Linear Models – Final Project Report

By Ashish Gupta
Masters in Data Science - 2021
05/06/2020

Index

1. Introduction

Communities and Crime Unnormalized Data Set
Subject data is for crimes happening in communities in the US. Data combines socio-economic data from the '90 Census, law enforcement data from the 1990 Law Enforcement Management and Admin Stats survey, and crime data from the 1995 FBI UCR.

The variables included in the dataset involve the community, such as the percent of the population considered urban, and the median family income, and involving law enforcement, such as per capita number of police officers, and percent of officers assigned to drug units. **The crime attributes (N=18) that could be predicted are the 8 crimes considered 'Index Crimes' by the FBI)(Murders, Rape, Robbery, ...), per capita (actually per 100,000 population) versions of each, and Per Capita Violent Crimes and Per Capita Nonviolent Crimes).**

A limitation was that the LEMAS survey was of the police departments with at least 100 officers, plus a random sample of smaller departments. For our purposes, communities not found in both census and crime datasets were omitted. Many communities are missing LEMAS data. The per capita crimes variables were calculated using population values included in the 1995 FBI data (which differ from the 1990 Census values).

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

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

Data Set Characteristics:	Multivariate	**Observations**	2215
Attribute Characteristics:	Real	**Number of Attributes:**	147
Associated Tasks:	Regression	**Missing Values?**	Yes

2. Objective

- Pick 1 crime attribute to be predicted from the subject data-set.
- Build the best fitting model to predict the chosen crime variable and demonstrate the step-wise process take to build it.

3. Predictor Variable Selection based on Intuition.

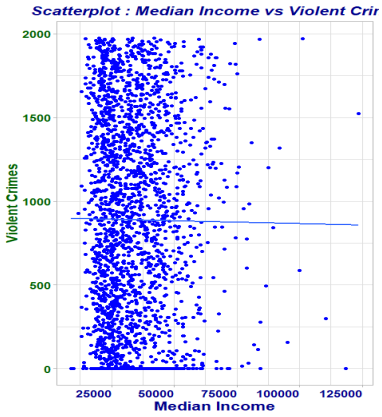
> *Pick the 4 predictor variables as you deem fit to build the first prediction model.*

I examined several variables. And decided on the following variables based on Linear Model Regression Line. I am sure given more time, we can pick the top 4 but I am sure there should be automated process to evaluate several variables for the same response variable using single regression model method.

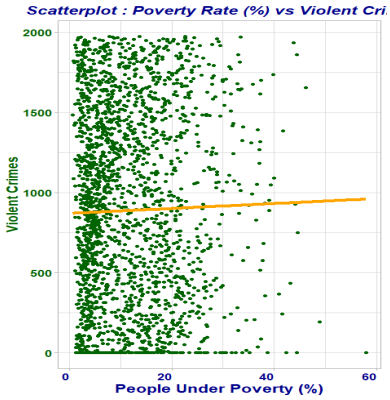
Sr. No.	Variable Name	Variable Type	Variable Description
1	ViolentCrimesPerPop	Response	Total number of violent crimes per 100K population.
2	medIncome	Predictor	Median household income.
3	PctPopUnderPov	Predictor	Percentage of people under the poverty level.
4	PctLess9thGrade	Predictor	Percentage of people 25 and over with less than a 9th grade education.
5	PctRecentImmig	Predictor	Percent of _population_ who have immigrated within the last 3 years.

4. Exploratory data analysis.

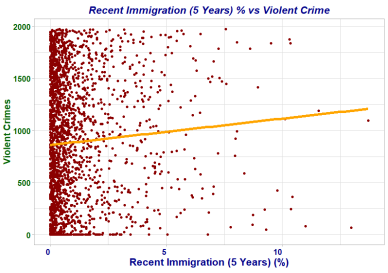
> *Build the linear model using the selected 4 predictor variables and analyze the model results.*



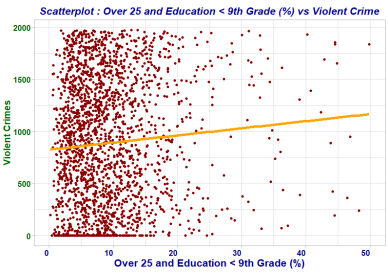
As income grows, violent crime per population comes down.



As Poverty Rate grows, violent crime per population goes up.



As Immigrated Count over Population grows, violent crime per population goes up.



As Uneducated Rate grows, violent crime per population goes up.

```
# ------------------------------------------------------------------------
# Building first model based on selected 4 prediction Variables
# ------------------------------------------------------------------------
> community_form <- as.formula(ViolentCrimesPerPop ~ medIncome + PctPopUnderPov +
                                                      PctLess9thGrade + PctRecentImmig)
> community_mdl_s4 <- lm(community_form, community_ds)
> summary(community_mdl_s4)

Call:
lm(formula = community_form, data = community_ds)

Residuals:
     Min       1Q   Median       3Q      Max
-1200.45  -562.11    -6.96   528.26  1132.30

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      7.874e+02  7.915e+01   9.948  < 2e-16 ***
medIncome        8.612e-04  1.586e-03   0.543  0.58710
PctPopUnderPov  -1.927e+00  2.567e+00  -0.751  0.45292
PctLess9thGrade  8.313e+00  2.730e+00   3.045  0.00235 **
PctRecentImmig   1.611e+01  9.025e+00   1.785  0.07447 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 613.5 on 2210 degrees of freedom
Multiple R-squared:  0.008713, Adjusted R-squared:  0.006919
F-statistic: 4.856 on 4 and 2210 DF,  p-value: 0.000671
```

Model Summary Findings

- Out of 4 predictor variables, 3 are with p-value greater than significant values.
- Model's Adjusted R^2 is .007% which is less than 1% prediction accuracy and is very low.

Conclusions

- Based on the model summary, it is clearly evident that this model is grossly inadequate and should be discarded.
- We need to try another set of predictor variables (perhaps larger than 4) based on understanding the data-set and predictor variables definitions.

5. Automated Model Selection

> *Use the automated model selection techniques to pick the best fitting model*
> *Compare all models and present the best fitting model.*

Since the first discretionary approach of building the linear model using selected 4 predictor variables, resulted in a model with Adjusted R2 Coefficient = 0.007 hence I chose to discard it. Now I am taking 3 different approaches and will pick one which provides best fitting model. I am selecting 3 sets of predictor variables to generate 3 different models and will compare them to pick the best one.

I have taken Automated model selection approach to generate following models on different set of predictor variables.

A. ML Model using fastbw() based on all clean predictors present in dataset.
B. ML Model using stepAIC() based on all clean predictors present in dataset.
C. ML Model using fastbw() based on subjective/discretionary approach.
D. ML Model using stepAIC() based on subjective/discretionary approach.

Then I plan to find the best fitting model for each approach and finally overall best fitting model out of these 2 methods.

Model Selection – fastbw()

A. Approach 1: ML Model using fastbw() based on all clean predictor variables (101) after removing rows/variables containing NAs/?.

