ST362 Final Project

Aaryan Agrawal

2024-08-01

Loading Required Libraries

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
#install.packages("corrgram")
#install.packages("car")
library(corrgram)
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```
#install.packages("psych")
library(psych)
```

```
##
## Attaching package: 'psych'

## The following object is masked from 'package:car':
##
##     logit
```

```r
#install.packages("ggplot2")
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```r
crime <- read.csv("crime.csV")
crime <- na.omit(crime)
```

To improve the setup of our data for analysis, we will split the `City..State` column into two separate columns: `City` and `State.` This will provide an additional categorical variable (`State`) that can be utilized in our analysis. Additionally, we will rearrange and rename the data columns to ensure clarity and consistency in the dataset.

```r
crimeData <- separate(crime, col = 3, into = c("City", "State"), sep = ", ")
crimeData <- crimeData %>% rename(HPI = index_nsa , TotalCrime = Violent.Crimes)
crimeData <- crimeData %>% select(Year,City,State,HPI,Population,everything(),-TotalCrime,TotalCrime)
```

Introduction

In this study, we analyze crime data from various cities and states in the USA to identify significant predictors of different types of crimes and understand the underlying patterns and trends. By applying concepts learned from our coursework, we aim to build models that accurately represent the relationships between socio-economic factors and crime rates, specifically focusing on how these factors influence the housing price index. A significant part of our analysis is examining how population size and violent crime rates can be used to predict changes in the housing price index over multiple years for the same cities. Understanding the relationship between predictors like population size, violent crime rates, and the housing price index will provide a broader context for the data. This analysis will highlight the socio-economic factors that contribute to the fluctuations in the housing prices and how these factors interact with crime rates, understanding the influence of population changes on housing prices can help government agencies to develop new initiatives to manage urban growth, housing affordability and crime rate.

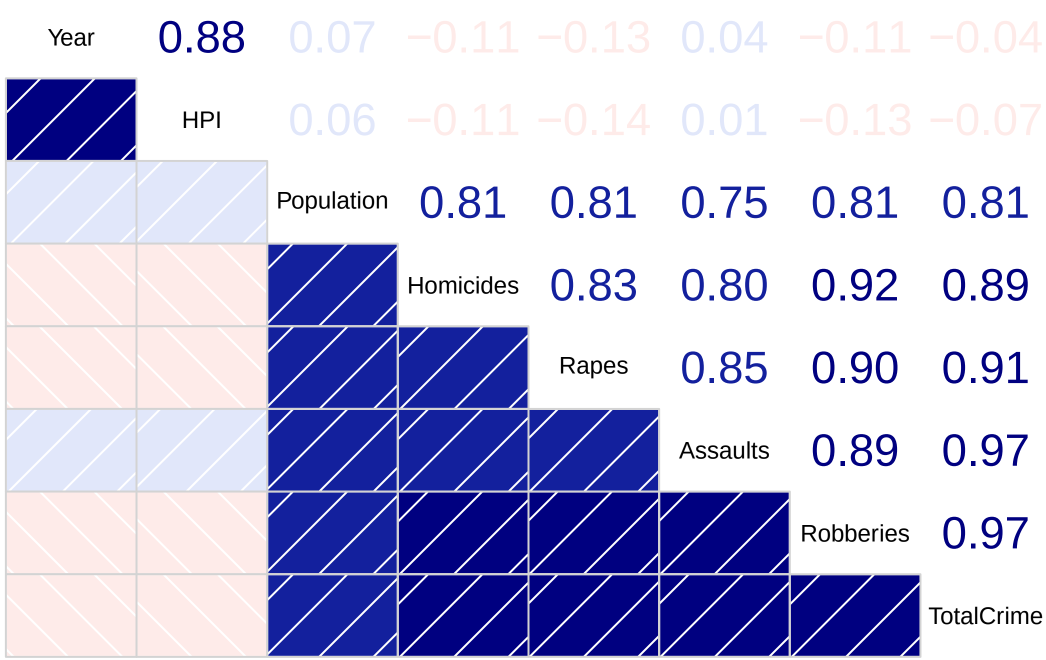
Exploratory Data Analysis

Correlations

```r
# Find correlations

# Method 1: Use a corrgram
corrgram(crimeData,upper.panel=panel.cor)
```

```
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
```

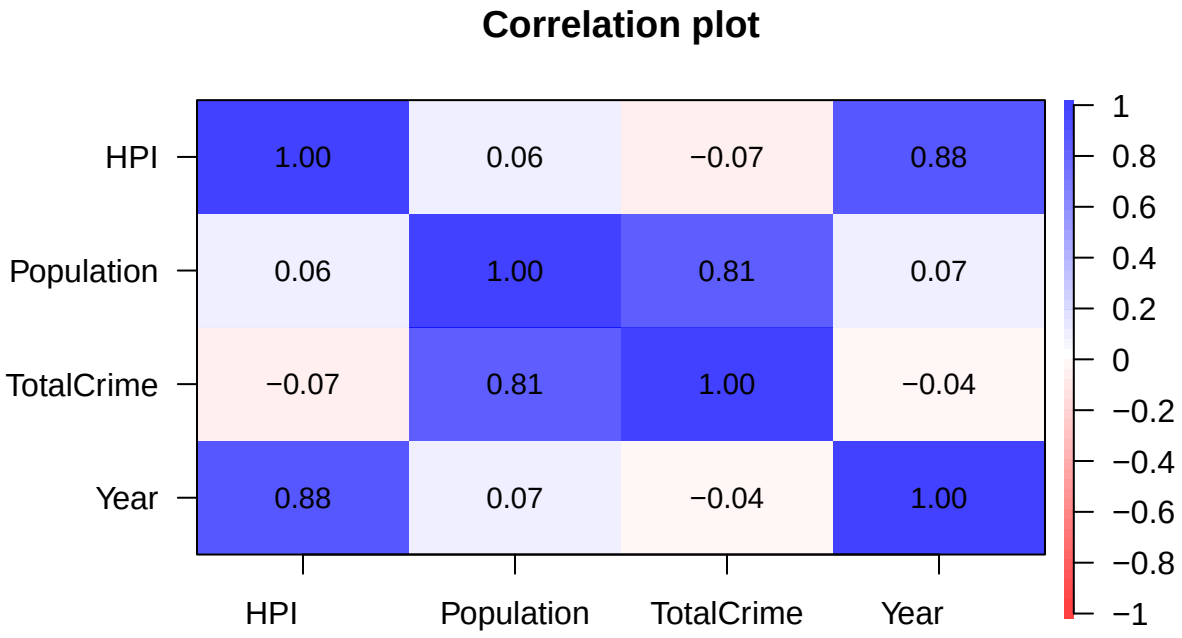
```
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
## Warning in par(usr): argument 1 does not name a graphical parameter
```



```r
# Method 2: Use a correlation plot

cor_data <- crimeData %>% select(HPI,Population, TotalCrime,Year)
cor_matrix <- cor(cor_data)
cor_matrix
```

```
##                    HPI Population  TotalCrime        Year
## HPI         1.00000000 0.05937148 -0.06697985  0.87549259
## Population  0.05937148 1.00000000  0.80871811  0.07322381
## TotalCrime -0.06697985 0.80871811  1.00000000 -0.04018779
## Year        0.87549259 0.07322381 -0.04018779  1.00000000
```

```r
corPlot(cor_matrix,cex =0.6)
```



Observations:

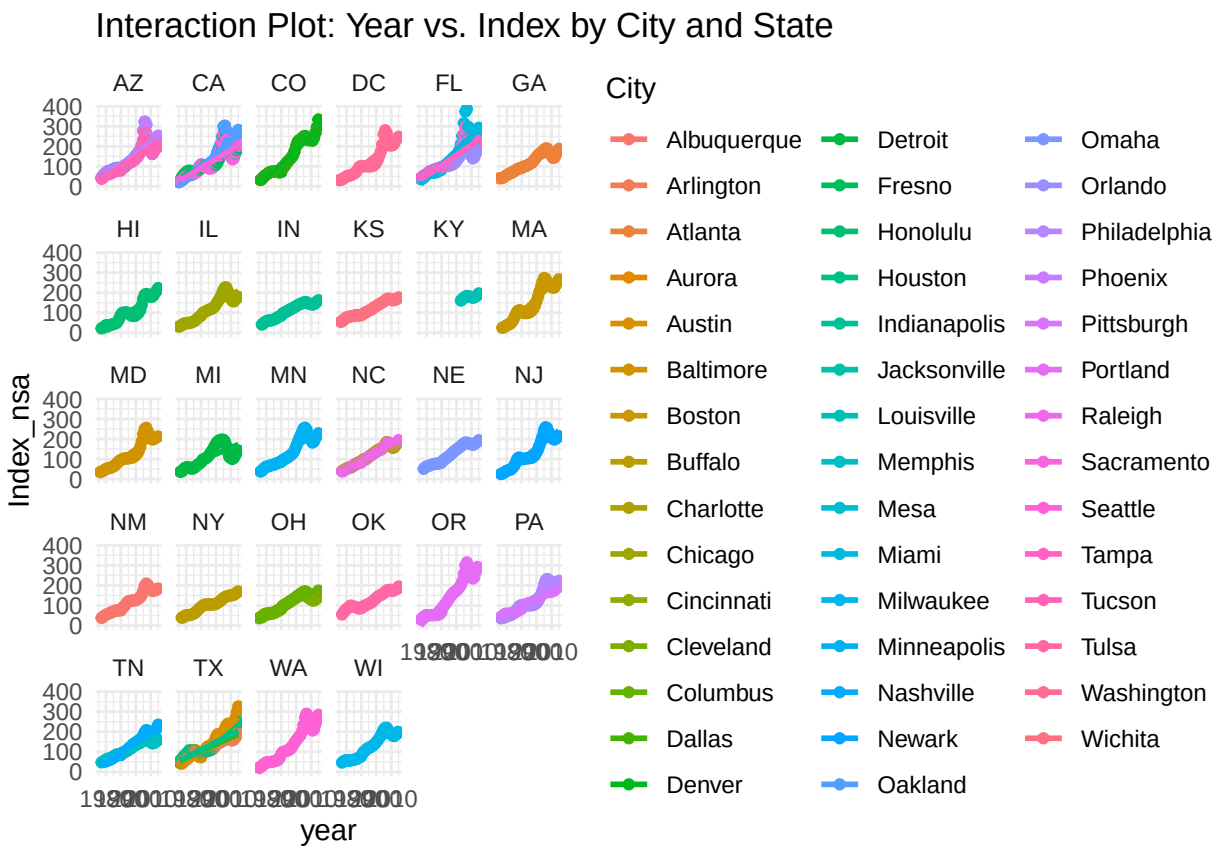
From the corrgram, we see that each individual crime (Rape, Robberies, Assaults, and Homicides) are all heavily correlated to one another. They are also all heavily correlated with total crime, which is intuitive as they directly impact the total number of observed crimes. Moreover, we see that each crime (and total crime) is correlated with the population size. Which makes sense as we expect to see a rise in crime with a bigger population. The final correlation observed is between year and HPI. HPI does vary year to year (as economic conditions vary year over year influencing housing prices), so this relationship is expected.

Our correlation plot further solidifies the results seen from the corrgram.

Interactions between a continuous and a categorical predictor

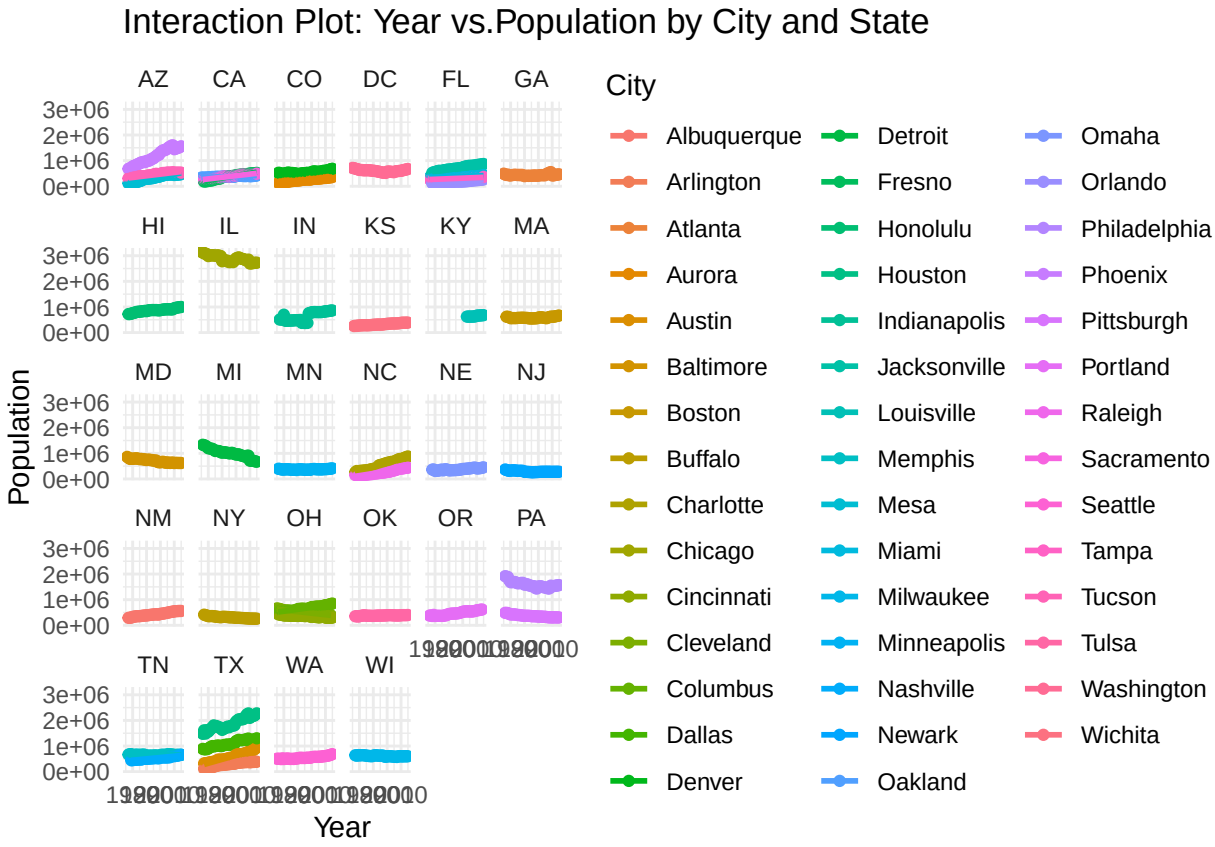
```
#interactions:
#categorical variables:city, state
#continuous variables: population, any crime, year, index_nsa

