent ~", paste(predictor_list, collapse = " + "), "+",
                                  i)
})

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

interaction_models_form <- vapply(formulas_interaction, 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

best_interaction <- row.names(interaction_table[1,])
formula_final <- as.formula(paste("logViolent ~",
                                  paste(predictor_list, collapse = " + "),  "+",
                                  best_interaction))
fit_final <- lm(formula_final, data = train_data)
summary(fit_final)
plot(fit_final)
par(mfrow = c(1, 1))
vif <- vif(fit_final)
print(vif)
cat("BIC: ", BIC(fit_final))
formula_final <- as.formula(paste("logViolent ~",
                                  paste(predictor_list, collapse = " + "),  "+",
                                  "PctPopUnderPov:PctBSorMore"))
fit_final <- lm(formula_final, data = train_data)
final_sums <- summary(fit_final)
final_sums
```

```r
plot(fit_final)
par(mfrow = c(1, 1))
vif <- vif(fit_final)
print(vif)
cat("BIC: ", BIC(fit_final))
predictor_list <- row.names(multi_var_summary$coefficients[-1,])
plots <- lapply(predictor_list, function(predictor) {
    partial_regression_plot(train_data, "logViolent", predictor, predictor_list)
}
)

# Print 9 plots per pg
print(wrap_plots(plots, ncol = 3))
# Compare models
cat("Comparison of models:\n")
cat("Original model adjusted R²: ", round(summary(fit_final)$adj.r.squared, 4), "\n") ## dit moet nog a
cat("interaction term model adjusted R²:  ", round(summary(fit_final)$adj.r.squared, 4), "\n")

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")

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

par(mfrow = c(1, 2))
# Residuals vs population
plot(train_data$logpopulation, fit_final$residuals,
     xlab = "log(population)", ylab = "Residuals")
abline(h = 0, col = "red", lty = 2)
lines(lowess(train_data$logpopulation, fit_final$residuals), col = "blue")

plot(train_data$logpopulation, stud_res_final,
     xlab = "log(population)", ylab = "Studentized Residuals")
abline(h = 0, col = "red", lty = 2)
lines(lowess(train_data$logpopulation, stud_res_final), col = "blue")

par(mfrow = c(1, 1))
p <- length(coef(fit_final))
n_train <- nrow(train_data)
n_test <- nrow(test_data)

library(olsrr)
olsrr::ols_plot_cooksd_bar(fit_final)
olsrr::ols_plot_resid_lev(fit_final, threshold = NULL, print_plot = TRUE)
olsrr::ols_plot_resid_stud_fit(fit_final, threshold = NULL, print_plot = TRUE)
# IMPORTANT remark: the labelled observations are the points with 2 times the threshold (the reason is
dffits_vals <- data.frame(dffits(fit_final))

# Plot
plot <- ggplot(dffits_vals) +
  geom_point(aes(x = 1:nrow(dffits_vals), y = dffits_vals[[1]])) +
  geom_segment(aes(x = 1:nrow(dffits_vals), xend = 1:nrow(dffits_vals), y = 0, yend = dffits_vals[[1]])
               color = 'cornflowerblue') +
  geom_hline(yintercept = c(2, -2) / sqrt(nrow(dffits_vals)), color = 'salmon') +
  labs(x = 'Observation index', y = 'DFFITS', title = 'DFFITS',
       subtitle = 'Thresholds are at \u00B1 2\u221A(p/n)') +
  geom_text(aes(x = 1:nrow(dffits_vals), y = ifelse(dffits_vals[[1]] > 0, dffits_vals[[1]] + .05, dffit
                label = ifelse(abs(dffits_vals[[1]]) > 2*2*sqrt(p/nrow(dffits_vals)),
                               paste0(round(dffits_vals[[1]], digits = 2), ' (', 1:nrow(dffits_vals), '
# Print plot
print(plot)
# IMPORTANT remark: the labelled observations are the points with 1,5 times the threshold (the reason i
# DFBETAS values (placing them in a data frame ensures easy compatibility with ggplot2)
dfbetas_vals <- data.frame(dfbetas(fit_final))