Sr. No.	Fastbw - All Predictors		
Adjusted R2	Initial List	Santized List (-(NA, ?)) 0.6583	Final List 0.6556
Count	124	101	24
1	PctForeignBorn	PctForeignBorn	PctKids2Par
2	PctBornSameState	PctBornSameState	PctWorkMom
3	PctFam2Par	PctFam2Par	PctKidsBornNeverMar
4	PctKids2Par	PctKids2Par	pctUrban
5	PctYoungKids2Par	PctYoungKids2Par	LemasPctOfficDrugUn
6	PctTeen2Par	PctTeen2Par	PctLess9thGrade
7	PctWorkMomYoungKids	PctWorkMomYoungKids	PctNotHSGrad
8	PctWorkMom	PctWorkMom	PctEmploy
9	NumKidsBornNeverMar	NumKidsBornNeverMar	PctEmplManu
10	PctKidsBornNeverMar	PctKidsBornNeverMar	PersPerOwnOccHous
11	numbUrban	numbUrban	PersPerRentOccHous
12	pctUrban	pctUrban	PctPersOwnOccup
13	LandArea	LandArea	PctPersDenseHous
14	PopDens	PopDens	HousVacant
15	PctUsePubTrans	PctUsePubTrans	PctHousOwnOcc
16	PolicCars		PctVacantBoarded

17	PolicOperBudg		pctWWage
18	LemasPctPolicOnPatr		pctWRetire
19	LemasGangUnitDeploy		MalePctDivorce
20	LemasPctOfficDrugUn	LemasPctOfficDrugUn	TotalPctDiv
21	PolicBudgPerPop		RentLowQ
22	PctLess9thGrade	PctLess9thGrade	MedRent
23	PctNotHSGrad	PctNotHSGrad	MedOwnCostPctIncNoMtg
24	PctBSorMore	PctBSorMore	racepctblack
25	PctUnemployed	PctUnemployed	
26	PctEmploy	PctEmploy	
27	PctEmplManu	PctEmplManu	
28	PctEmplProfServ	PctEmplProfServ	
29	PctOccupManu	PctOccupManu	
30	PctOccupMgmtProf	PctOccupMgmtProf	
31	PersPerFam	PersPerFam	
32	NumInShelters	NumInShelters	
33	NumStreet	NumStreet	
34	householdsize	householdsize	
35	PctLargHouseFam	PctLargHouseFam	
36	PctLargHouseOccup	PctLargHouseOccup	
37	PersPerOccupHous	PersPerOccupHous	
38	PersPerOwnOccHous	PersPerOwnOccHous	
39	PersPerRentOccHous	PersPerRentOccHous	
40	PctPersOwnOccup	PctPersOwnOccup	
41	PctPersDenseHous	PctPersDenseHous	
42	PctHousLess3BR	PctHousLess3BR	
43	MedNumBR	MedNumBR	
44	HousVacant	HousVacant	
45	PctHousOccup	PctHousOccup	
46	PctHousOwnOcc	PctHousOwnOcc	
47	PctVacantBoarded	PctVacantBoarded	
48	PctVacMore6Mos	PctVacMore6Mos	
49	MedYrHousBuilt	MedYrHousBuilt	
50	PctHousNoPhone	PctHousNoPhone	
51	PctWOFullPlumb	PctWOFullPlumb	
52	OwnOccLowQuart	OwnOccLowQuart	
53	OwnOccMedVal	OwnOccMedVal	
54	OwnOccHiQuart	OwnOccHiQuart	
55	OwnOccQrange	OwnOccQrange	
56	NumImmig	NumImmig	
57	PctImmigRecent	PctImmigRecent	
58	PctImmigRec5	PctImmigRec5	
59	PctImmigRec8	PctImmigRec8	
60	PctImmigRec10	PctImmigRec10	
61	PctRecentImmig	PctRecentImmig	

62	PctRecImmig5	PctRecImmig5	
63	PctRecImmig8	PctRecImmig8	
64	PctRecImmig10	PctRecImmig10	
65	medIncome	medIncome	
66	pctWWage	pctWWage	
67	pctWFarmSelf	pctWFarmSelf	
68	pctWInvInc	pctWInvInc	
69	pctWSocSec	pctWSocSec	
70	pctWPubAsst	pctWPubAsst	
71	pctWRetire	pctWRetire	
72	medFamInc	medFamInc	
73	perCapInc	perCapInc	
74	whitePerCap	whitePerCap	
75	blackPerCap	blackPerCap	
76	indianPerCap	indianPerCap	
77	AsianPerCap	AsianPerCap	
78	OtherPerCap		
79	HispPerCap	HispPerCap	
80	PctSpeakEnglOnly	PctSpeakEnglOnly	
81	PctNotSpeakEnglWell	PctNotSpeakEnglWell	
82	MalePctDivorce	MalePctDivorce	
83	MalePctNevMarr	MalePctNevMarr	
84	FemalePctDiv	FemalePctDiv	
85	TotalPctDiv	TotalPctDiv	
86	RentLowQ	RentLowQ	
87	RentMedian	RentMedian	
88	RentHighQ	RentHighQ	
89	RentQrange	RentQrange	
90	MedRent	MedRent	
91	MedRentPctHousInc	MedRentPctHousInc	
92	MedOwnCostPctInc	MedOwnCostPctInc	
93	MedOwnCostPctIncNoMtg	MedOwnCostPctIncNoMtg	
94	LemasSwornFT		
95	LemasSwFTPerPop		
96	LemasSwFTFieldOps		
97	LemasSwFTFieldPerPop		
98	LemasTotalReq		
99	LemasTotReqPerPop		
100	PolicReqPerOffic		
101	PolicPerPop		
102	RacialMatchCommPol		
103	PctPolicWhite		
104	PctPolicBlack		
105	PctPolicHisp		
106	PctPolicAsian		

107	PctPolicMinor		
108	OfficAssgnDrugUnits		
109	NumKindsDrugsSeiz		
110	PolicAveOTWorked		
111	population	population	
112	racepctblack	racepctblack	
113	racePctWhite	racePctWhite	
114	racePctAsian	racePctAsian	
115	racePctHisp	racePctHisp	
116	agePct12t21	agePct12t21	
117	agePct12t29	agePct12t29	
118	agePct16t24	agePct16t24	
119	agePct65up	agePct65up	
120	NumUnderPov	NumUnderPov	
121	PctPopUnderPov	PctPopUnderPov	
122	PctSameHouse85	PctSameHouse85	
123	PctSameCity85	PctSameCity85	
124	PctSameState85	PctSameState85	

B. Approach 2: ML Model using stepAIC() based on all clean predictor variables (101) after removing rows/variables containing NAs/?.

Sr. No.	Week 3 - stepAIC - All Predictors		
	Initial List	Santized List (-(NA, ?)	Final List (Preferred)
R2(a) / AIC			.6637 / 23479.43
Count	124	101	44
1	PctForeignBorn	PctForeignBorn	agePct12t21
2	PctBornSameState	PctBornSameState	PctImmigRecent
3	PctFam2Par	PctFam2Par	medFamInc
4	PctKids2Par	PctKids2Par	NumKidsBornNeverMar
5	PctYoungKids2Par	PctYoungKids2Par	PctPopUnderPov
6	PctTeen2Par	PctTeen2Par	OwnOccHiQuart
7	PctWorkMomYoungKids	PctWorkMomYoungKids	NumImmig
8	PctWorkMom	PctWorkMom	medIncome
9	NumKidsBornNeverMar	NumKidsBornNeverMar	pctWFarmSelf
10	PctKidsBornNeverMar	PctKidsBornNeverMar	numbUrban
11	numbUrban	numbUrban	pctWInvInc
12	pctUrban	pctUrban	AsianPerCap
13	LandArea	LandArea	PctLargHouseOccup
14	PopDens	PopDens	NumInShelters
15	PctUsePubTrans	PctUsePubTrans	PctRecImmig5
16	PolicCars		PctNotHSGrad
17	PolicOperBudg		PctEmplManu
18	LemasPctPolicOnPatr		PctVacMore6Mos
19	LemasGangUnitDeploy		TotalPctDiv