# Year vs Index_nsa by City and State
ggplot(crimeData, aes(x = Year, y = HPI, color = City))+geom_point()+geom_smooth(method="lm", se = FALSE
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



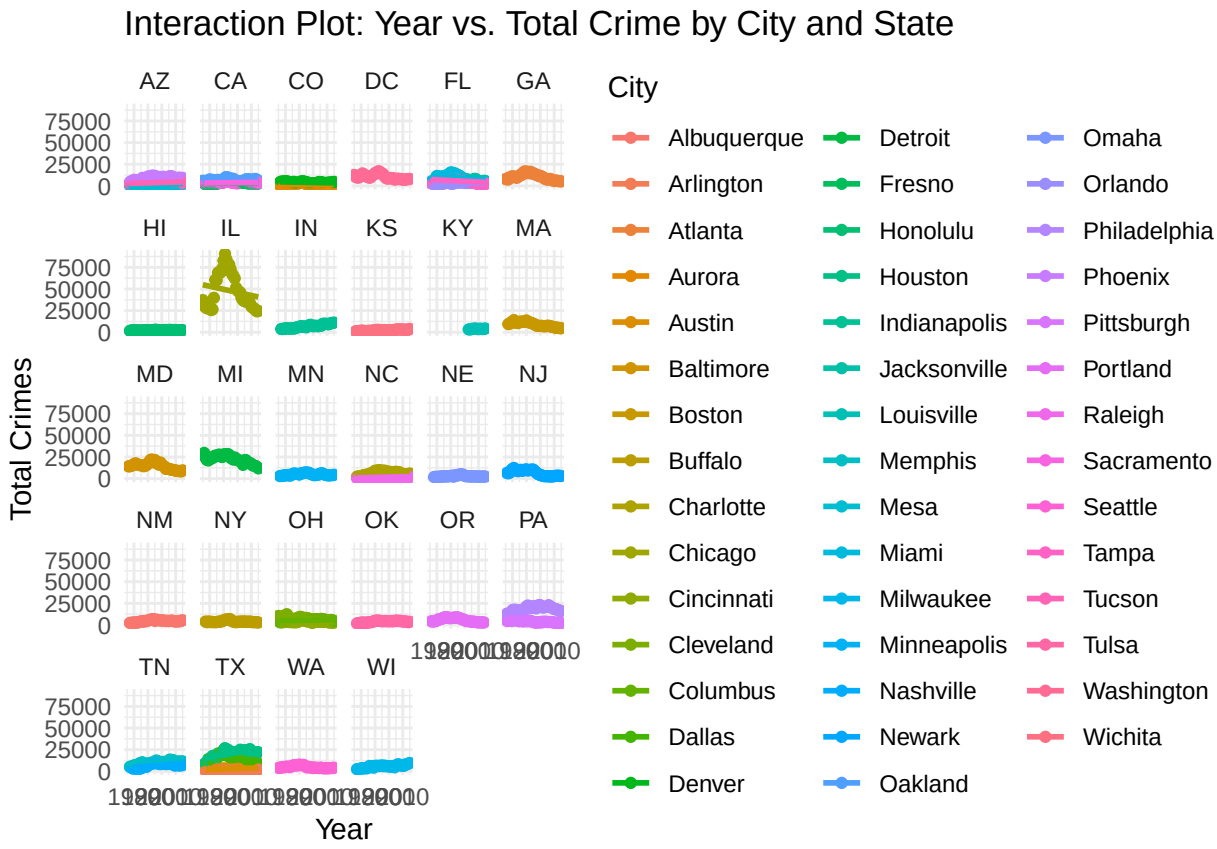

```
# Year vs Popluation by City and State
ggplot(crimeData, aes(x = Year, y = Population, color = City))+geom_point()+geom_smooth(method="lm", se
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



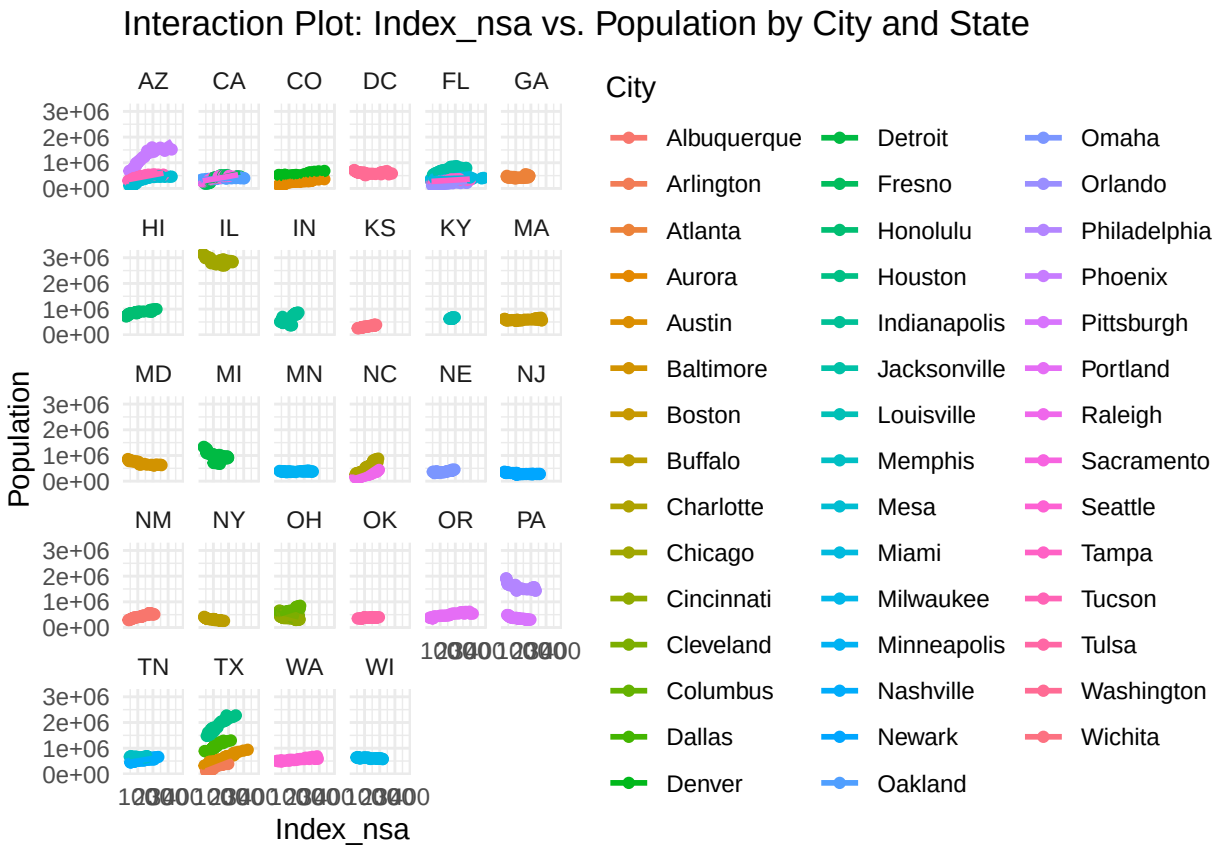

```
# Year vs Total Crimes by City and State
ggplot(crimeData, aes(x = Year, y = TotalCrime, color = City))+geom_point()+geom_smooth(method="lm", se
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



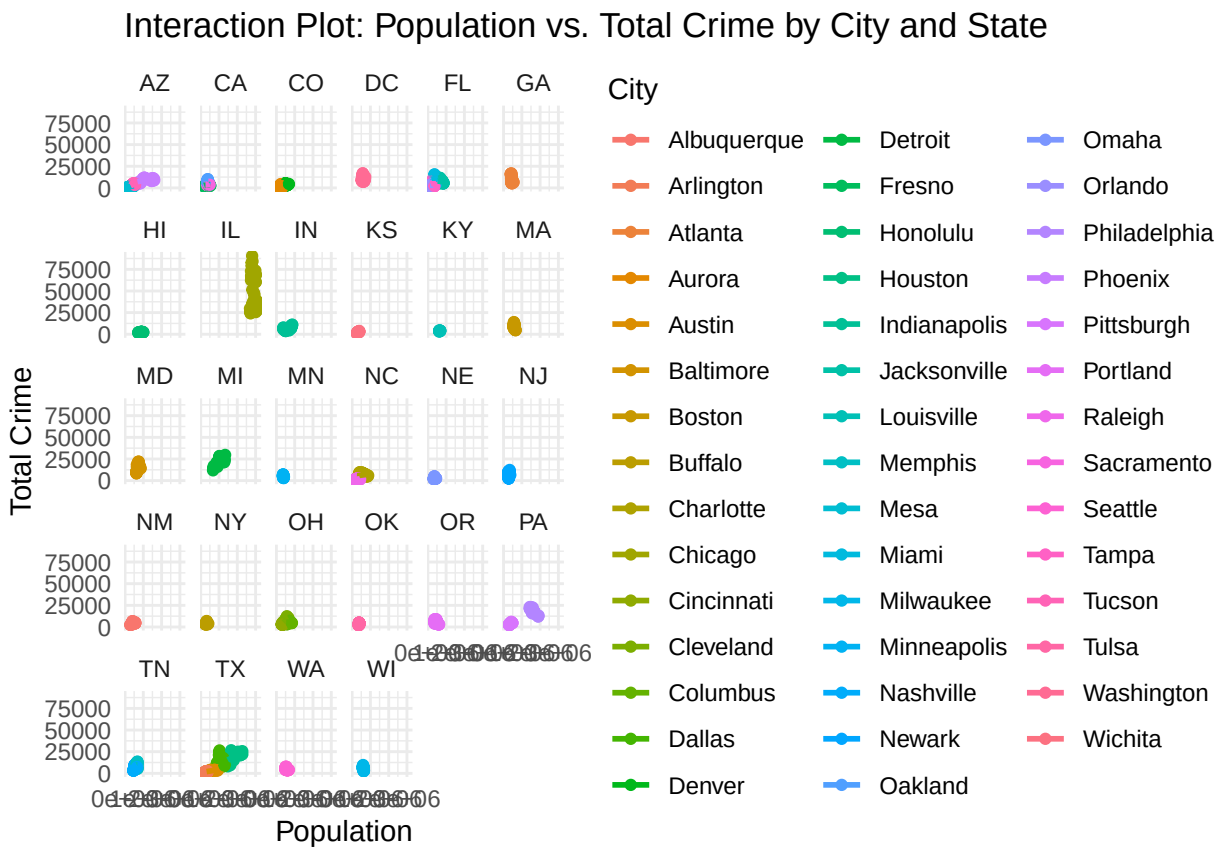

```
# Index_nsa vs Population by City and State
ggplot(crimeData, aes(x = HPI, y = Population, color = City))+geom_point()+geom_smooth(method="lm", se =
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# Population vs Total Crimes by City and State
ggplot(crimeData, aes(x = Population, y = TotalCrime, color = City))+geom_point()+geom_smooth(method="lm
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Observations:

Our data had two categorical variables, city and state. Thus, we created 5 interaction plots to measure the relationship of continuous variables within different states and the cities within those states. When comparing the HPI by year in each state and city, we see fairly similar results across the American states; although we can observe distinct variances between cities in the same state. When looking at population by year we see similar trends across the states, but there is clear variances between certain cities (even ones in the same state; but obviously some metropolis's will have larger populations). For total crime by year we once again can observe a specific trend across most states.States such as Illinois, Texas, Michigan and Pennsylvania have noticeably different trends (but this looks like its being driven from cities such as Detroit, Chicago, Houston and Philadelphia - cities more prolific for crime). When comparing HPI by population size its hard to observe any distinct trends in both American states and cities.For total crime vs population size, it appears that generally for most states a smaller population equates to a smaller number of total crimes, but within certain cities (namely, Chicago) this trend doesn't necessarily tie out.

Fitting a model by hand and performing an ESS test

```
# Fit model by hand

#We want to use city and state for our categorical variables, so lets make them into dummies
dummy_variable1=model.matrix(~City, data = crimeData)
dummy_variable1=dummy_variable1[,-1]
dummy_variable2=model.matrix(~State, data = crimeData)
dummy_variable2=dummy_variable2[,-1]
```

```r
# Potential Models

# This model has a lower R^2 value based off its ESS Test
y<-crimeData$HPI
A<-cbind(1,crimeData$Robberies,crimeData$Assaults,crimeData$Homicides,crimeData$Rapes,crimeData$Populat

B=solve(t(A) %*% A) %*% t(A) %*% y

SSE=t(y)%*%y-t(B)%*%t(A)%*%y
SSR=t(B)%*%t(A)%*%y-nrow(A)%*%(mean(y))^2
SST=t(y)%*%y-nrow(A)%*%(mean(y))^2
df_SSE=nrow(A)-nrow(B)
df_SSR=nrow(B)-1
df_SST=nrow(A)-1
r_squared=SSR/SST
r_squared
```

```
##            [,1]
## [1,] 0.3822636
```

```r
# This model had a very low r^2 value based off its ESS test
y1<-crimeData$HPI
A1<-cbind(1,crimeData$TotalCrime,crimeData$Population,dummy_variable2)

B1=solve(t(A1) %*% A1) %*% t(A1) %*% y1
SSE1=t(y)%*%y1-t(B1)%*%t(A1)%*%y
SSR1=t(B1)%*%t(A1)%*%y1-nrow(A1)%*%(mean(y1))^2
SST1=t(y1)%*%y1-nrow(A1)%*%(mean(y1))^2
df_SSE1=nrow(A1)-nrow(B1)
df_SSR1=nrow(B1)-1
df_SST1=nrow(A1)-1

r_squared1=SSR1/SST1
r_squared1
```

```
##          [,1]
## [1,] 0.10197
```

```r
# This model has a very high r^2 value, but not the highest!
y2<-crimeData$HPI
A2<-cbind(1,crimeData$TotalCrime,crimeData$Year,dummy_variable2)

B2=solve(t(A2) %*% A2) %*% t(A2) %*% y2
SSE2=t(y2)%*%y2-t(B2)%*%t(A2)%*%y2
SSR2=t(B2)%*%t(A2)%*%y2-nrow(A2)%*%(mean(y2))^2
SST2=t(y2)%*%y2-nrow(A2)%*%(mean(y2))^2
df_SSE2=nrow(A2)-nrow(B2)
df_SSR2=nrow(B2)-1
df_SST2=nrow(A2)-1

r_squared2=SSR2/SST2
r_squared2
```

```
##           [,1]
## [1,] 0.8030788
```

```
# This model has the highest r^2 value
y3<-crimeData$HPI
A3<-model.matrix(~ Year + TotalCrime + City + City:TotalCrime, data =crimeData)