# Plot
colnames_dfbetas <- colnames(dfbetas_vals)
plots <- lapply(colnames_dfbetas[-1], function(predictor) {
ggplot(dfbetas_vals) +
  geom_point(aes(x = 1:nrow(dfbetas_vals), y = dfbetas_vals[[predictor]])) +
  geom_segment(aes(x = 1:nrow(dfbetas_vals), xend = 1:nrow(dfbetas_vals), y = 0, yend = dfbetas_vals[[p
               color = 'cornflowerblue') +
  geom_hline(yintercept = c(2, -2) / sqrt(nrow(dfbetas_vals)), color = 'salmon') +
  labs(x = 'Observation index', y = 'DFBETAS',
       title = paste0('DFBETAS values for coefficient of ', predictor),
       subtitle = 'Thresholds are at \u00B1(2\u00F7\u221An)') +
  geom_text(aes(x = 1:nrow(dfbetas_vals), y = ifelse(dfbetas_vals[[predictor]] > 0, dfbetas_vals[[predic
                label = ifelse(abs(dfbetas_vals[[predictor]]) > 1.5*(2/sqrt(nrow(dfbetas_vals))),
                               paste0(round(dfbetas_vals[[predictor]], digits = 2), ' (', 1:nrow(dfbetas
})
# Print plots
print(plots)
# Calculate diagnostics and tresholds
stud_res_final <- rstudent(fit_final)
leverage <- hatvalues(fit_final)
```

```r
cooks_dist <- cooks.distance(fit_final)
dffits_values <- dffits(fit_final)
dfbetas_valuues <- dfbetas(fit_final)


leverage_threshold <- 2 * p / n_train
dffits_threshold <- 2 * sqrt(p / n_train)
dfbetas_threshold <- 2 / sqrt(n_train)

# new dataframe with all diagnoistics
diagnostics <- data.frame(
  obs = 1:n_train,
  # population = train_data$population,
  stud_residual = stud_res_final,
  leverage = leverage,
  cooks_dist = cooks_dist,
  dffits = dffits_values
)

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

high_dfbetas <- apply(abs(dfbetas_valuues), 1, function(x) any(x > dfbetas_threshold))

diagnostics$high_dfbetas <- high_dfbetas

# Select flag columns
flag_cols <- c("outlier_residual", "high_leverage", "high_cooks", "high_dffits", "high_dfbetas")

# Count TRUE flags
diagnostics$flag_count <- rowSums(diagnostics[flag_cols])

# summary
cat("Outliers by studentized residuals (|r*| > 2):", sum(diagnostics$outlier_residual), "\n")
cat("High leverage observations (h >", round(leverage_threshold, 4), "):",
    sum(diagnostics$high_leverage), "\n")
cat("High Cook's distance (D >", round(4/n_train, 4), "):",
    sum(diagnostics$high_cooks), "\n")
cat("High DFFITS (|DFFITS| >", round(dffits_threshold, 4), "):",
    sum(diagnostics$high_dffits), "\n")
# Extract flagged observations
problematic <- diagnostics[
  diagnostics$outlier_residual |
  diagnostics$high_leverage |
  diagnostics$high_cooks |
  diagnostics$high_dffits |
  diagnostics$high_dfbetas , ]

problematic_sorted <- problematic[order(-problematic$flag_count), ]
```

```r
problematic_with_data <- merge(problematic_sorted, train_data, by.x = "obs", by.y = "row.names", all.x =

problematic_with_data <- problematic_with_data[order(-problematic_with_data$flag_count), ]

problematic_with_data[, c("flag_count", "communityname", "population", "PctPopUnderPov", "PctBSorMore",

library(MASS)

robust <- rlm(formula(fit_final), data = train_data, psi = psi.bisquare)
summary(robust, method = "XtX")
par(mfrow=c(2,2))
plot(fit_final)
par(mfrow=c(2,2))
plot(robust, which = 1)
qqnorm(resid(robust))
qqline(resid(robust))

# 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))
summary(fit_final)
print(confint(fit_final))

b <- coef(fit_final)

mean_bsormore <- mean(train_data$PctBSorMore)
mean_poverty <- mean(train_data$PctPopUnderPov)

coef_mean <- b["PctPopUnderPov"] +
      b["PctPopUnderPov:PctBSorMore"]     * mean_bsormore

confidence_intervals <- confint(fit_final)
ci_pov <- cond_effect_boot(fit_final,
            x = "PctPopUnderPov", w = "PctBSorMore", nboot = 2000)
ci_lower <- ci_pov$"CI Lower"[ci_pov$Level == "Medium"]
ci_upper <- ci_pov$"CI Upper"[ci_pov$Level == "Medium"]
ci_pov2 <- cond_effect_boot(fit_final,
            x = "PctBSorMore", w = "PctPopUnderPov", nboot = 2000)
ci_lower2 <- ci_pov2$"CI Lower"[ci_pov2$Level == "Medium"]
ci_upper2 <- ci_pov2$"CI Upper"[ci_pov2$Level == "Medium"]
crimes_table_subset$logViolent <- log(crimes_table_subset$ViolentCrimesPerPop + 1)
crimes_table_subset$race_black_logit <- pct_to_logit(crimes_table_subset$racepctblack)