20	LemasPctOfficDrugUn	LemasPctOfficDrugUn	PctForeignBorn
21	PolicBudgPerPop		PopDens
22	PctLess9thGrade	PctLess9thGrade	LemasPctOfficDrugUn
23	PctNotHSGrad	PctNotHSGrad	RentLowQ
24	PctBSorMore	PctBSorMore	MedRent
25	PctUnemployed	PctUnemployed	agePct12t29
26	PctEmploy	PctEmploy	MalePctNevMarr
27	PctEmplManu	PctEmplManu	PersPerRentOccHous
28	PctEmplProfServ	PctEmplProfServ	PersPerOwnOccHous
29	PctOccupManu	PctOccupManu	PctKids2Par
30	PctOccupMgmtProf	PctOccupMgmtProf	PctEmploy
31	PersPerFam	PersPerFam	pctWRetire
32	NumInShelters	NumInShelters	MalePctDivorce
33	NumStreet	NumStreet	PctLess9thGrade
34	householdsize	householdsize	pctWWage
35	PctLargHouseFam	PctLargHouseFam	PctVacantBoarded
36	PctLargHouseOccup	PctLargHouseOccup	PctWorkMom
37	PersPerOccupHous	PersPerOccupHous	PctHousOwnOcc
38	PersPerOwnOccHous	PersPerOwnOccHous	HousVacant
39	PersPerRentOccHous	PersPerRentOccHous	PctPersOwnOccup
40	PctPersOwnOccup	PctPersOwnOccup	PctKidsBornNeverMar
41	PctPersDenseHous	PctPersDenseHous	pctUrban
42	PctHousLess3BR	PctHousLess3BR	PctPersDenseHous
43	MedNumBR	MedNumBR	MedOwnCostPctIncNoMtg
44	HousVacant	HousVacant	racepctblack
45	PctHousOccup	PctHousOccup	
46	PctHousOwnOcc	PctHousOwnOcc	
47	PctVacantBoarded	PctVacantBoarded	
48	PctVacMore6Mos	PctVacMore6Mos	
49	MedYrHousBuilt	MedYrHousBuilt	
50	PctHousNoPhone	PctHousNoPhone	
51	PctWOFullPlumb	PctWOFullPlumb	
52	OwnOccLowQuart	OwnOccLowQuart	
53	OwnOccMedVal	OwnOccMedVal	
54	OwnOccHiQuart	OwnOccHiQuart	
55	OwnOccQrange	OwnOccQrange	
56	NumImmig	NumImmig	
57	PctImmigRecent	PctImmigRecent	
58	PctImmigRec5	PctImmigRec5	
59	PctImmigRec8	PctImmigRec8	
60	PctImmigRec10	PctImmigRec10	
61	PctRecentImmig	PctRecentImmig	
62	PctRecImmig5	PctRecImmig5	
63	PctRecImmig8	PctRecImmig8	
64	PctRecImmig10	PctRecImmig10	

65	medIncome	medIncome	
66	pctWWage	pctWWage	
67	pctWFarmSelf	pctWFarmSelf	
68	pctWInvInc	pctWInvInc	
69	pctWSocSec	pctWSocSec	
70	pctWPubAsst	pctWPubAsst	
71	pctWRetire	pctWRetire	
72	medFamInc	medFamInc	
73	perCapInc	perCapInc	
74	whitePerCap	whitePerCap	
75	blackPerCap	blackPerCap	
76	indianPerCap	indianPerCap	
77	AsianPerCap	AsianPerCap	
78	OtherPerCap		
79	HispPerCap	HispPerCap	
80	PctSpeakEnglOnly	PctSpeakEnglOnly	
81	PctNotSpeakEnglWell	PctNotSpeakEnglWell	
82	MalePctDivorce	MalePctDivorce	
83	MalePctNevMarr	MalePctNevMarr	
84	FemalePctDiv	FemalePctDiv	
85	TotalPctDiv	TotalPctDiv	
86	RentLowQ	RentLowQ	
87	RentMedian	RentMedian	
88	RentHighQ	RentHighQ	
89	RentQrange	RentQrange	
90	MedRent	MedRent	
91	MedRentPctHousInc	MedRentPctHousInc	
92	MedOwnCostPctInc	MedOwnCostPctInc	
93	MedOwnCostPctIncNoMtg	MedOwnCostPctIncNoMtg	
94	LemasSwornFT		
95	LemasSwFTPerPop		
96	LemasSwFTFieldOps		
97	LemasSwFTFieldPerPop		
98	LemasTotalReq		
99	LemasTotReqPerPop		
100	PolicReqPerOffic		
101	PolicPerPop		
102	RacialMatchCommPol		
103	PctPolicWhite		
104	PctPolicBlack		
105	PctPolicHisp		
106	PctPolicAsian		
107	PctPolicMinor		
108	OfficAssgnDrugUnits		
109	NumKindsDrugsSeiz		

110	PolicAveOTWorked		
111	population	population	
112	racepctblack	racepctblack	
113	racePctWhite	racePctWhite	
114	racePctAsian	racePctAsian	
115	racePctHisp	racePctHisp	
116	agePct12t21	agePct12t21	
117	agePct12t29	agePct12t29	
118	agePct16t24	agePct16t24	
119	agePct65up	agePct65up	
120	NumUnderPov	NumUnderPov	
121	PctPopUnderPov	PctPopUnderPov	
122	PctSameHouse85	PctSameHouse85	
123	PctSameCity85	PctSameCity85	
124	PctSameState85	PctSameState85	

C. Approach 3: ML Model using fastbw() based on selected predictor variables (8).
D. Approach 4: ML Model using stepAIC() based on selected predictor variables (8).

Sr. No.	Fastbw and AIC - Selected Predictors (8)		
Adjusted R2/AIC	Initial List	fastbw 0.6397	AIC 0.6397 / 18912.68
Count	8	8	8
1	racepctblack	racepctblack	racepctblack
2	PctPersDenseHous	PctPersDenseHous	PctPersDenseHous
3	pctUrban	pctUrban	pctUrban
4	PctKidsBornNeverMar	PctKidsBornNeverMar	PctKidsBornNeverMar
5	HousVacant	HousVacant	HousVacant
6	pctWWage	pctWWage	pctWWage
7	MalePctDivorce	MalePctDivorce	MalePctDivorce
8	pctWRetire	pctWRetire	pctWRetire

> Does it match your intuition?
Yes. It matches my intuition based on predictor variables selected in the final model.

> How do the automatically selected models compare to your model from Step 2?
Its far more sophisticated model than the one from step 2. It has final AIC Value as 18912 and Adjusted R2 = 0.6397 with 8 variables.

> Which model will you choose to proceed with?
I select the Approach 4 as best fitting model which has 8 final predictor variables. It started with selected 19 predictor variables to final model with 8 predictor variables. My chosen predictive model is stepAIC one with AIC Value as 18912 and Adjusted R2 = 0.6397 with 8 variables.

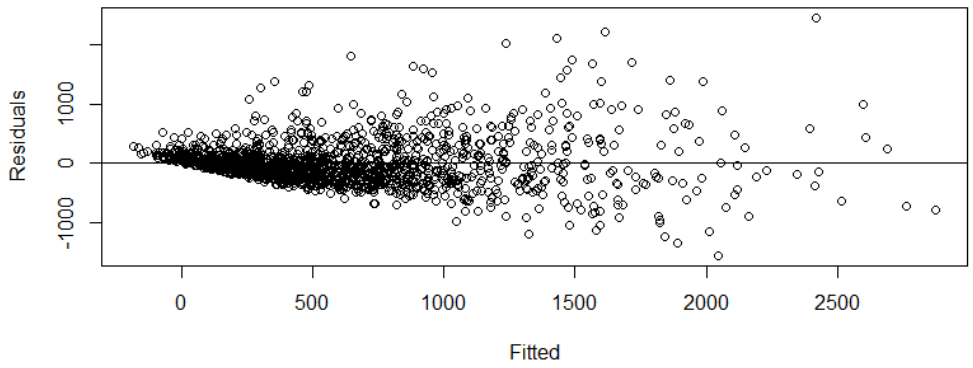
Model Diagnostics

> *Apply Model Diagnostics to evaluate the best fit model.*

```
> # ------------------------------------------------------------------------------
> # checking for Multi-colinearity
> # Multicolinearity is not present here.
> # ------------------------------------------------------------------------------
> comm_vif <- vif(community_mdl_s8_AIC)

> comm_vif
      racepctblack     PctPersDenseHous            pctUrban PctKidsBornNeverMar
          3.394204             1.475989             1.212607             4.591688
       HousVacant            pctWWage      MalePctDivorce           pctWRetire
          1.090446             2.089594             1.425718             1.835077
```

As VIF Factors for all predictor variables are smaller than 10, hence we can say conclusively that multicollinearity is not present in this model

```
> # ------------------------------------------------------------------------------
> # Fitted values vs. residuals plot Comparison for Constant Error Variance
> # As per plot, errors have constant variance and are not random.
> # Heteroscedasticity is present
> # ------------------------------------------------------------------------------

> fitted_values <- fitted(community_mdl_s8_AIC)
> residual_values <- residuals(community_mdl_s8_AIC)

> plot(fitted_values, residual_values, xlab = "Fitted", ylab = "Residuals")
> abline(community_mdl_s8_AIC)
```