B3=solve(t(A3) %*% A3) %*% t(A3) %*% y3
SSE3=t(y3)%*%y3-t(B3)%*%t(A3)%*%y3
SSR3=t(B3)%*%t(A3)%*%y3-nrow(A3)%*%(mean(y3))^2
SST3=t(y3)%*%y3-nrow(A3)%*%(mean(y3))^2
df_SSE3=nrow(A3)-nrow(B3)
df_SSR3=nrow(B3)-1
df_SST3=nrow(A3)-1

r_squared3=SSR3/SST3
r_squared3
```

```
##           [,1]
## [1,] 0.8471171
```

```
# Encountered errors when modelling by hand

# Adding the following interaction terms resulted in singularity issues
# 1. Year:Population Interaction, #2. Year:TotalCrime Interaction, #3. Population:TotalCrime Interactio

y3<-crimeData$HPI
A3<-model.matrix(~ Population + Year + Population:Year, data =crimeData)

#B3=solve(t(A3) %*% A3) %*% t(A3) %*% y3

# Cannot include city and state, too much aliased variables
# For example, the model below couldn't do a VIF test since is has several aliased variables
A1<-cbind(1,crimeData$TotalCrime,crimeData$Population,dummy_variable1,dummy_variable2)
aliased_variables1=alias(lm(y~A1-1))

# Cannot include Year and Pop.
# Take the model below for an example, this model suggested serious multicollinearity
vif_data=data.frame(HPI = y,TotalCrime=crimeData$TotalCrime, Population = crimeData$Population, Year=cr
fit=lm(HPI~.,data=vif_data)
vif_results=vif(fit)
vif_results
```

```
##       TotalCrime       Population             Year     CityArlington
##         5.468003        29.716802         1.224754         2.014092
##      CityAtlanta       CityAurora       CityAustin    CityBaltimore
##         2.059382         2.068572         2.017179         2.307912
##       CityBoston      CityBuffalo    CityCharlotte      CityChicago
##         2.039088         1.990365         1.988844        20.041896
##   CityCincinnati    CityCleveland     CityColumbus       CityDallas
##         1.945501         2.006931         2.127602         3.227099
##       CityDenver      CityDetroit       CityFresno     CityHonolulu
##         2.014883         3.188749         1.963211         2.488579
```

```
##      CityHouston CityIndianapolis CityJacksonville   CityLouisville
##         7.598519         2.057710         2.095475         1.359315
##      CityMemphis         CityMesa        CityMiami    CityMilwaukee
##         2.107248         1.989270         2.018913         2.041355
##  CityMinneapolis    CityNashville       CityNewark      CityOakland
##         1.986392         1.938008         2.040142         2.019660
##        CityOmaha      CityOrlando CityPhiladelphia      CityPhoenix
##         1.940888         2.085025         5.687140         3.358429
##   CityPittsburgh     CityPortland      CityRaleigh   CitySacramento
##         1.988296         1.961513         2.032359         1.984229
##      CitySeattle        CityTampa       CityTucson        CityTulsa
##         2.044188         1.999302         1.956085         1.960996
##   CityWashington      CityWichita
##         2.132644         2.011910
```

```
# Cannot include all 4 individual crimes with total crimes too much correlation
```

Filler space of observations

When trying to build various models by hand, we ran into several errors and limitations. Firstly, including both categorical variables (state and city) within the same linear model resulted in too many aliased variables. Moreover, including Year and Population within the same linear model created problems as well. For example, when we performed a VIF test for a model that included both of these categories (as well as total crime and city), we got a VIF of ~29 for population, ~20 for Chicago:Population and ~5 for total crime, suggesting serious multicollinearity. We also couldn't create a model that included all 4 individual crimes with total crime (which made sense do to the observed correlation from the corrgram). Finally the addition of the following interaction terms resulted in singularity issues when solving the model by hand: Year:Population, Year:TotalCrime, Population:TotalCrime, Population:State.

For the models we were able to test (Robberies+Assaults+Homicides+Rapes+Population+City, TotalCrime+Population+State, TotalCrime+Year+State, Year+TotalCrime+City+City:TotalCrime to name a few), we saw observed some promising models. Models that included Year yielded high R^2 values, whereas models with population resulted in some of the lowest observed r^2 values. This is likely due to the correlation Year had with HPI. It also seemed that the inclusion of city resulted in better models than the ones with State.

From the information above we can determine what combination of predictors will create the "best" linear model. Firstly, since we cannot include both categorical variables (due to the observed errors), we should choose city over state as it appeared to yield better results. Additionally, we cannot use both population and year together due to the issues we previously endured and thus we will choose year over population based on our findings above. I think it's also trivial to choose Total Crime over each individual crime as it encompasses the information as a whole (listing each individual crime didn't improve our findings). The individual crimes are act more as proxies for total crime. Finally, we will include interaction terms in our final model where possible to improve the accuracy of model (Especially between City and Total Crime based off the interactions we identified earlier from our plots).

Investigate transformations of the Predictors and of the response

```
# Lets choose the lm from above with the greatest r^2
# Lets Use a stepwise AIC approach to see if there is any modifications we should make to the model
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
lm1<-lm(HPI ~ Year + TotalCrime + City + City:TotalCrime, data =crimeData)
summary1<-summary(lm1)
summary1$adj.r.squared
```

```
## [1] 0.8388072
```

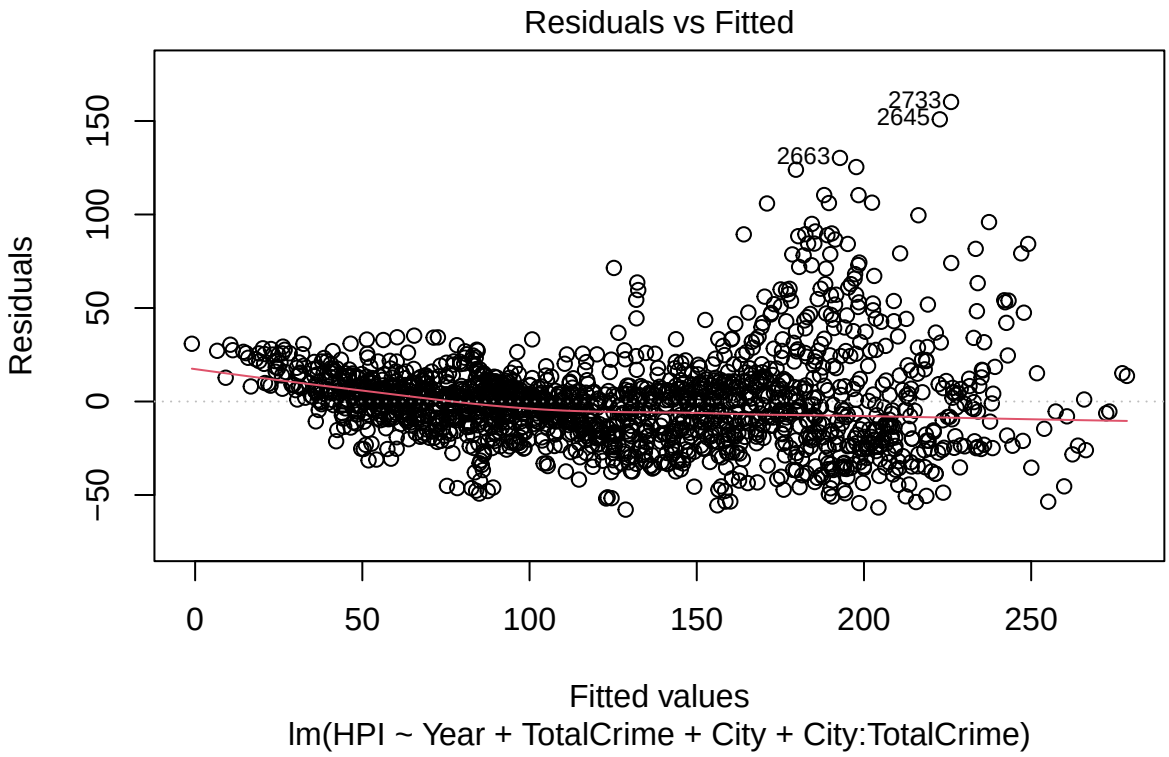
```
stepwise_approach<-stepAIC(lm1, direction="both")
```

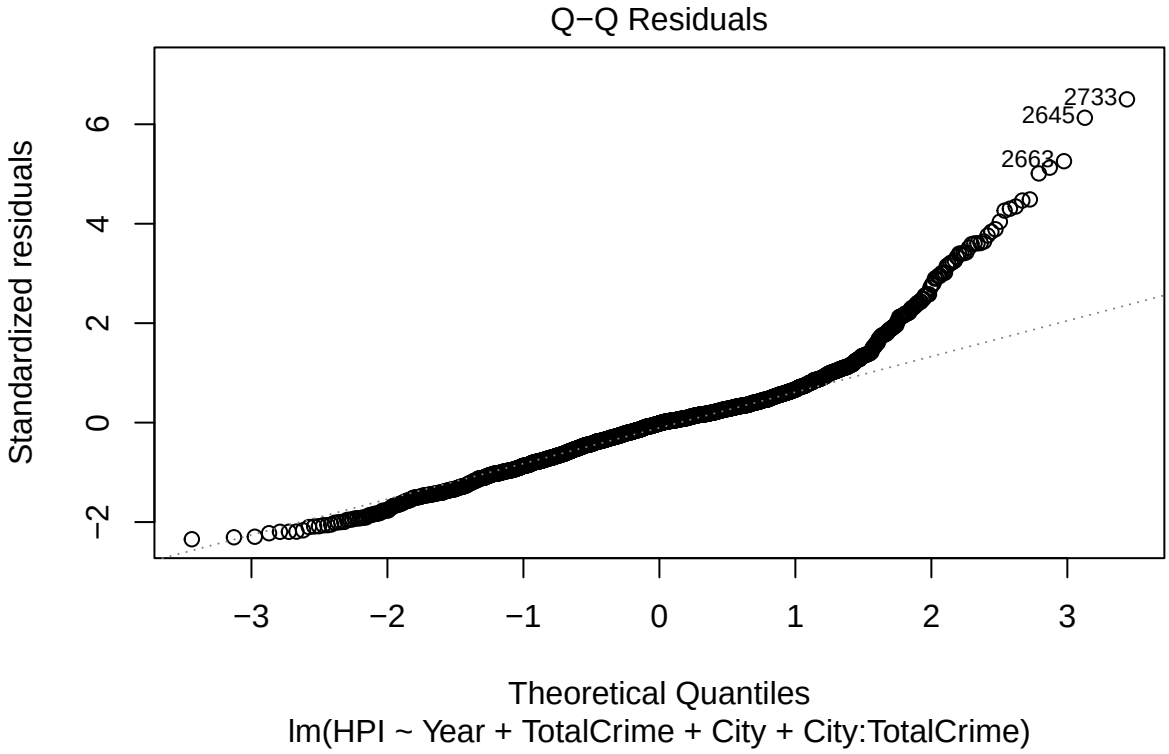
```
## Start:  AIC=11105.34
## HPI ~ Year + TotalCrime + City + City:TotalCrime
##
##                    Df Sum of Sq     RSS   AIC
## <none>                           1025630 11105
## - TotalCrime:City 43    238957 1264587 11377
## - Year             1   3457409 4483039 13623
```

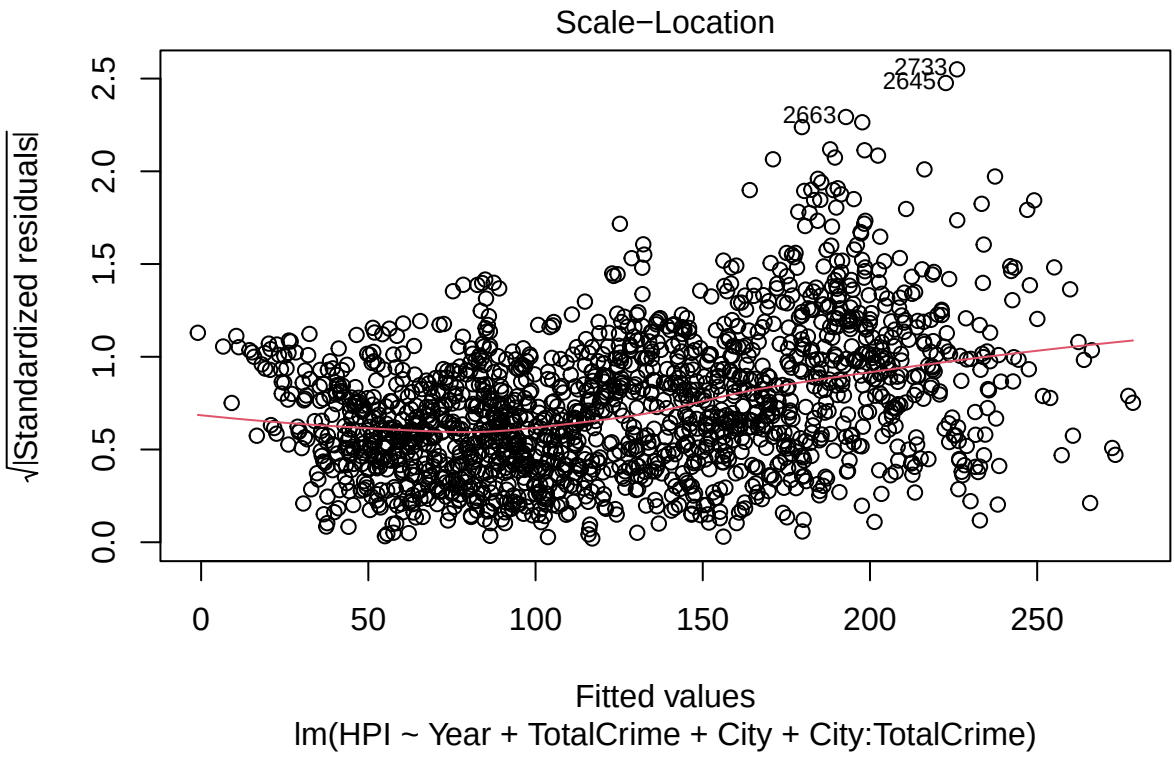
```
summary<-summary(stepwise_approach)
summary$adj.r.squared
```

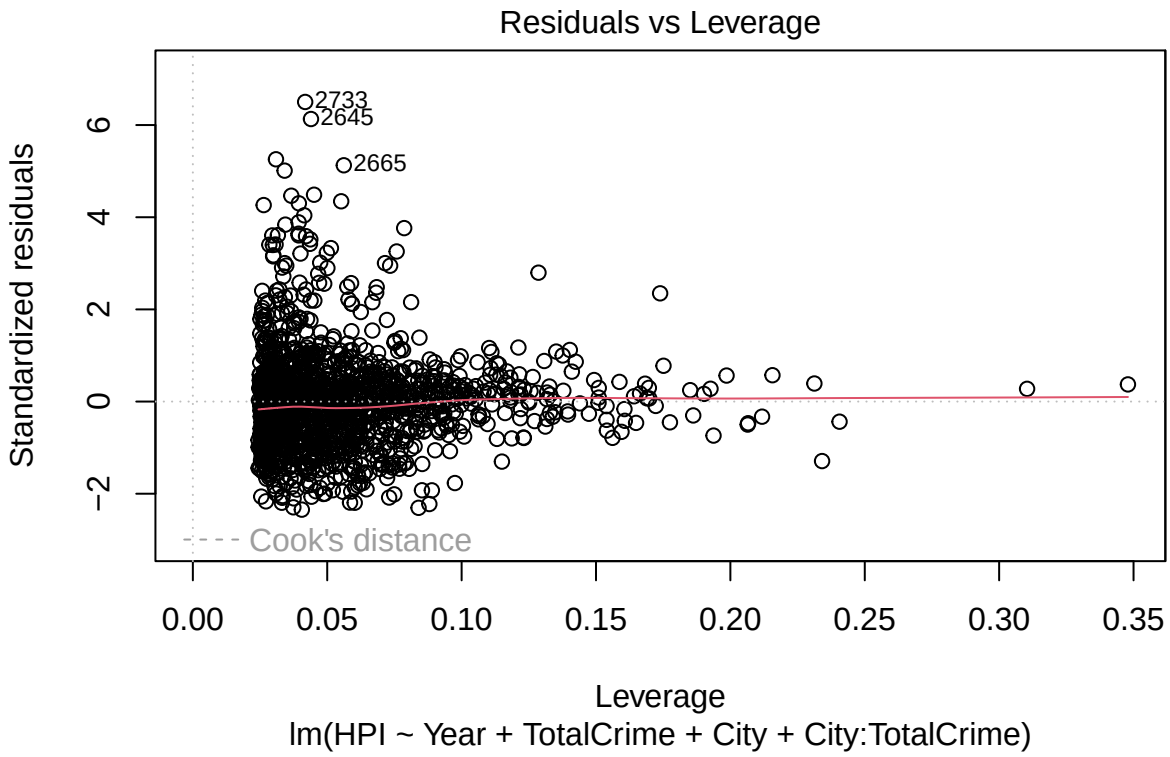
```
## [1] 0.8388072
```