fit_final_test <- lm(formula_final, data = test_data)
fit_final_total <- lm(formula_final, data = crimes_table_subset)
summary(fit_final_test)
df1 <- data.frame(
  fit_on_training_data = coef(fit_final),
  fit_on_test_data = coef(fit_final_test),
  fit_on_total_data = coef(fit_final_total),
  p_value_training = summary(fit_final)$coefficients[, 4],
  p_value_test = summary(fit_final_test)$coefficients[, 4],
  p_value_total = summary(fit_final_total)$coefficients[, 4]
)
df1
df2 <- data.frame(
  R_squared = c(summary(fit_final)$r.squared, summary(fit_final_test)$r.squared, summary(fit_final_tota
  Adjusted_R_squared = c(summary(fit_final)$adj.r.squared, summary(fit_final_test)$adj.r.squared, summa
  BIC_model = c(BIC(fit_final), BIC(fit_final_test), BIC(fit_final_total))
)
rownames(df2) <- c("fit on training data","fit on test data", "fit on all data")
df2
### Predictions on test set
# backtransformation of the data to the normal scale
backtransformation <- function(value){
  exp(value) - 1
}
pred_interval_test_log <- predict(fit_final, newdata = test_data, interval = "prediction", level = 0.95
pred_interval_train_log <- predict(fit_final, newdata = train_data, interval = "prediction", level = 0.
pred_interval_test <- apply(pred_interval_test_log, 1:2, backtransformation)
pred_interval_train <- apply(pred_interval_train_log, 1:2, backtransformation)
pred_interval_test_simple <- predict(fit_simple, newdata = test_data, interval = "prediction", level =
pred_interval_train_simple <- predict(fit_simple, newdata = train_data, interval = "prediction", level
inside_log <- test_data$logViolent >= pred_interval_test_log[, "lwr"] & test_data$logViolent <= pred_in
inside <- test_data$ViolentCrimesPerPop >= pred_interval_test[, "lwr"] & test_data$ViolentCrimesPerPop
inside_simple <- test_data$ViolentCrimesPerPop >= pred_interval_test_simple[, "lwr"] & test_data$Violen
cat("Fraction of datapoints in prediction interval (logscale):", mean(inside_log), "\n")
cat("Fraction of datapoints in prediction interval:", mean(inside), "\n")
cat("Fraction of datapoints in prediction interval (univariate model):", mean(inside_simple), "\n")

hist(pred_interval_test[, "upr"] - pred_interval_test[, "lwr"],
     main = "Width prediction interval",
     xlab = "Width prediction interval",
     xlim = c(0, 5000),
     breaks = 50)
plot(pred_interval_test[, "fit"], pred_interval_test[, "upr"] - pred_interval_test[, "lwr"], main = "Wi
     xlab = "Point estimate", ylab = "Width prediction interval")

hist(pred_interval_test_log[, "upr"] - pred_interval_test_log[, "lwr"],
     main = "Width prediction interval (log)",
     xlab = "Width prediction interval",
     #xlim = c(0, 15000),
     breaks = 50)
hist(pred_interval_test_simple[, "upr"] - pred_interval_test_simple[, "lwr"],
     main = "Width prediction interval (univariate model)",
     xlab = "Width prediction interval",
     #xlim = c(0, 15000),
     breaks = 50)
# prediction intervals test data logscale
df_prediction_test_logscale <- data.frame(observed = test_data$logViolent, predicted = pred_interval_te
```

```r
ggplot(df_prediction_test_logscale, aes(observed, predicted))+
  geom_point()+
  geom_smooth(se = FALSE)+
  geom_abline(intercept = 0, slope = 1, color = "red")+
  theme_bw() +
  labs(x = "number of violent crimes per 100K population (observed)",
       y = "number of violent crimes per 100K population (predicted)",
       title = "Test dataset (log scale)")

# prediction intervals training data logscale
df_prediction_train_logscale <- data.frame(observed = train_data$logViolent, predicted = pred_interval_

ggplot(df_prediction_train_logscale, aes(observed, predicted))+
  geom_point()+
  geom_smooth(se = FALSE)+
  geom_abline(intercept = 0, slope = 1, color = "red")+
  theme_bw() +
  labs(x = "number of violent crimes per 100K population (observed)",
       y = "number of violent crimes per 100K population (predicted)",
       title = "Train dataset (log scale)")