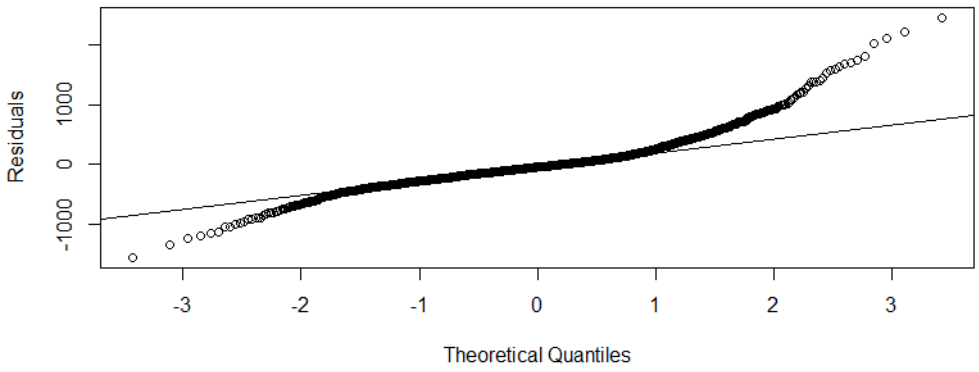
It seems, there is constant variance present in the model, as plot presents a clearly a cone shaped plot. As the fitted values grow, residuals seem to go proportionately.

```
> # ------------------------------------------------------------------------------
> # Formal Test for Constant Variance: BP Test
> # it also confirms that Heteroscedasticity is present
> # ------------------------------------------------------------------------------

> comm_bp <- bptest(community_mdl_s8_AIC)
> comm_bp

        studentized Breusch-Pagan test
```

```
data:  community_mdl_s8_AIC
BP = 278.76, df = 8, p-value < 2.2e-16
```

Since p-value is far smaller than the significant value hence we can safely say that heteroscedastic ity is clearly present in the model.

```
> # ------------------------------------------------------------------------------
> # checking for Normality
> # Trend line is short-tailed
> # Non-normality is present
> # ------------------------------------------------------------------------------

> qqnorm(residuals(community_mdl_s8_AIC), ylab = "Residuals", main="")
> qqline(residuals(community_mdl_s8_AIC))
```



It seems to be long-tailed plot here hence we can accept non-normalcy to be present here. We mig ht need to resample to confirm this inference.

```
> # ------------------------------------------------------------------------------
> # checking for Shapiro Test: Formal test for normality
> # p-value is less than significant value so normality is not present.
> # ------------------------------------------------------------------------------

> shapiro.test(residuals(community_mdl_s8_AIC))

        Shapiro-Wilk normality test
data:  residuals(community_mdl_s8_AIC)
W = 0.90274, p-value < 2.2e-16
```

Since p-value is far smaller than the significant value hence we can safely say that normality is not present in the model.

```
> # ------------------------------------------------------------------------------
> # checking for Durban-Watson Test : Formal Test for co-related errors
> # Co-relation is present
> # ------------------------------------------------------------------------------
> dwtest(community_mdl_s8_AIC)

        Durbin-Watson test
data:  community_mdl_s8_AIC
DW = 1.9559, p-value = 0.1886
alternative hypothesis: true autocorrelation is greater than 0
```

Positive Co-relation is clearly evident as It is verified by Durbin-Watson Test.