```
# The Multiple R^2 value was the same as the one we calculated (0.8471), the adjusted was the same too

# Check diagnostic plots to see what transformations to apply
plot(lm1)
```


Residuals vs Fitted
Residuals
Fitted values
lm(HPI ~ Year + TotalCrime + City + City:TotalCrime)
2733
2645
2663
150
100
50
0
−50
0
50
100
150
200
250


Q−Q Residuals
Standardized residuals
−2
0
2
4
6
2733
2645
2663
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(HPI ~ Year + TotalCrime + City + City:TotalCrime)

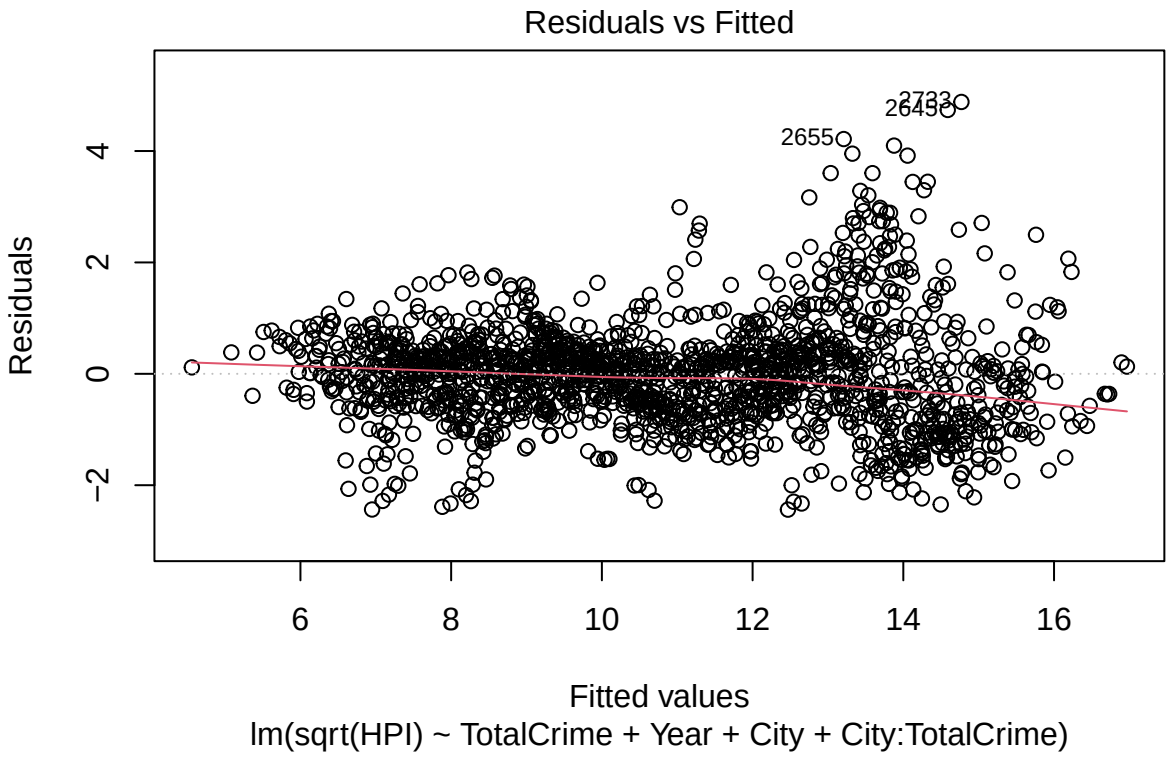

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
2733
2645
2663
0
50
100
150
200
250
Fitted values
lm(HPI ~ Year + TotalCrime + City + City:TotalCrime)

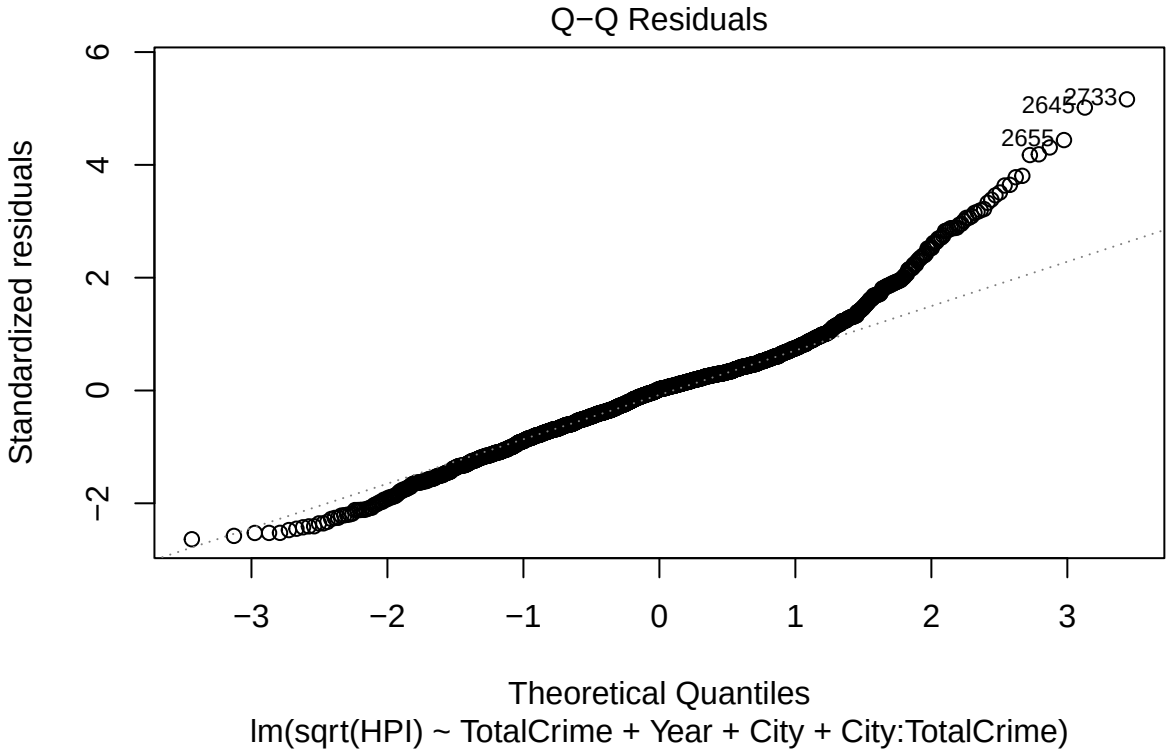


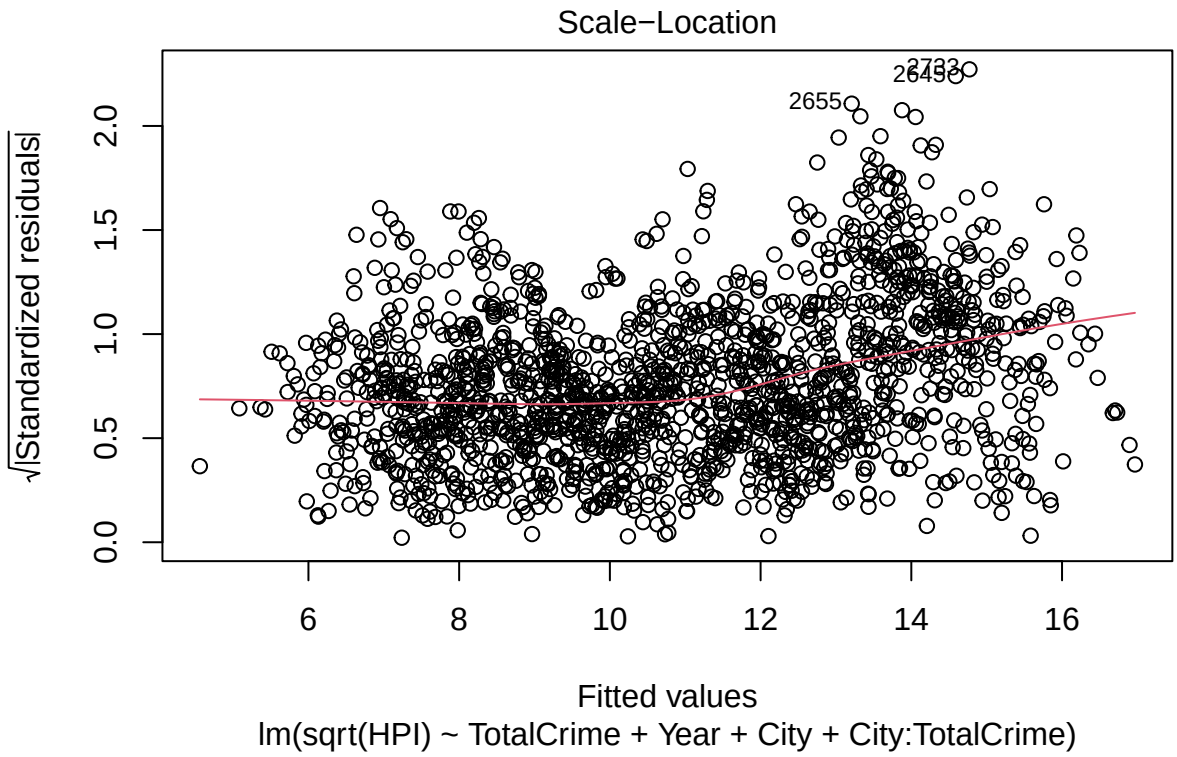

```
# Determined from the QQ-plot the residuals are not normal as in the tails we see they trail off the 45
# From the residuals vs leverage we do see some influential points (this could be Chicago as it has cau
# From earlier scatter plots, we tested against HPI vs Total Crime and HPI vs year we see a non-linear
# Scale location is a curved line oppose to a straight horizontal one, could attempt a log transformati

# Lets start by trying a basic sqrt transformation
lm2<-lm(sqrt(HPI) ~ TotalCrime + Year + City + City:TotalCrime, data =crimeData)
summary2<-summary(lm2)
summary2$adj.r.squared
```

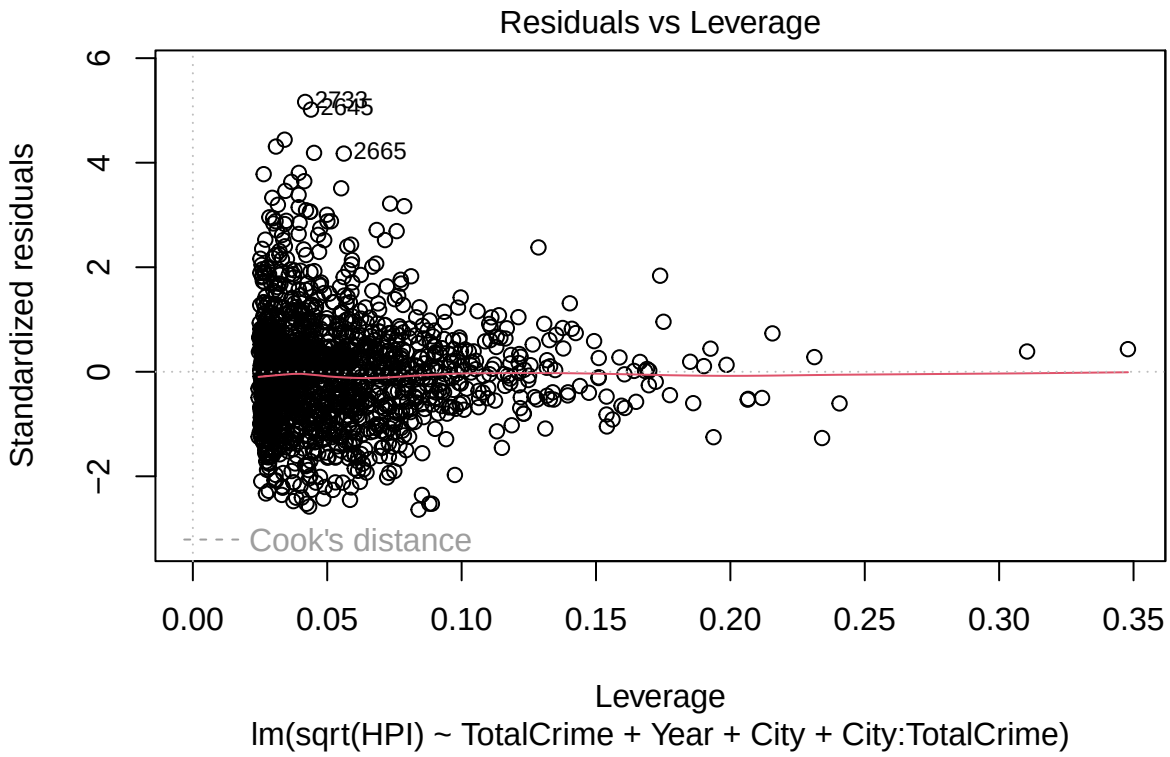
```
## [1] 0.8789059
```