# prediction intervals test data simple
df_prediction_test_simple <- data.frame(observed = test_data$ViolentCrimesPerPop, predicted = pred_inte

ggplot(df_prediction_test_simple, aes(observed, predicted))+
  geom_point()+
  geom_smooth(se = FALSE)+
  geom_abline(intercept = 0, slope = 1, color = "red")+
  theme_bw() +
  labs(x = "number of violent crimes per 100K population (observed)",
       y = "number of violent crimes per 100K population (predicted)",
       title = "Test dataset (univariate)")

# prediction intervals training data simple
df_prediction_train_simple <- data.frame(observed = train_data$ViolentCrimesPerPop, predicted = pred_in

ggplot(df_prediction_train_simple, aes(observed, predicted))+
  geom_point()+
  geom_smooth(se = FALSE)+
  geom_abline(intercept = 0, slope = 1, color = "red")+
  theme_bw() +
  labs(x = "number of violent crimes per 100K population (observed)",
       y = "number of violent crimes per 100K population (predicted)",
       title = "Train dataset (univariate)")
# MSPR for test data
mspr_test_log <- mean((test_data$logViolent - pred_interval_test_log[, "fit"])^2)

# MSE for training data
mse_train_log <- sum((train_data$logViolent - pred_interval_train_log[, "fit"])^2)/(n_train - p)

cat("MSPR:", round(mspr_test_log, 2), "\n")
cat("MSE:", round(mse_train_log, 2), "\n")
cat("Ratio MSPR/MSE:", round(mspr_test_log/mse_train_log, 4), "\n")
```

```r
# R-squared on test data
ss_res_test <- sum((test_data$logViolent - pred_interval_test_log[, "fit"])^2)
ss_tot_test <- sum((test_data$logViolent - mean(test_data$logViolent))^2)
r2_test_log <- 1 - ss_res_test/ss_tot_test
cat("R² on test set:", round(r2_test_log, 4), "\n")

# R-squared on training data
ss_res_train <- sum((train_data$logViolent - pred_interval_train_log[, "fit"])^2)
ss_tot_train <- sum((train_data$logViolent - mean(train_data$logViolent))^2)
r2_train_log <- 1 - ss_res_train/ss_tot_train
cat("R² on training set:", round(r2_train_log, 4), "\n")

# Results for multivariate model backtransformed logscale
validation_logscale <- c("log(ViolentCrimesPerPop)", round(mspr_test_log, 2), round(mse_train_log, 2),
# MSPR for test data
mspr_test_simple <- mean((test_data$ViolentCrimesPerPop - pred_interval_test_simple[, "fit"])^2)

# MSPR for training data
mse_simple <- sum((train_data$ViolentCrimesPerPop - pred_interval_train_simple[, "fit"])^2)/(n_train -

cat("MSE:", round(mse_simple, 2), "\n")
cat("MSPR:", round(mspr_test_simple, 2), "\n")
cat("Ratio MSPR/MSE:", round(mspr_test_simple/mse_simple, 4), "\n")

# R-squared on test data
ss_res_test_simple <- sum((test_data$ViolentCrimesPerPop - pred_interval_test_simple[, "fit"])^2)
ss_tot_test <- sum((test_data$ViolentCrimesPerPop - mean(test_data$ViolentCrimesPerPop))^2)
r2_test_simple <- 1 - ss_res_test_simple/ss_tot_test
cat("R² on test set:", round(r2_test_simple, 4), "\n")

# R-squared on training data
ss_res_train_simple <- sum((train_data$ViolentCrimesPerPop - pred_interval_train_simple[, "fit"])^2)
ss_tot_train <- sum((train_data$ViolentCrimesPerPop - mean(train_data$ViolentCrimesPerPop))^2)
r2_train_simple <- 1 - ss_res_train_simple/ss_tot_train
cat("R² on training set:", round(r2_train_simple, 4), "\n")

# Results for univariate model
validation_simple <- c("ViolentCrimesPerPop", round(mspr_test_simple, 2), round(mse_simple, 2), round(ms

df_validation <- data.frame(multivar_logscale = validation_logscale, univar = validation_simple)
rownames(df_validation) <- c("outcome variable", "MSPR", "MSE", "Ratio MSPR/MSE", "R² (test)", "R² (tra
df_validation
```