6. Outliers and Influential Observations

> *Check for Outliers and Influential Observations and take corrective action if needed.*

```
# ------------------------------------------------------------------------
# Calculating Leverages
# ------------------------------------------------------------------------
> # Generating Design Matrix
> X <-model.matrix(community_mdl_s8_AIC)
> n <-dim(X)[1]
> p <-dim(X)[2]

> #Identifying high leverage observations by hand.
> hatmat <- X%*%solve(t(X)%*%X)%*%t(X)
> community_ds_leverages <- diag(hatmat)

# Isolating high leverage observations into a separate dataset
> community_ds[which(community_ds_leverages > 2*p/n),] -> comm_ds_influencers
> glimpse(comm_ds_influencers)

Observations: 150
Variables: 102
$ ViolentCrimesPerPop   <dbl> 374.07, 2097.71, 609.81, 420.91, 1087.25, 1279.60, ...
$ PctForeignBorn        <dbl> 45.19, 28.45, 7.18, 4.97, 34.82, 6.61, 0.97, 2.92, ...
$ PctBornSameState      <dbl> 29.68, 53.09, 58.08, 92.04, 26.78, 73.58, 79.97, 66...
$ PctFam2Par            <dbl> 77.10, 57.64, 82.90, 72.16, 75.73, 53.45, 82.16, 89...
$ PctKids2Par           <dbl> 74.78, 52.24, 75.36, 61.18, 77.69, 47.22, 77.61, 89...
$ PctYoungKids2Par      <dbl> 86.01, 67.42, 88.30, 76.35, 90.56, 62.33, 83.46, 95...
$ PctTeen2Par           <dbl> 79.01, 59.25, 80.68, 72.71, 81.23, 56.66, 74.35, 90...
$ PctWorkMomYoungKids   <dbl> 52.45, 47.37, 48.57, 39.62, 49.30, 54.09, 68.08, 55...
$ PctWorkMom            <dbl> 59.33, 56.42, 57.79, 44.65, 55.66, 62.23, 72.52, 65...
$ NumKidsBornNeverMar   <int> 3034, 527557, 536, 489, 223, 138864, 125, 228, 1179...
$ PctKidsBornNeverMar   <dbl> 2.62, 10.50, 3.46, 3.45, 1.04, 11.53, 1.16, 0.43, 3...
$ numbUrban             <int> 180038, 7322564, 23302, 0, 31971, 1585577, 9827, 75...
$ pctUrban              <dbl> 100.00, 100.00, 100.00, 0.00, 100.00, 100.00, 73.60...
$ LandArea              <dbl> 31.7, 320.1, 14.2, 9.6, 5.9, 140.0, 36.1, 20.7, 129...
$ PopDens               <dbl> 5677.3, 22878.2, 1641.4, 1345.2, 5437.2, 11326.0, 3...
$ PctUsePubTrans        <dbl> 4.15, 54.33, 0.27, 1.23, 4.31, 29.31, 0.20, 2.34, 1...
$ LemasPctOfficDrugUn   <dbl> 5.88, 6.91, 0.00, 0.00, 4.76, 4.19, 0.00, 0.00, 4.8...
$ PctLess9thGrade       <dbl> 11.54, 14.10, 4.93, 42.31, 3.60, 11.29, 8.16, 1.05,...
$ PctNotHSGrad          <dbl> 22.83, 31.68, 24.25, 61.80, 10.57, 35.69, 25.83, 6....
$ PctBSorMore           <dbl> 28.55, 22.98, 10.32, 6.31, 46.95, 15.22, 13.65, 29....
$ PctUnemployed         <dbl> 6.95, 8.98, 5.10, 12.26, 3.80, 9.62, 5.41, 3.56, 6....
$ PctEmploy             <dbl> 60.04, 56.12, 48.14, 44.19, 60.76, 52.77, 55.46, 72...
$ PctEmplManu           <dbl> 14.43, 11.41, 21.21, 16.67, 8.37, 13.58, 24.72, 12....
$ PctEmplProfServ       <dbl> 24.78, 28.40, 14.52, 21.07, 28.42, 29.55, 22.89, 22...
$ PctOccupManu          <dbl> 9.07, 11.64, 14.49, 22.14, 1.37, 14.13, 17.76, 8.39...
$ PctOccupMgmtProf      <dbl> 33.09, 30.55, 23.72, 14.35, 51.85, 24.81, 27.61, 31...
$ PersPerFam            <dbl> 3.22, 3.27, 3.49, 3.96, 2.87, 3.26, 3.02, 4.10, 2.9...
$ NumInShelters         <int> 82, 23383, 0, 0, 0, 3416, 4, 0, 1553, 0, 0, 0, 4597...
$ NumStreet             <int> 17, 10447, 0, 0, 5, 1069, 0, 0, 149, 0, 0, 0, 3109,...
$ householdsize         <dbl> 2.62, 2.60, 4.17, 3.59, 2.20, 2.63, 2.82, 3.86, 2.3...
$ PctLargHouseFam       <dbl> 6.98, 9.16, 10.56, 18.52, 4.05, 8.48, 3.26, 20.44, ...
$ PctLargHouseOccup     <dbl> 4.60, 5.71, 8.94, 15.67, 2.25, 5.42, 2.66, 18.40, 2...
$ PersPerOccupHous      <dbl> 2.59, 2.54, 3.27, 3.56, 2.19, 2.56, 2.69, 3.85, 2.2...
$ PersPerOwnOccHous     <dbl> 2.63, 2.80, 3.29, 3.53, 2.71, 2.75, 2.72, 3.94, 2.4...
$ PersPerRentOccHous    <dbl> 2.56, 2.43, 3.19, 3.63, 1.78, 2.25, 2.51, 3.21, 2.0...
$ PctPersOwnOccup       <dbl> 39.38, 31.59, 82.83, 68.41, 54.32, 66.55, 83.85, 89...
$ PctPersDenseHous      <dbl> 18.31, 12.30, 6.06, 20.25, 3.27, 4.69, 1.20, 3.24, ...
$ PctHousLess3BR        <dbl> 75.17, 73.76, 22.51, 50.24, 61.04, 42.28, 32.59, 13...
$ MedNumBR              <int> 2, 2, 3, 2, 2, 3, 3, 4, 2, 4, 3, 3, 2, 2, 2, 3, 3, ...
$ HousVacant            <int> 3510, 172768, 193, 554, 1159, 71824, 132, 687, 1110...
```

```
$ PctHousOccup          <dbl> 95.13, 94.23, 96.66, 86.60, 92.63, 89.36, 97.29, 96...
$ PctHousOwnOcc         <dbl> 38.71, 28.64, 82.40, 68.98, 43.85, 61.95, 82.69, 87...
$ PctVacantBoarded      <dbl> 2.25, 4.83, 3.63, 25.81, 1.38, 21.96, 2.27, 1.31, 5...
$ PctVacMore6Mos        <dbl> 17.69, 40.33, 26.42, 75.63, 31.23, 59.26, 40.91, 32...
$ MedYrHousBuilt        <int> 1962, 1946, 1970, 1960, 1949, 1939, 1957, 1978, 194...
$ PctHousNoPhone        <dbl> 1.53, 7.33, 2.50, 17.21, 0.40, 4.25, 2.30, 0.86, 4....
$ PctWOFullPlumb        <dbl> 0.46, 1.25, 0.24, 3.15, 0.20, 0.88, 1.35, 0.26, 0.6...
$ OwnOccLowQuart        <int> 249800, 149700, 153000, 19200, 500001, 27900, 46700...
$ OwnOccMedVal          <int> 343600, 189600, 203400, 30700, 500001, 49400, 66300...
$ OwnOccHiQuart         <int> 469800, 244600, 259300, 43700, 500001, 77100, 90100...
$ OwnOccQrange          <int> 220000, 94900, 106300, 24500, 0, 49200, 43400, 4080...
$ NumImmig              <int> 81352, 2082931, 1672, 644, 11133, 104814, 130, 2193...
$ PctImmigRecent        <dbl> 30.33, 15.87, 6.76, 0.00, 13.57, 16.38, 3.85, 10.21...
$ PctImmigRec5          <dbl> 41.41, 25.51, 16.87, 3.26, 22.29, 24.30, 3.85, 13.2...
$ PctImmigRec8          <dbl> 50.77, 36.37, 18.66, 6.83, 31.93, 33.12, 11.54, 18....
$ PctImmigRec10         <dbl> 60.40, 45.77, 33.19, 20.96, 41.76, 42.58, 11.54, 26...
$ PctRecentImmig        <dbl> 13.71, 4.51, 0.48, 0.00, 4.73, 1.08, 0.04, 0.30, 1....
$ PctRecImmig5          <dbl> 18.71, 7.26, 1.21, 0.16, 7.76, 1.61, 0.04, 0.39, 2....
$ PctRecImmig8          <dbl> 22.94, 10.34, 1.34, 0.34, 11.12, 2.19, 0.11, 0.53, ...
$ PctRecImmig10         <dbl> 27.29, 13.02, 2.38, 1.05, 14.54, 2.81, 0.11, 0.79, ...
$ medIncome             <int> 34372, 29823, 51594, 16180, 54348, 24603, 32713, 43...
$ pctWWage              <dbl> 76.17, 73.57, 88.74, 69.32, 73.00, 70.12, 73.70, 90...
$ pctWFarmSelf          <dbl> 0.62, 0.48, 2.55, 1.59, 0.56, 0.35, 1.76, 0.85, 0.7...
$ pctWInvInc            <dbl> 40.04, 35.38, 38.24, 13.09, 55.83, 32.15, 46.04, 45...
$ pctWSocSec            <dbl> 20.30, 24.60, 17.30, 32.32, 28.00, 31.63, 32.15, 10...
$ pctWPubAsst           <dbl> 11.27, 13.12, 5.92, 24.32, 3.47, 13.98, 5.23, 2.59,...
$ pctWRetire            <dbl> 11.51, 12.56, 12.75, 9.20, 9.02, 18.20, 24.43, 9.81...
$ medFamInc             <int> 39652, 34360, 52659, 18788, 83272, 30140, 36199, 45...
$ perCapInc             <int> 17966, 16281, 15142, 5740, 55463, 12091, 14243, 128...
$ whitePerCap           <int> 19362, 21972, 16771, 6694, 57381, 15027, 13832, 128...
$ blackPerCap           <int> 17693, 10505, 4590, 5575, 30294, 9061, 12981, 14276...
$ indianPerCap          <int> 20931, 10861, 6108, 0, 13326, 10146, 5114, 15114, 8...
$ AsianPerCap           <int> 16696, 12851, 12843, 0, 39787, 8285, 106165, 10192,...
$ HispPerCap            <int> 11182, 8420, 9872, 5022, 28719, 6053, 5742, 11001, ...
$ PctSpeakEnglOnly      <dbl> 46.84, 59.04, 84.60, 15.13, 64.34, 86.31, 97.24, 94...
$ PctNotSpeakEnglWell   <dbl> 15.46, 9.84, 2.25, 14.64, 4.86, 2.63, 0.48, 0.55, 2...
$ MalePctDivorce        <dbl> 8.68, 9.13, 8.43, 8.53, 8.33, 10.62, 7.80, 4.95, 14...
$ MalePctNevMarr        <dbl> 34.45, 41.00, 42.59, 30.36, 32.58, 42.11, 22.04, 25...
$ FemalePctDiv          <dbl> 13.33, 14.02, 12.03, 12.40, 15.23, 14.12, 9.11, 7.7...
$ TotalPctDiv           <dbl> 11.12, 11.77, 9.94, 10.59, 12.23, 12.53, 8.48, 6.38...
$ RentLowQ              <int> 511, 320, 458, 105, 660, 252, 158, 293, 269, 257, 7...
$ RentMedian            <int> 626, 448, 617, 179, 902, 358, 248, 397, 340, 367, 8...
$ RentHighQ             <int> 763, 615, 843, 243, 1001, 483, 322, 538, 418, 544, ...
$ RentQrange            <int> 252, 295, 385, 138, 341, 231, 164, 245, 149, 287, 2...
$ MedRent               <int> 688, 496, 714, 265, 925, 452, 341, 500, 397, 512, 8...
$ MedRentPctHousInc     <dbl> 30.5, 25.7, 28.2, 28.7, 29.5, 29.8, 24.3, 22.8, 25....
$ MedOwnCostPctInc      <dbl> 25.3, 21.8, 25.1, 21.4, 24.0, 19.3, 16.9, 21.9, 20....
$ MedOwnCostPctIncNoMtg <dbl> 11.6, 14.1, 11.9, 14.0, 12.9, 14.8, 12.1, 11.4, 14....
$ population            <int> 180038, 7322564, 23302, 12849, 31971, 1585577, 1335...
$ racepctblack          <dbl> 1.30, 28.71, 7.95, 1.85, 1.70, 39.86, 2.55, 0.19, 7...
$ racePctWhite          <dbl> 74.02, 52.26, 82.42, 62.03, 91.28, 53.52, 97.01, 97...
$ racePctAsian          <dbl> 14.14, 7.00, 1.36, 0.19, 5.46, 2.74, 0.28, 1.69, 5....
$ racePctHisp           <dbl> 20.96, 24.36, 19.55, 91.07, 5.40, 5.63, 0.36, 2.54,...
$ agePct12t21           <dbl> 12.04, 13.06, 14.06, 18.80, 11.38, 13.92, 13.91, 19...
$ agePct12t29           <dbl> 26.68, 27.46, 30.74, 30.03, 22.12, 28.02, 21.82, 28...
$ agePct16t24           <dbl> 12.37, 13.09, 14.58, 14.87, 10.53, 14.12, 10.96, 11...
$ agePct65up            <dbl> 11.54, 11.62, 4.44, 10.44, 17.88, 13.74, 14.84, 3.1...
$ NumUnderPov           <int> 25484, 1384994, 973, 4767, 2105, 313374, 1116, 3141...
$ PctPopUnderPov        <dbl> 14.37, 19.29, 5.34, 37.04, 6.60, 20.27, 8.79, 4.21,...
$ PctSameHouse85        <dbl> 37.69, 63.06, 45.88, 69.40, 52.74, 64.33, 63.34, 51...
$ PctSameCity85         <dbl> 73.29, 82.76, 59.47, 95.85, 82.67, 88.86, 91.08, 80...
$ PctSameState85        <dbl> 76.85, 89.78, 94.53, 99.90, 85.41, 92.65, 95.84, 85...
```

```
> #-------------------------------------------------------------
> # LEVERAGES: There are 150 observations based on thumb-rule
> #            leverage > 2p/n. They have been isolated above.
> #            We need to investigate further based on Cooke's
> #            Distance to identify the outliers.
> #-------------------------------------------------------------
# ----------------------------------------------------------------
# Checking for Influential Observations based on Cooks Distance
# ----------------------------------------------------------------

> #Calculating F-Value Threshold

> num_df <- p
> den_df <- n-p

> F_thresh <- qf(0.5, num_df,den_df)
> F_thresh
[1] 0.9855624

> #calculate Cook's distances

> comm_cd <- cooks.distance(community_mdl_s8_AIC)

> #Identifying the outliers

> community_outliers <- which(comm_cd > F_thresh)
> community_outliers
18
> #-------------------------------------------------------------
> # OUTLIER: Based on Cooke's Distance, I found one observation
> #          as the outlier. Next step will be to assess the
> #          outlier impact by comparing the model summaries
> #          with and without it.
> #-------------------------------------------------------------
> #To check the influence of outlier, recreating the data-set minus the outlier

> community_ds2 <- community_ds[-community_outliers,]

> #Generating the model without the outlier
> community_mdl_y <- lm(final_formula, community_ds2)

> # Comparing the model summaries to check the impact of outliers.
> summary(community_mdl_s8_AIC)

Call:
lm(formula = final_formula)

Residuals:
     Min       1Q   Median       3Q      Max
-1639.76  -188.66   -38.47   128.55  2247.59

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          3.035e+03  5.146e+02   5.898 4.32e-09 ***
PctForeignBorn       1.109e+01  3.838e+00   2.889 0.003907 **
PctKids2Par         -1.217e+01  3.523e+00  -3.455 0.000563 ***
PctWorkMom          -7.996e+00  1.955e+00  -4.091 4.47e-05 ***
NumKidsBornNeverMar -5.020e-03  2.923e-03  -1.718 0.086030 .
```