```
plot(lm2)
```


Residuals vs Fitted
2732
2645
2655
Residuals
4
2
0
−2
6
8
10
12
14
16
Fitted values
lm(sqrt(HPI) ~ TotalCrime + Year + City + City:TotalCrime)


Q−Q Residuals
Standardized residuals
6
4
2
0
−2
2733
2645
2655
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(sqrt(HPI) ~ TotalCrime + Year + City + City:TotalCrime)

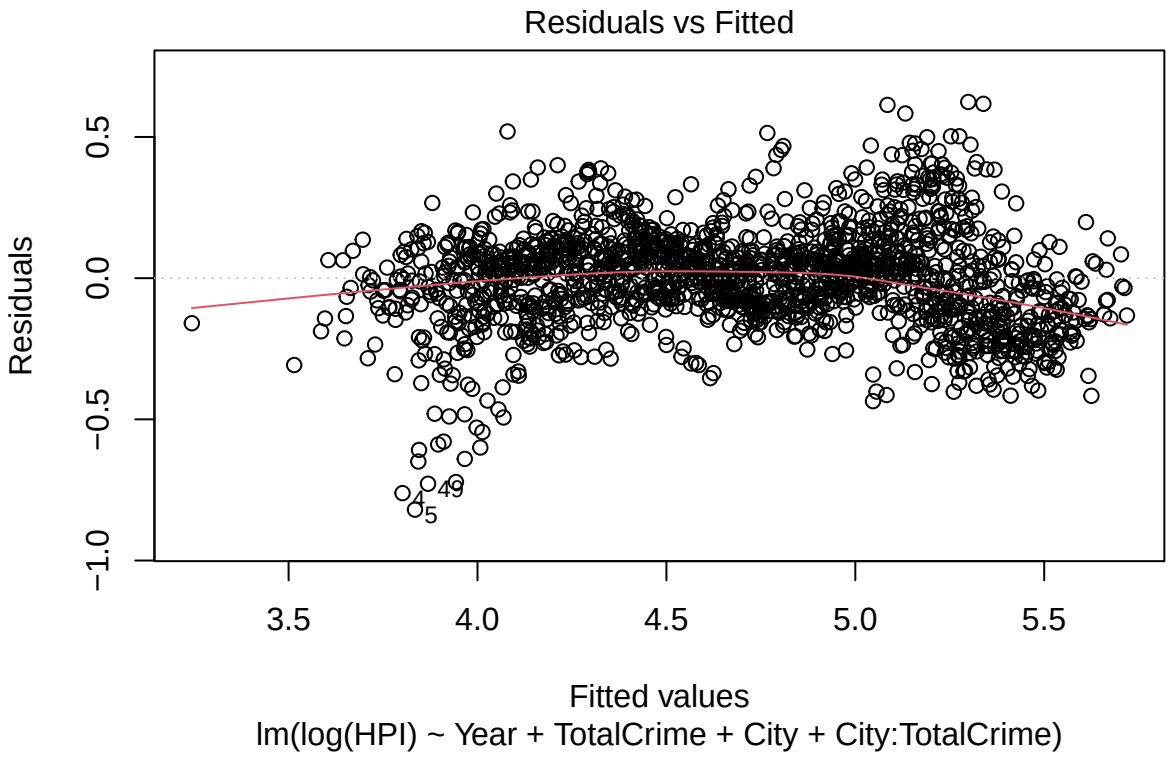

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2733
2645
2655
6
8
10
12
14
16
Fitted values
lm(sqrt(HPI) ~ TotalCrime + Year + City + City:TotalCrime)

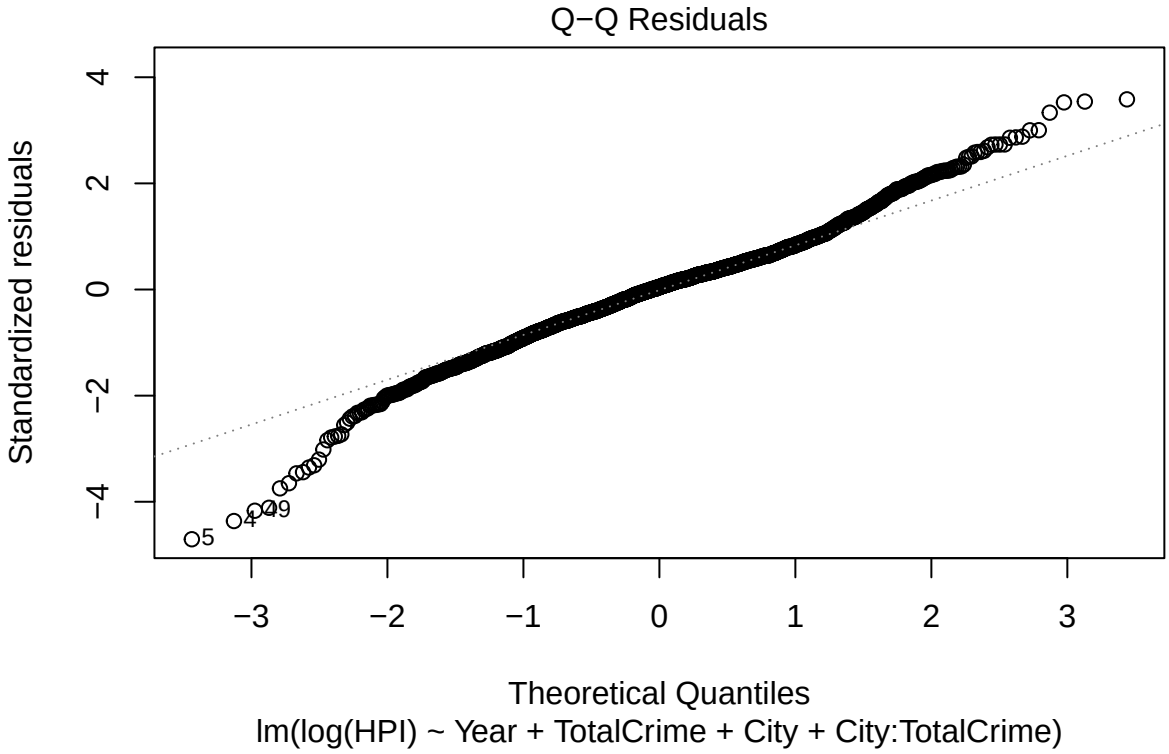


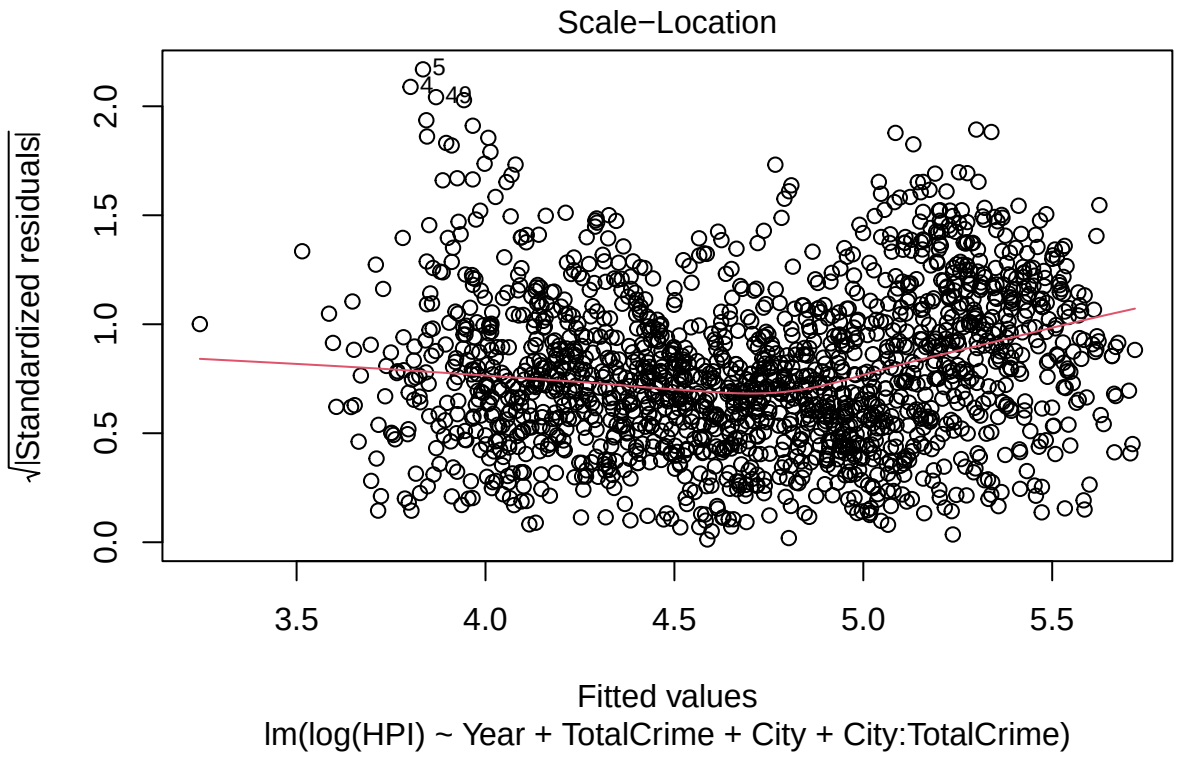

```
#We see improvements in both the diagnostic plots and the adj. r^2 value (0.8789059). But based of the
lm3<-lm(log(HPI) ~ Year + TotalCrime + City + City:TotalCrime, data =crimeData)
summary3<-summary(lm3)
summary3$adj.r.squared
```

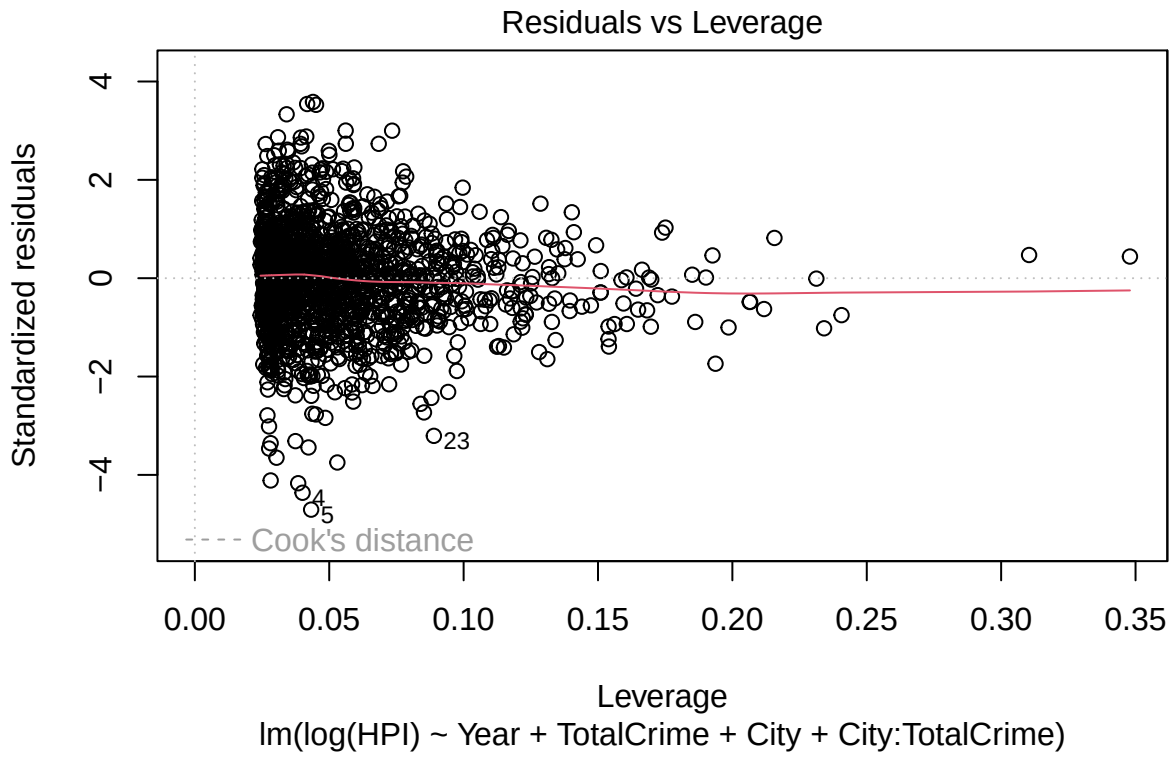
```
## [1] 0.8867949
```

```
plot(lm3)
```


Residuals vs Fitted
Residuals
0.5
0.0
−0.5
−1.0
3.5
4.0
4.5
5.0
5.5
Fitted values
lm(log(HPI) ~ Year + TotalCrime + City + City:TotalCrime)


Q−Q Residuals
Standardized residuals
4
2
0
−2
−4
−3
−2
−1
0
1
2
3
5
4
49
Theoretical Quantiles
lm(log(HPI) ~ Year + TotalCrime + City + City:TotalCrime)

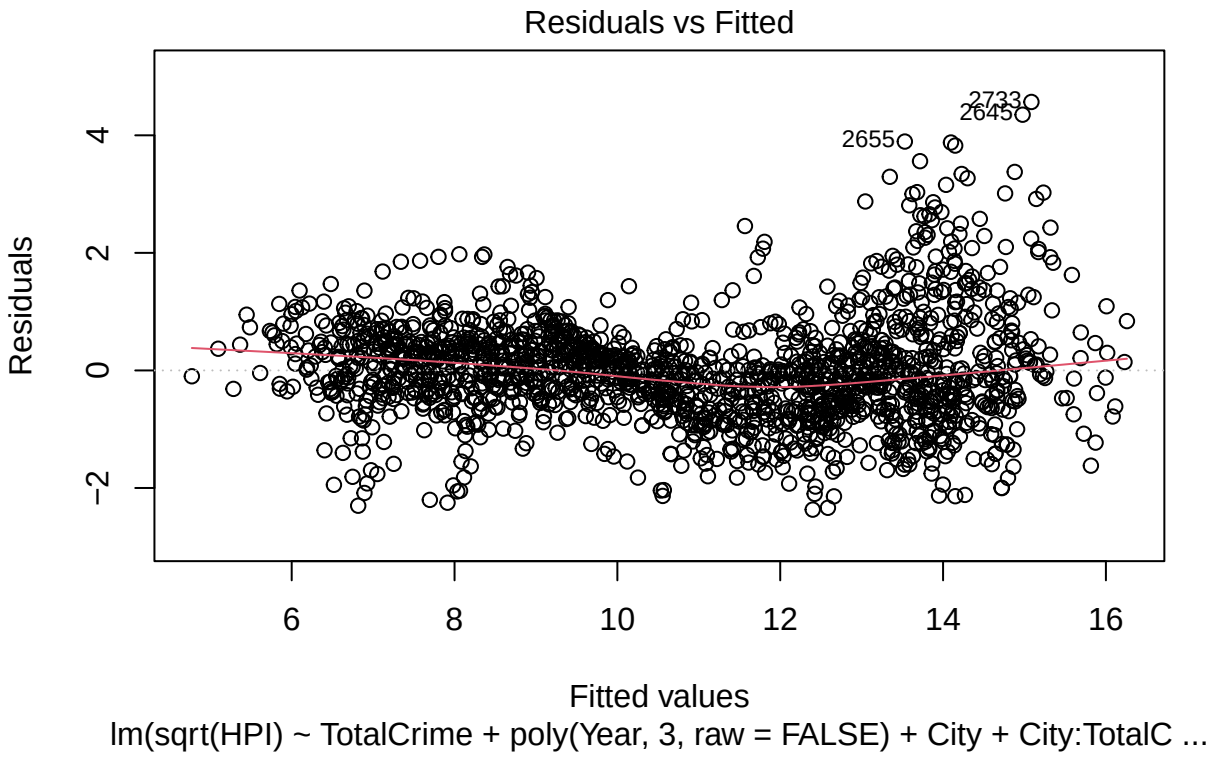

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
3.5
4.0
4.5
5.0
5.5
Fitted values
lm(log(HPI) ~ Year + TotalCrime + City + City:TotalCrime)

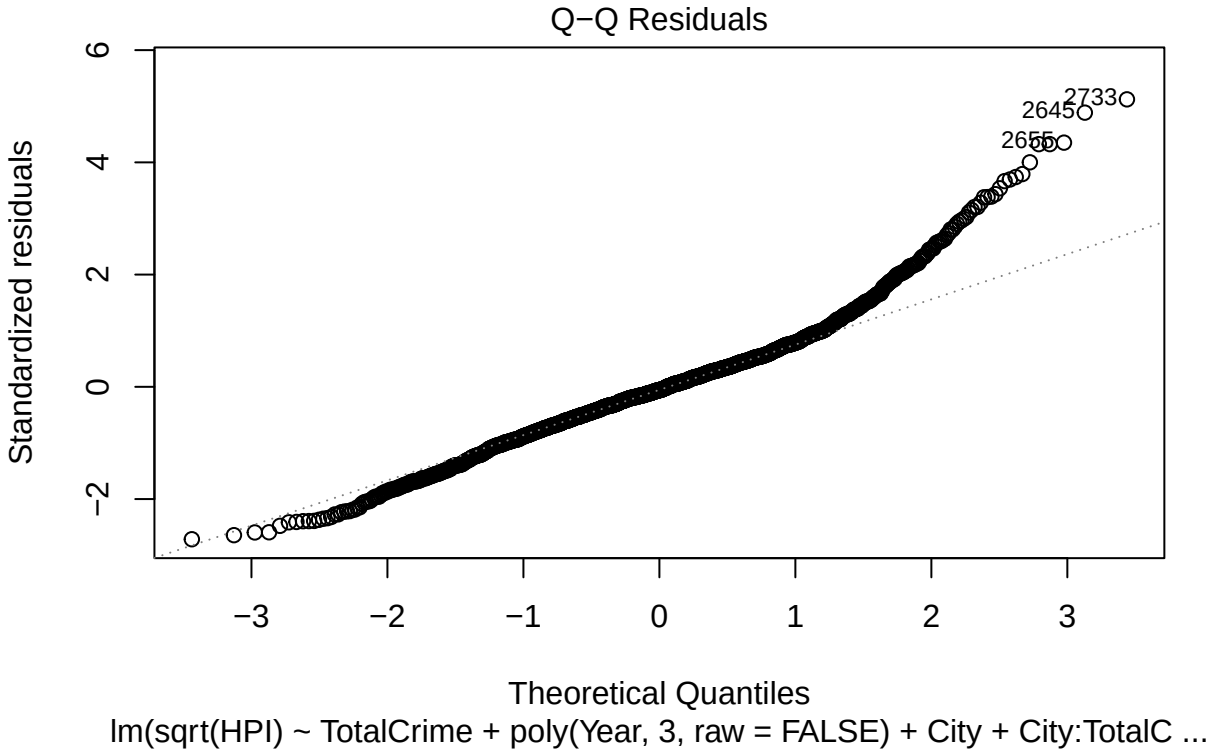


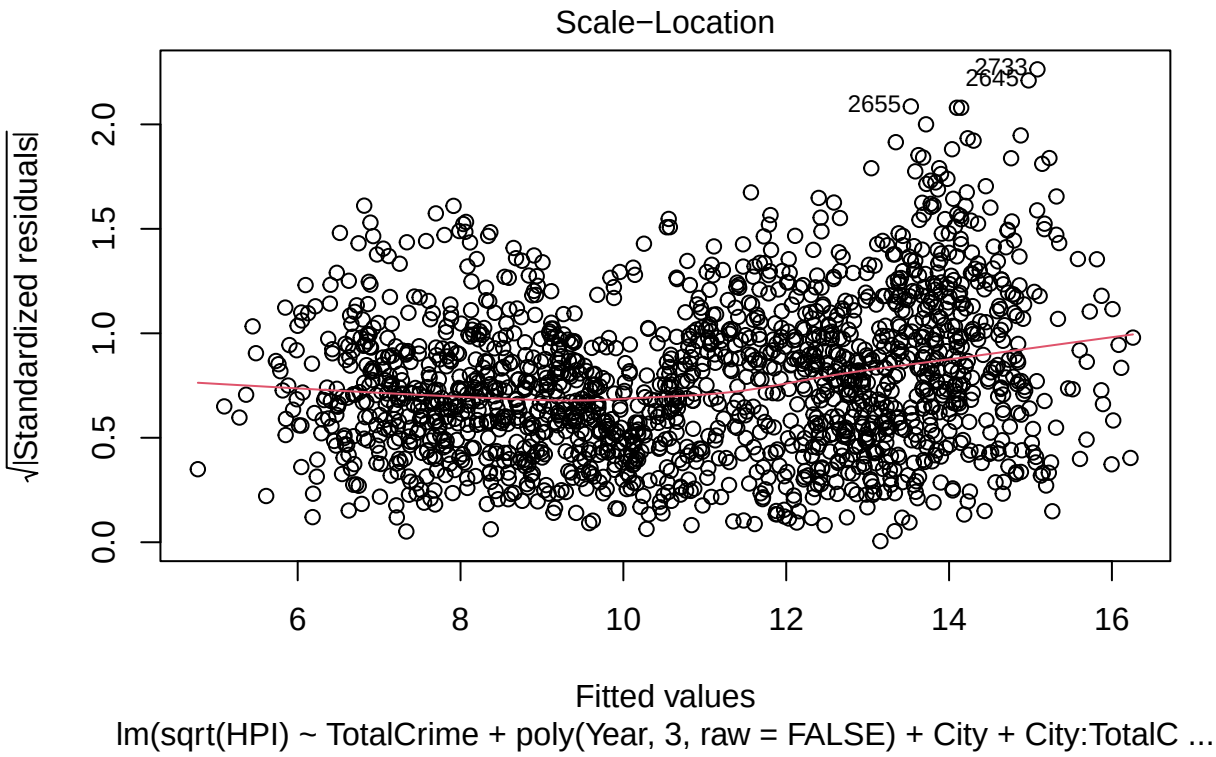

```
#From both the diagnostic plots and the adj. r^2 value (0.8867949), we see some improvements so lets ex
lm4<-lm(sqrt(HPI) ~ TotalCrime + poly(Year, 3, raw = FALSE) + City + City:TotalCrime, data =crimeData)
summary4<-summary(lm4)
summary4$adj.r.squared
```

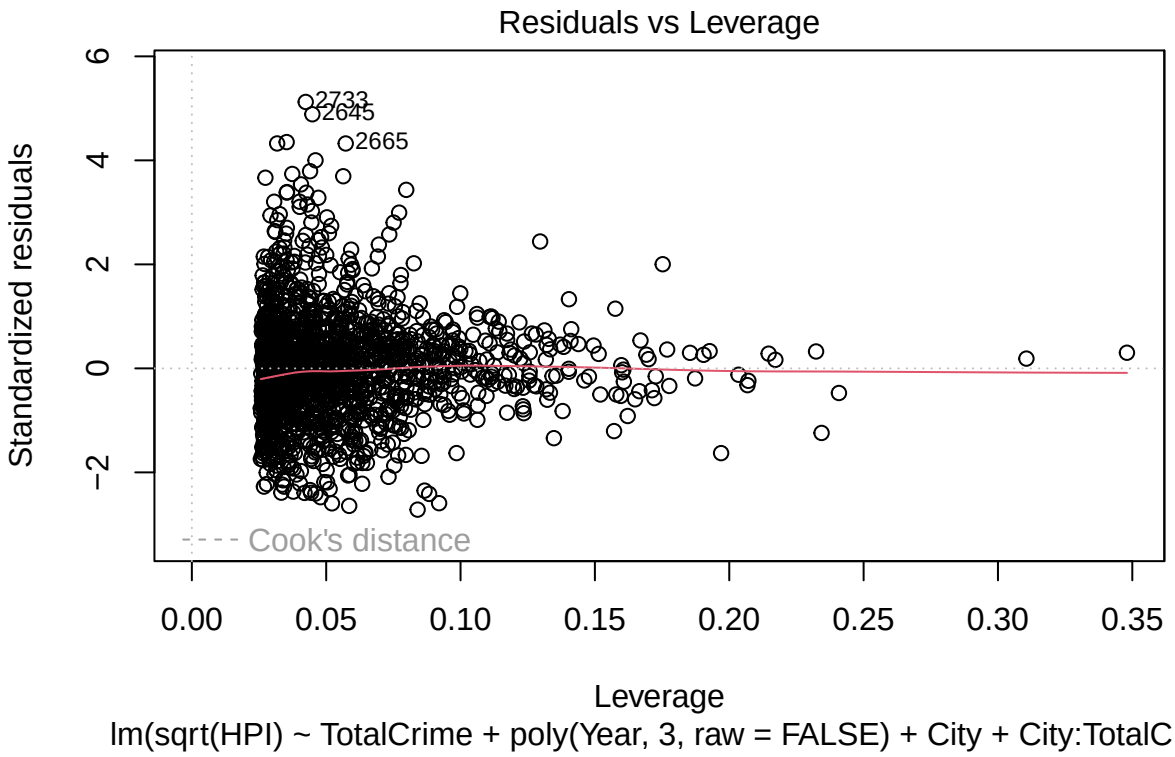
```
## [1] 0.8922871
```

```
plot(lm4)
```

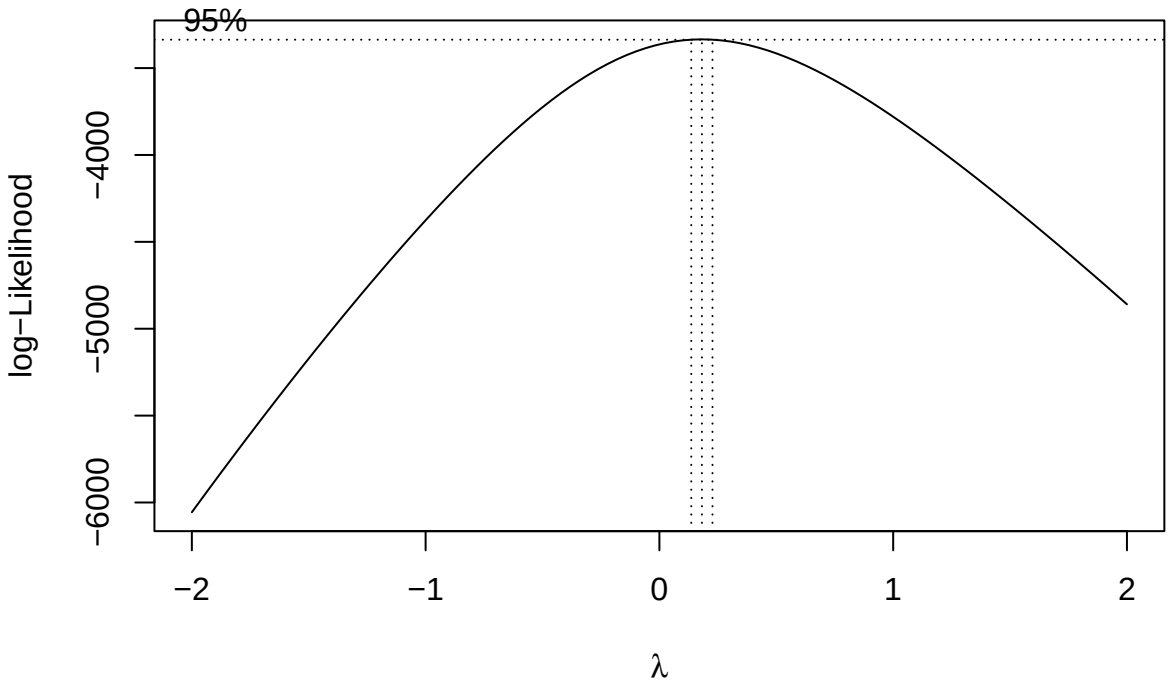

Residuals vs Fitted
Residuals
Fitted values
lm(sqrt(HPI) ~ TotalCrime + poly(Year, 3, raw = FALSE) + City + City:TotalC ...
2733
2645
2655
−2
0
2
4
6
8
10
12
14
16


Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(sqrt(HPI) ~ TotalCrime + poly(Year, 3, raw = FALSE) + City + City:TotalC ...
2733
2645
2655
6
4
2
0
−2
−3
−2
−1
0
1
2
3


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2733
2645
2655
6
8
10
12
14
16
Fitted values
lm(sqrt(HPI) ~ TotalCrime + poly(Year, 3, raw = FALSE) + City + City:TotalC ...