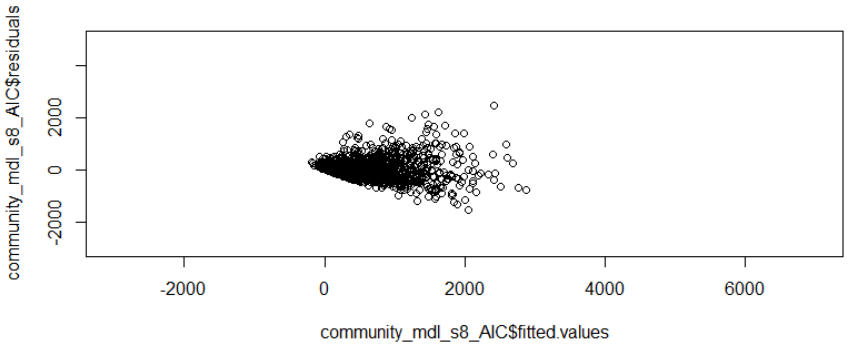
```
PctKidsBornNeverMar    3.658e+01  8.419e+00   4.344 1.47e-05 ***
numbUrban             -6.307e-04  3.184e-04  -1.980 0.047798 *
pctUrban               1.147e+00  2.566e-01   4.470 8.26e-06 ***
PopDens               -1.295e-02  4.474e-03  -2.894 0.003845 **
LemasPctOfficDrugUn    9.398e+00  3.146e+00   2.987 0.002853 **
PctLess9thGrade       -2.035e+01  5.286e+00  -3.850 0.000122 ***
PctNotHSGrad           9.323e+00  3.976e+00   2.345 0.019133 *
PctEmploy              1.226e+01  3.430e+00   3.574 0.000360 ***
PctEmplManu           -3.314e+00  1.294e+00  -2.560 0.010530 *
NumInShelters          9.755e-02  4.364e-02   2.235 0.025519 *
PctLargHouseOccup     -1.894e+01  8.784e+00  -2.156 0.031238 *
PersPerOwnOccHous      4.824e+02  1.397e+02   3.453 0.000566 ***
PersPerRentOccHous    -3.326e+02  9.983e+01  -3.331 0.000881 ***
PctPersOwnOccup       -5.341e+01  1.231e+01  -4.340 1.50e-05 ***
PctPersDenseHous       2.308e+01  5.151e+00   4.480 7.90e-06 ***
HousVacant             2.130e-02  5.029e-03   4.236 2.38e-05 ***
PctHousOwnOcc          4.970e+01  1.188e+01   4.183 3.01e-05 ***
PctVacantBoarded       1.345e+01  3.393e+00   3.964 7.64e-05 ***
PctVacMore6Mos        -2.072e+00  8.037e-01  -2.578 0.010003 *
OwnOccHiQuart         -4.128e-04  2.321e-04  -1.778 0.075479 .
NumImmig               1.079e-03  5.713e-04   1.889 0.059095 .
PctImmigRecent         1.696e+00  1.132e+00   1.498 0.134197
PctRecImmig5          -2.737e+01  1.201e+01  -2.280 0.022733 *
medIncome             -1.159e-02  6.135e-03  -1.889 0.059092 .
pctWWage              -1.622e+01  4.124e+00  -3.934 8.65e-05 ***
pctWFarmSelf           2.915e+01  1.472e+01   1.980 0.047806 *
pctWInvInc            -4.702e+00  2.243e+00  -2.096 0.036228 *
pctWRetire            -1.128e+01  3.155e+00  -3.576 0.000357 ***
medFamInc              8.749e-03  5.306e-03   1.649 0.099336 .
AsianPerCap            1.997e-03  9.424e-04   2.119 0.034244 *
MalePctDivorce         5.701e+01  1.556e+01   3.664 0.000255 ***
MalePctNevMarr         1.140e+01  3.432e+00   3.321 0.000913 ***
TotalPctDiv           -4.469e+01  1.651e+01  -2.707 0.006853 **
RentLowQ              -6.641e-01  2.150e-01  -3.090 0.002032 **
MedRent                6.901e-01  2.143e-01   3.221 0.001301 **
MedOwnCostPctIncNoMtg -3.405e+01  7.578e+00  -4.494 7.41e-06 ***
racepctblack           8.445e+00  1.261e+00   6.699 2.74e-11 ***
agePct12t21            8.812e+00  6.196e+00   1.422 0.155099
agePct12t29           -2.267e+01  6.882e+00  -3.294 0.001007 **
PctPopUnderPov        -5.878e+00  3.383e+00  -1.738 0.082454 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 356.5 on 1949 degrees of freedom
Multiple R-squared:  0.6711,   Adjusted R-squared:  0.6637
F-statistic:  90.4 on 44 and 1949 DF,  p-value: < 2.2e-16

> summary(community_mdl_y)

Call:
lm(formula = final_formula, data = community_ds2)

Residuals:
    Min      1Q  Median      3Q     Max
-1612.8  -187.9   -40.2   127.0  2296.3

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           3.082e+03  5.135e+02   6.002 2.32e-09 ***
PctForeignBorn        1.160e+01  3.831e+00   3.027 0.002505 **
PctKids2Par           -1.215e+01  3.514e+00  -3.456 0.000559 ***
PctWorkMom            -7.643e+00  1.952e+00  -3.914 9.38e-05 ***
NumKidsBornNeverMar   -2.618e-03  3.003e-03  -0.872 0.383466
```

```
PctKidsBornNeverMar    3.279e+01  8.474e+00   3.869 0.000113 ***
numbUrban             -6.361e-04  3.176e-04  -2.003 0.045341 *
pctUrban               1.174e+00  2.560e-01   4.585 4.84e-06 ***
PopDens               -1.304e-02  4.463e-03  -2.921 0.003530 **
LemasPctOfficDrugUn    9.238e+00  3.139e+00   2.943 0.003286 **
PctLess9thGrade       -2.032e+01  5.272e+00  -3.855 0.000120 ***
PctNotHSGrad           9.162e+00  3.966e+00   2.310 0.020983 *
PctEmploy              1.165e+01  3.426e+00   3.400 0.000687 ***
PctEmplManu           -3.205e+00  1.291e+00  -2.482 0.013156 *
NumInShelters          2.124e-01  5.558e-02   3.822 0.000137 ***
PctLargHouseOccup     -1.828e+01  8.764e+00  -2.085 0.037173 *
PersPerOwnOccHous      4.769e+02  1.394e+02   3.422 0.000634 ***
PersPerRentOccHous    -3.417e+02  9.961e+01  -3.431 0.000614 ***
PctPersOwnOccup       -5.475e+01  1.228e+01  -4.457 8.76e-06 ***
PctPersDenseHous       2.366e+01  5.141e+00   4.602 4.45e-06 ***
HousVacant             1.590e-02  5.273e-03   3.016 0.002597 **
PctHousOwnOcc          5.073e+01  1.185e+01   4.279 1.97e-05 ***
PctVacantBoarded       1.269e+01  3.392e+00   3.742 0.000188 ***
PctVacMore6Mos        -1.943e+00  8.025e-01  -2.421 0.015569 *
OwnOccHiQuart         -4.708e-04  2.322e-04  -2.028 0.042726 *
NumImmig               1.044e-03  5.699e-04   1.831 0.067191 .
PctImmigRecent         1.731e+00  1.129e+00   1.533 0.125457
PctRecImmig5          -3.005e+01  1.200e+01  -2.504 0.012377 *
medIncome             -1.211e-02  6.122e-03  -1.979 0.047985 *
pctWWage              -1.608e+01  4.114e+00  -3.908 9.63e-05 ***
pctWFarmSelf           2.812e+01  1.469e+01   1.915 0.055681 .
pctWInvInc            -4.999e+00  2.239e+00  -2.232 0.025720 *
pctWRetire            -1.150e+01  3.148e+00  -3.654 0.000265 ***
medFamInc              9.619e-03  5.299e-03   1.815 0.069643 .
AsianPerCap            2.027e-03  9.400e-04   2.157 0.031162 *
MalePctDivorce         5.506e+01  1.553e+01   3.545 0.000402 ***
MalePctNevMarr         1.039e+01  3.436e+00   3.024 0.002531 **
TotalPctDiv           -4.315e+01  1.647e+01  -2.619 0.008883 **
RentLowQ              -6.363e-01  2.146e-01  -2.966 0.003057 **
MedRent                6.980e-01  2.137e-01   3.266 0.001111 **
MedOwnCostPctIncNoMtg -3.246e+01  7.574e+00  -4.286 1.91e-05 ***
racepctblack           8.645e+00  1.259e+00   6.867 8.76e-12 ***
agePct12t21            8.929e+00  6.180e+00   1.445 0.148685
agePct12t29           -2.200e+01  6.867e+00  -3.204 0.001378 **
PctPopUnderPov        -5.685e+00  3.375e+00  -1.684 0.092254 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 355.6 on 1948 degrees of freedom
Multiple R-squared:  0.672,    Adjusted R-squared:  0.6646
F-statistic: 90.71 on 44 and 1948 DF,  p-value: < 2.2e-16

> #-------------------------------------------------------------
> # CONCLUSION: Removal of Outlier has had no significant impact
> #             on the Model hence outlier will not be removed.
> #-------------------------------------------------------------
```