```
# Once again we saw an improvement (the adj. r^2 value is now 0.9167559). Lets finally apply a box-cot
BC=boxcox(lm1)
```




```
L=BC$x[which.max(BC$y)]
crimeData$HPI_BCM=((crimeData$HPI)^L-1)/L
lm5_BCM=lm(HPI_BCM~Year + TotalCrime + City + City:TotalCrime, data = crimeData)
summary5<-summary(lm5_BCM)
summary5$adj.r.squared
```

```
## [1] 0.8882729
```

```
# Actually did not result in the highest r^2 value seen thus far (0.8882729)

#Thus lm4 is our best model, we can further verify this by comparing its AIC versus the AIC for the ori
library(MASS)
stepwise_approach1<-stepAIC(lm4, direction="both")
```

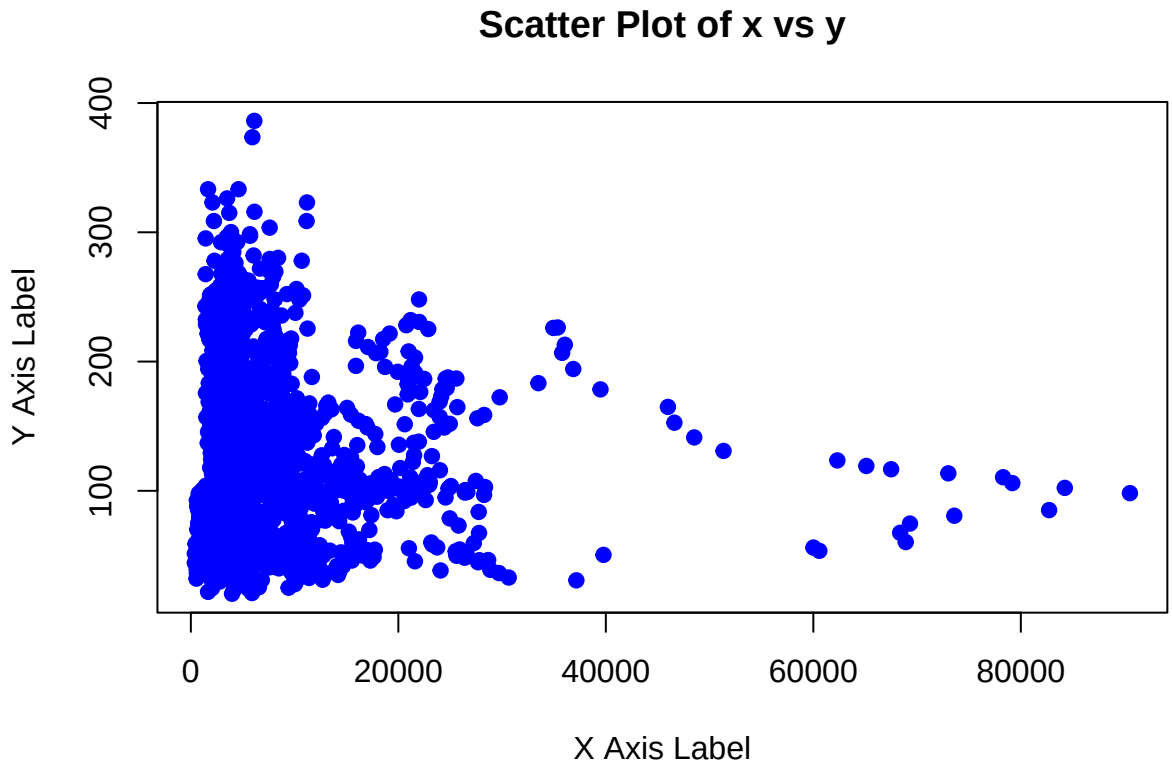
```
## Start:  AIC=-229.78
## sqrt(HPI) ~ TotalCrime + poly(Year, 3, raw = FALSE) + City +
##     City:TotalCrime
##
##                              Df Sum of Sq    RSS     AIC
## <none>                                    1342.1 -229.78
## - TotalCrime:City            43     334.0 1676.1   63.77
## - poly(Year, 3, raw = FALSE)  3    7491.8 8833.9 2982.71
```

```r
summary1<-summary(stepwise_approach1)
summary1$adj.r.squared
```

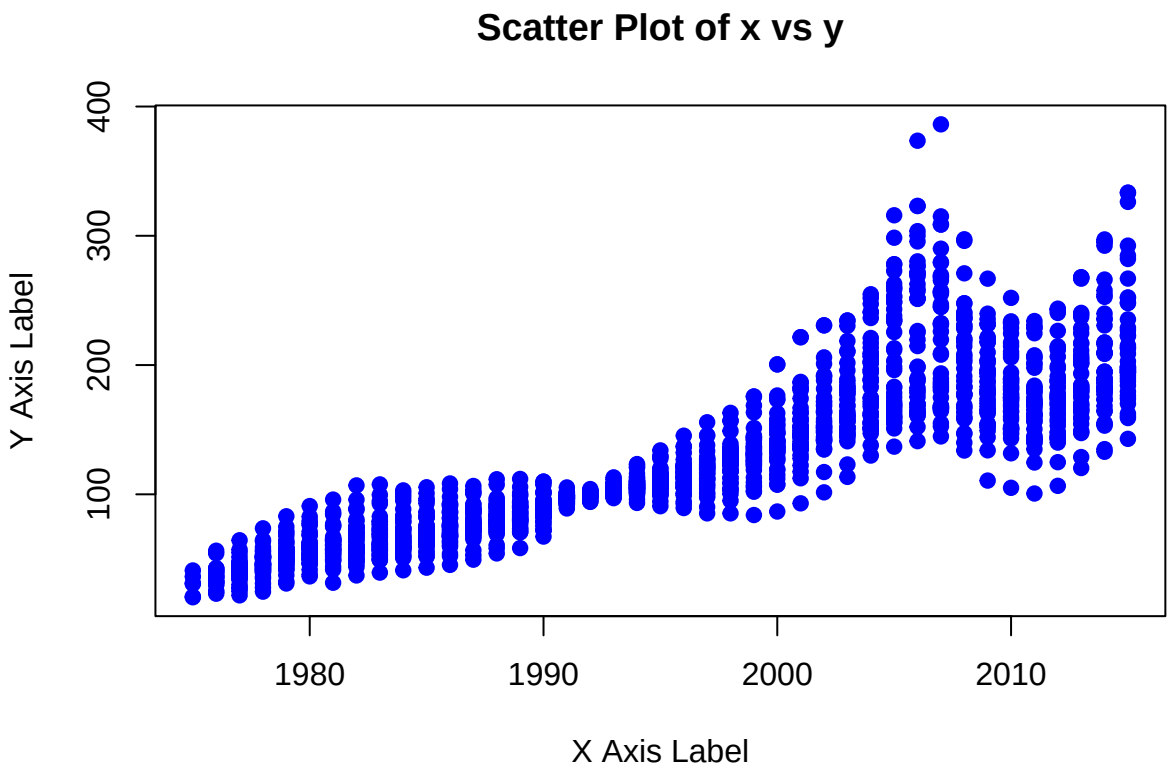
```
## [1] 0.8922871
```

```r
# This model has a significantly lower AIC indicating its a better fit. Also both models determined tha
```

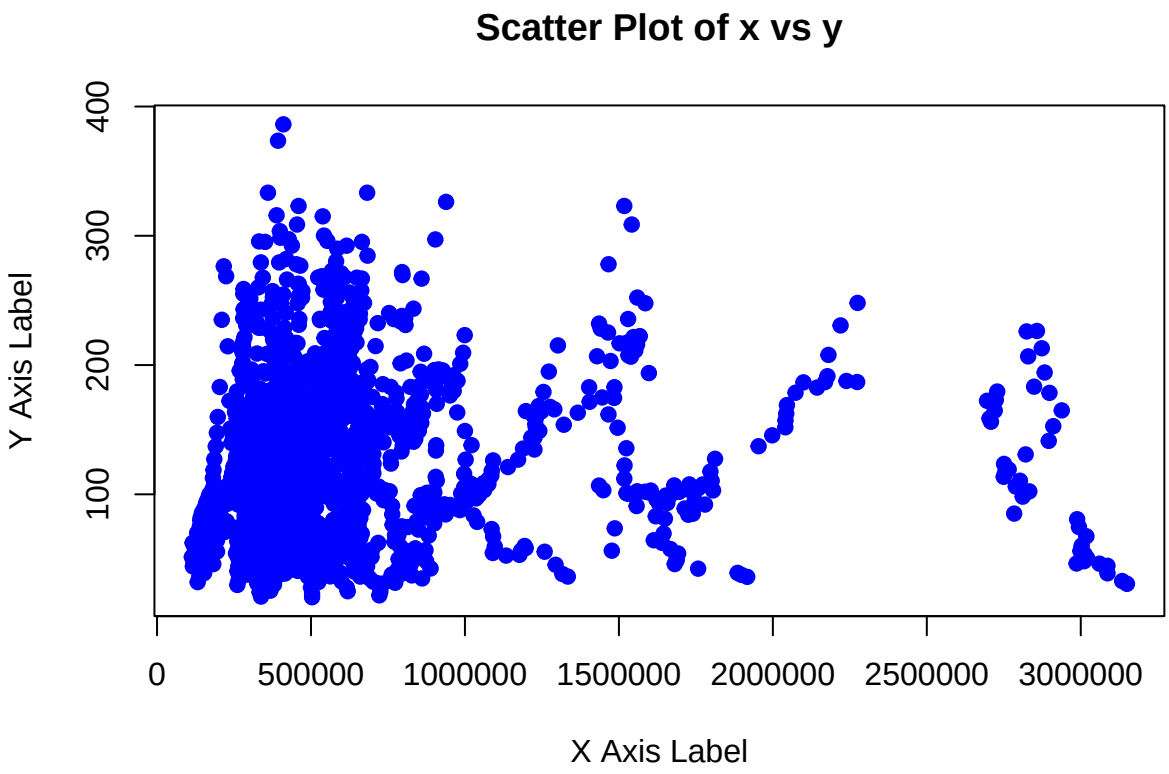
```r
plot(crimeData$TotalCrime, crimeData$HPI,
     main = "Scatter Plot of x vs y",
     xlab = "X Axis Label",
     ylab = "Y Axis Label",
     pch = 19,          # Type of point (19 is filled circle)
     col = "blue")
```




```r
plot(crimeData$Year, crimeData$HPI,
     main = "Scatter Plot of x vs y",
     xlab = "X Axis Label",
     ylab = "Y Axis Label",
     pch = 19,          # Type of point (19 is filled circle)
     col = "blue")
```



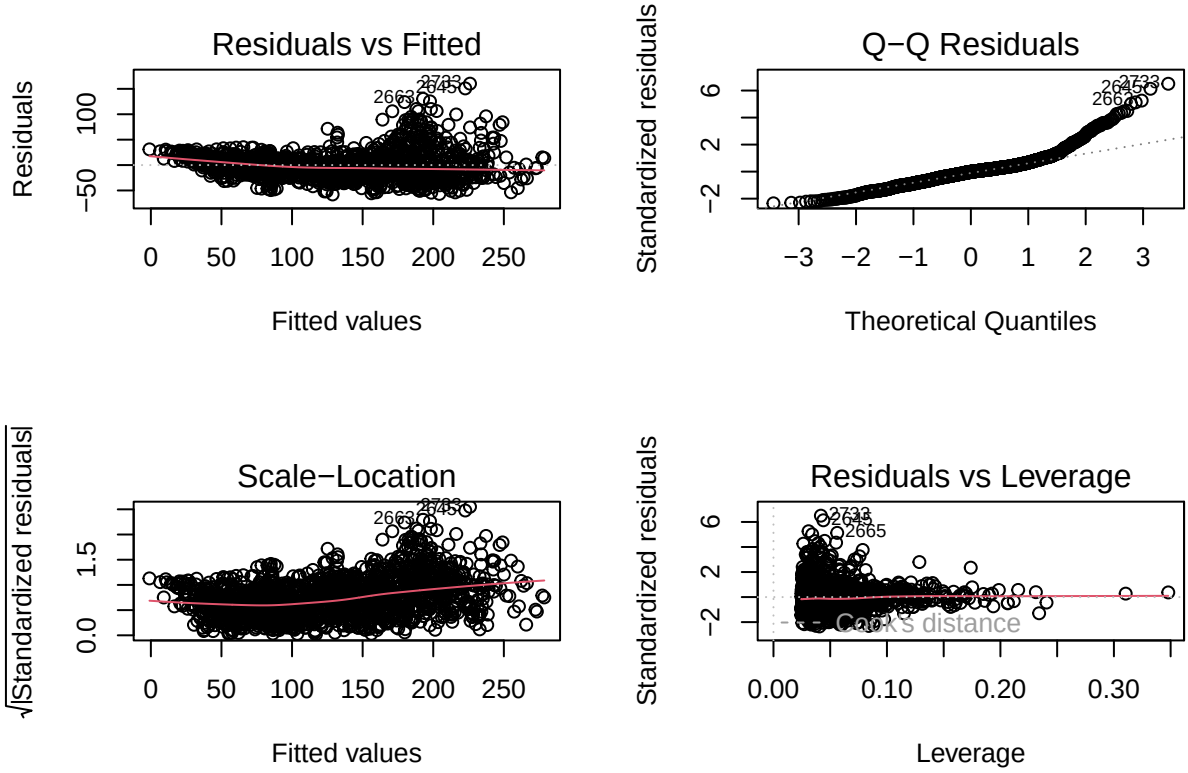

```r
plot(crimeData$Population, crimeData$HPI,
     main = "Scatter Plot of x vs y",
     xlab = "X Axis Label",
     ylab = "Y Axis Label",
     pch = 19,          # Type of point (19 is filled circle)
     col = "blue")
```




```r
# Fit the initial model with all theoretically justified variables
lm1 <- lm(HPI ~ Year + TotalCrime + City + City:TotalCrime, data = crimeData)

# Print summary statistics, focusing on adjusted R-squared for model fit
summary1 <- summary(lm1)
cat("Adjusted R-squared:", summary1$adj.r.squared, "\n")
```

```
## Adjusted R-squared: 0.8388072
```

```r
# Check model diagnostics, focusing on plots that help assess model assumptions
par(mfrow=c(2,2))  # Set up the plotting area to view multiple plots at once
plot(lm1)
```



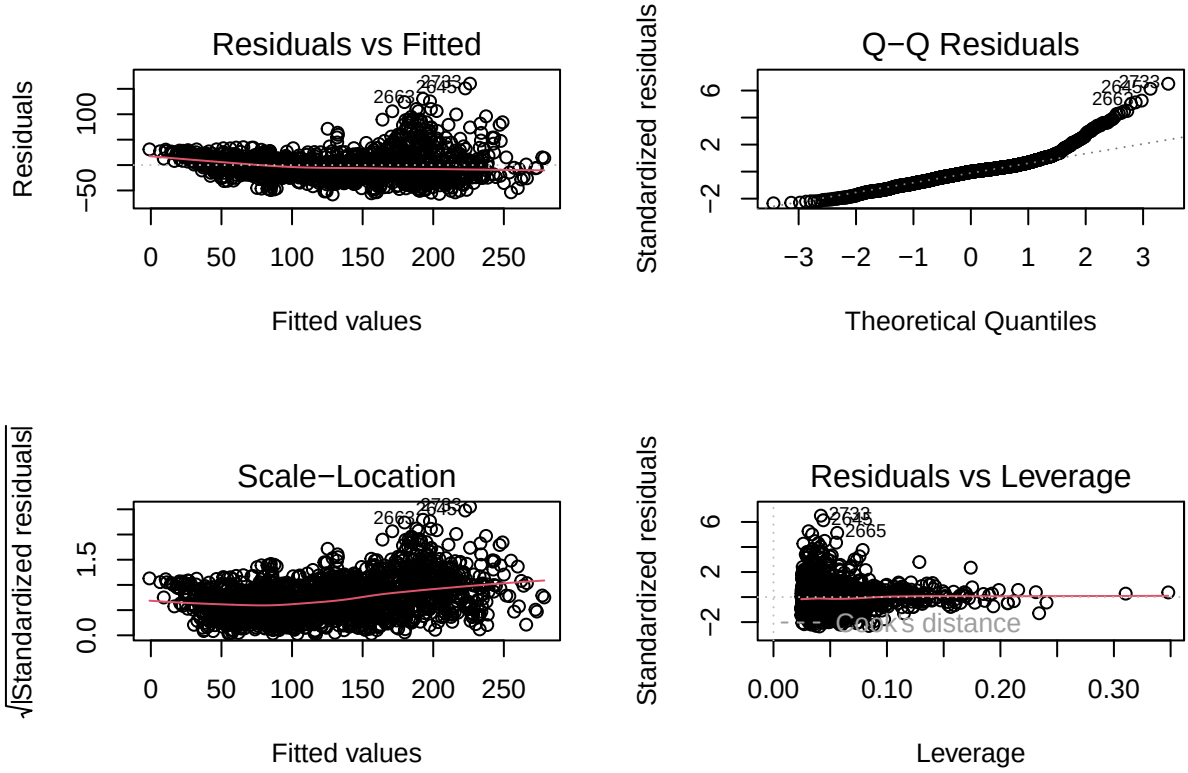

```
# Use a stepwise approach (like AIC) only if necessary, and mostly for exploratory purposes.
stepwise_approach <- stepAIC(lm1, direction = "both", trace = FALSE)
summary_stepwise <- summary(stepwise_approach)
cat("Adjusted R-squared after stepwise:", summary_stepwise$adj.r.squared, "\n")
```

```
## Adjusted R-squared after stepwise: 0.8388072
```

```
# Fit the initial model with all theoretically justified variables
lm1 <- lm(HPI ~ Year + TotalCrime + City + City:TotalCrime, data = crimeData)

# Print summary statistics, focusing on adjusted R-squared for model fit
summary1 <- summary(lm1)
cat("Adjusted R-squared:", summary1$adj.r.squared, "\n")
```

```
## Adjusted R-squared: 0.8388072
```

```
# Check model diagnostics, focusing on relevant plots
par(mfrow=c(2,2))  # Set up the plotting area to view multiple plots at once
plot(lm1)
```




```
png("Residuals_vs_Fitted.png")
plot(lm1, which = 1)  # This generates the Residuals vs Fitted plot
dev.off()
```

```
## pdf
##   2
```

```
# Load necessary library
library(car)  # For Anova and other diagnostic tools

# Fit the original model
lm1 <- lm(HPI ~ Year + TotalCrime + City + City:TotalCrime, data = crimeData)
original_f_stat <- summary(lm1)$fstatistic[1]

# Set up permutation test
set.seed(123)  # For reproducibility
n_permutations <- 1000  # Number of permutations
perm_f_stats <- numeric(n_permutations)

# Perform the permutation test
for (i in 1:n_permutations) {
  permuted_data <- crimeData
  permuted_data$HPI <- sample(permuted_data$HPI)  # Permute the response variable
  perm_model <- lm(HPI ~ Year + TotalCrime + City + City:TotalCrime, data = permuted_data)
  perm_f_stats[i] <- summary(perm_model)$fstatistic[1]
```

```
}

# Calculate the p-value
p_value <- mean(perm_f_stats >= original_f_stat)

# Display the results
cat("Original F-statistic:", original_f_stat, "\n")
```

```
## Original F-statistic: 101.9409
```

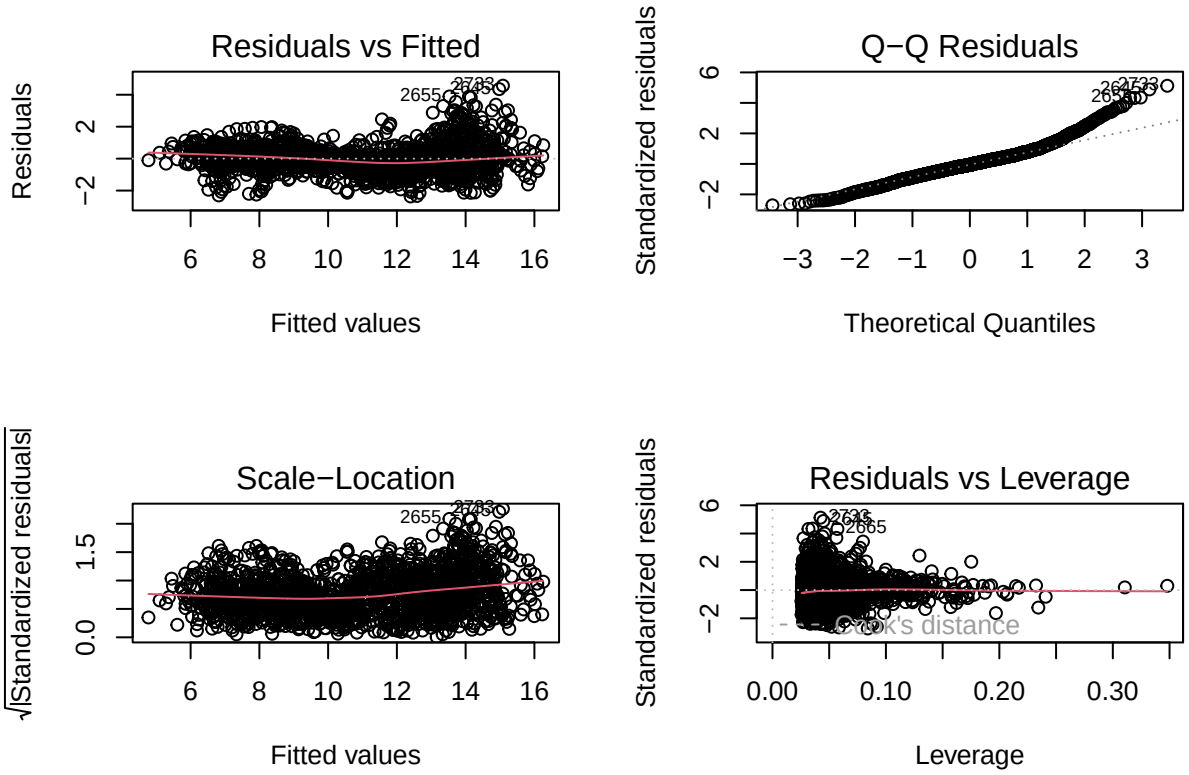
```
cat("Permutation p-value:", p_value, "\n")
```

```
## Permutation p-value: 0
```

```
# Fit the final model
lm4 <- lm(sqrt(HPI) ~ TotalCrime + poly(Year, 3, raw = FALSE) + City + City:TotalCrime, data = crimeDat
summary4 <- summary(lm4)
cat("Adjusted R-squared:", summary4$adj.r.squared, "\n")
```

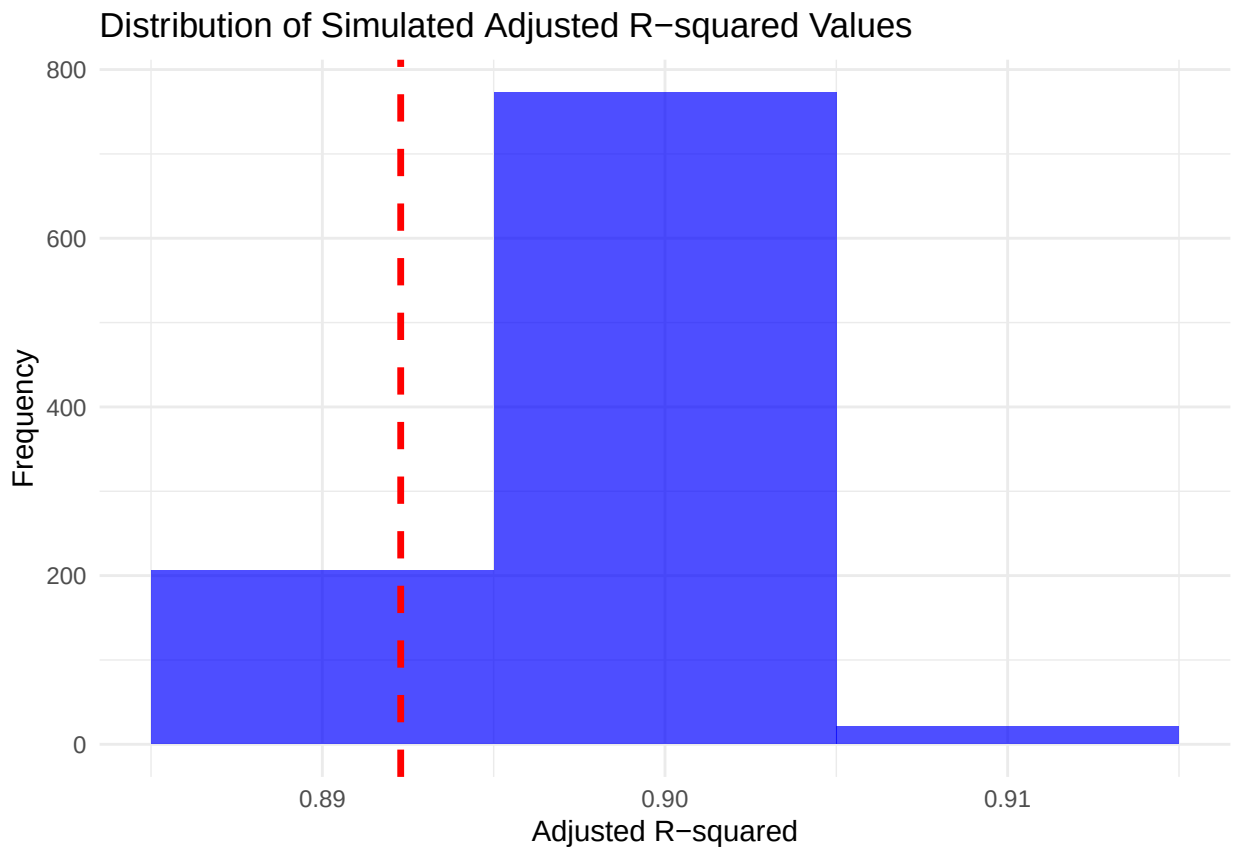
```
## Adjusted R-squared: 0.8922871
```

```
# Check model diagnostics
par(mfrow = c(2, 2))
plot(lm4)
```



```r
# Load necessary library
library(ggplot2)

# Fit the original model
lm4 <- lm(sqrt(HPI) ~ TotalCrime + poly(Year, 3, raw = FALSE) + City + City:TotalCrime, data = crimeDat
original_coef <- coef(lm4)
original_r_squared <- summary(lm4)$adj.r.squared

# Extract residual variance
residual_variance <- var(resid(lm4))

# Simulate new response variables
n_simulations <- 1000  # Number of simulations
simulated_r_squared <- numeric(n_simulations)

set.seed(123)

for (i in 1:n_simulations) {
  # Generate new response variable with the same residual variance
  simulated_response <- predict(lm4) + rnorm(n = nrow(crimeData), mean = 0, sd = sqrt(residual_variance

  # Fit the model to the simulated data
  sim_model <- lm(simulated_response ~ TotalCrime + poly(Year, 3, raw = FALSE) + City + City:TotalCrime

  # Store the adjusted R-squared value
  simulated_r_squared[i] <- summary(sim_model)$adj.r.squared
}

# Visualize the distribution of simulated R-squared values
simulated_r_squared_df <- data.frame(R_squared = simulated_r_squared)

ggplot(simulated_r_squared_df, aes(x = R_squared)) +
  geom_histogram(binwidth = 0.01, fill = "blue", alpha = 0.7) +
  geom_vline(aes(xintercept = original_r_squared), color = "red", linetype = "dashed", size = 1.2) +
  labs(title = "Distribution of Simulated Adjusted R-squared Values",
       x = "Adjusted R-squared",
       y = "Frequency") +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```


Distribution of Simulated Adjusted R−squared Values
Frequency
800
600
400
200
0
0.89
0.90
0.91
Adjusted R−squared