7. Model Transformation

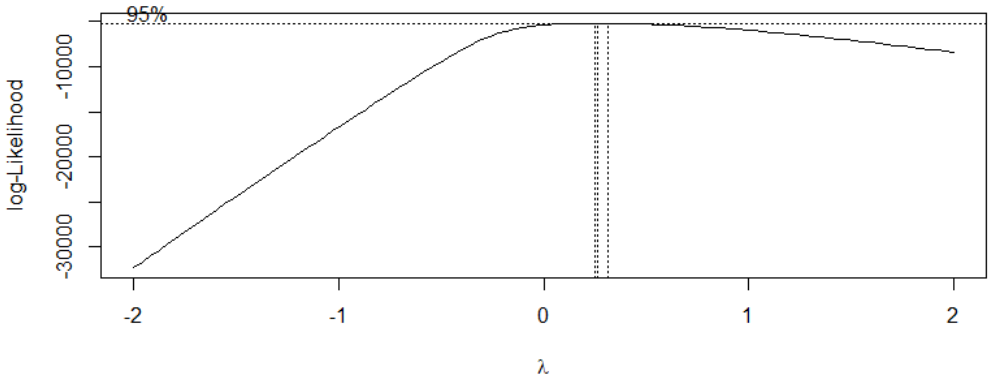
> *Apply the Transformations to the best fit model if necessary.*

```
> plot(community_mdl_s8_AIC$fitted.values, community_mdl_s8_AIC$residuals, xlim = c(-3000
, 7000), ylim = c(-3000, 5000))
```

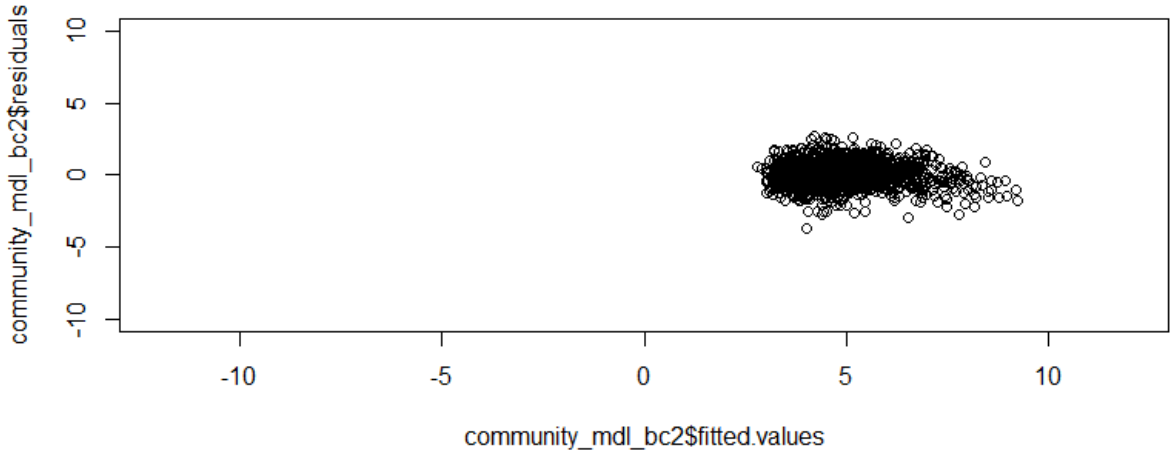



```
> #Try Box-Cox to remove heteroscedasticity
> bc <- boxcox(community_mdl_s8_AIC)
Error in boxcox.default(community_mdl_s8_AIC) : response variable must be positive

> # ------------------------------------------------
> # Varifying Response Variable. If it contains 0
> # then adding an offset to response variable
> # ------------------------------------------------

> summary(ViolentCrimesPerPop)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0   161.8   375.6   586.6   797.2  4877.1

> community_mdl_bc1 <- lm(ViolentCrimesPerPop + 0.01 ~ racepctblack + PctPersDenseHous +
pctUrban +   PctKidsBornNeverMar + HousVacant + pctWWage + MalePctDivorce + pctWRetire )

> summary(community_mdl_bc1)

Call:
lm(formula = ViolentCrimesPerPop + 0.01 ~ racepctblack + PctPersDenseHous +
    pctUrban + PctKidsBornNeverMar + HousVacant + pctWWage +
    MalePctDivorce + pctWRetire)

Residuals:
     Min       1Q   Median       3Q      Max
-1564.71  -189.12   -36.95   120.44  2449.56

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        725.759454 175.840651   4.127 3.86e-05 ***
racepctblack         8.619682   1.195490   7.210 8.62e-13 ***
PctPersDenseHous    20.354836   1.922747  10.586  < 2e-16 ***
pctUrban             1.387255   0.225801   6.144 1.02e-09 ***
PctKidsBornNeverMar 70.231731   6.320571  11.112  < 2e-16 ***
HousVacant           0.008299   0.001520   5.461 5.49e-08 ***
pctWWage           -11.041426   1.714942  -6.438 1.60e-10 ***
MalePctDivorce      40.123894   3.866355  10.378  < 2e-16 ***
pctWRetire          -9.227563   2.739564  -3.368 0.000775 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 364.2 on 1586 degrees of freedom
Multiple R-squared:  0.6462,   Adjusted R-squared:  0.6444
F-statistic: 362.1 on 8 and 1586 DF,  p-value: < 2.2e-16

> # ------------------------------------------------------------------------
> # Re-calculating the lambda
> # ------------------------------------------------------------------------

> bc <- boxcox(community_mdl_bc1)
```

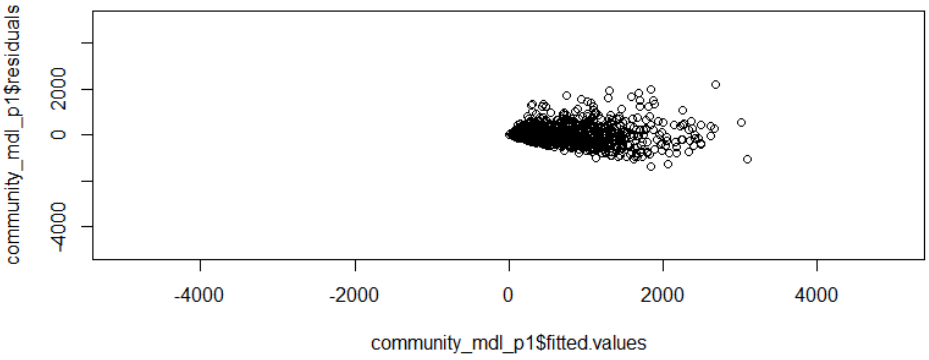



```
> lambda <- bc$x[which.max(bc$y)]

> lambda
[1] 0.3030303

> community_mdl_bc2 <- lm(((ViolentCrimesPerPop + 0.01)^lambda) ~ racepctblack + PctPersDenseHous + pctUrban + PctKidsBornNeverMar + HousVacant + pctWWage + MalePctDivorce + pctWRetire)

> plot(community_mdl_bc2$fitted.values, community_mdl_bc2$residuals, xlim = c(-12, 12), ylim = c(-10, 10))
```




```
> bptest(community_mdl_s8_AIC)

        studentized Breusch-Pagan test

data:  community_mdl_s8_AIC
```

```
BP = 278.76, df = 8, p-value < 2.2e-16

> bptest(community_mdl_bc1)

        studentized Breusch-Pagan test

data:  community_mdl_bc1
BP = 278.76, df = 8, p-value < 2.2e-16

> bptest(community_mdl_bc2)

        studentized Breusch-Pagan test

data:  community_mdl_bc2
BP = 36.757, df = 8, p-value = 1.276e-05

> # ------------------------------------------------------------------------
> # Applying Polynomial Transformation
> # ------------------------------------------------------------------------

> community_mdl_p1 <- lm(ViolentCrimesPerPop ~ polym(racepctblack, PctPersDenseHous, pctU
rban, PctKidsBornNeverMar, HousVacant, pctWWage, MalePctDivorce, pctWRetire, degree = 2),
community_train_ds)

> plot(community_mdl_p1$fitted.values, community_mdl_p1$residuals, xlim = c(-5000, 5000),
+      ylim = c(-5000, 5000))
```



After studying all 3 models: OLS, BoxCox Transformed and Polynomial results and their residual vs fitted plots, CoxBox Transformation based plot seem to provide the best fitting linear model.

8. Predict,

> *Report Inferences and make predictions.*

Parameter estimates and p-values for your final model.

Sr No	Predictor Variable	Estimate	p-Value
0	(Intercept)	5.08E+00	2E-16
1	racepctblack	1.98E-02	4.80E-14
2	PctPersDenseHous	6.33E-02	<2E-16
3	pctUrban	3.28E-03	9.02E-11
4	PctKidsBornNeverMar	9.40E-02	1.26E-11
5	HousVacant	1.22E-05	0.000301
6	pctWWage	-2.98E-02	2.30E-14
7	MalePctDivorce	1.51E-01	<2E-16
8	pctWRetire	-1.80E-02	0.003325

R^2 Coefficients for the model.

	Multiple R-squared:	0.6507	
	Adjusted R-squared:	0.6487	

```
> # ------------------------------------------------------------------
> # Week 7 Assignment Q1 & Q2:
> # ------------------------------------------------------------------
> a1 <- summary(community_mdl_bc2)
> a1

Call:
lm(formula = ((ViolentCrimesPerPop + 0.01)^lambda) ~ racepctblack +
    PctPersDenseHous + pctUrban + PctKidsBornNeverMar + HousVacant +
    pctWWage + MalePctDivorce + pctWRetire)

Residuals:
    Min      1Q  Median      3Q     Max
-4.7189 -0.7870 -0.0457  0.7369  5.4946

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        6.586e+00  5.619e-01  11.720  < 2e-16 ***
racepctblack       2.976e-02  3.820e-03   7.791 1.20e-14 ***
PctPersDenseHous   9.131e-02  6.144e-03  14.861  < 2e-16 ***
pctUrban           4.784e-03  7.216e-04   6.630 4.59e-11 ***
PctKidsBornNeverMar 1.427e-01  2.020e-02   7.066 2.37e-12 ***
HousVacant         1.879e-05  4.857e-06   3.868 0.000114 ***
pctWWage          -4.288e-02  5.480e-03  -7.825 9.20e-15 ***
MalePctDivorce     2.134e-01  1.236e-02  17.275  < 2e-16 ***
pctWRetire        -2.972e-02  8.755e-03  -3.394 0.000705 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.164 on 1586 degrees of freedom
Multiple R-squared:  0.6587,   Adjusted R-squared:  0.657
F-statistic: 382.7 on 8 and 1586 DF,  p-value: < 2.2e-16
```

95% confidence interval for the slope of most important predictor variable: PctPersDenseHou

```
> # ------------------------------------------------------------------
> # Week 7 Assignment Q3: Most Important Variable: PctPersDenseHou
> # ------------------------------------------------------------------

> a1$coefficients[3,]
    Estimate   Std. Error      t value     Pr(>|t|)
9.131342e-02 6.144488e-03 1.486103e+01 7.200200e-47

> v_coeff  <- a1$coefficients[3,1]
> v_stderr <- a1$coefficients[3,2]
> n <- dim(community_ds)[1]
> p <- length(community_mdl_bc2$coefficients)
> t_val <- qt(0.975, (n-p))

> v_coeff + (c(-1, 1) * t_val * v_stderr)
[1] 0.07926384 0.10336301
```

Compute and report a 95% confidence interval for a prediction. In other words, choose particular values of your predictors that are meaningful (say, perhaps the median of each) and compute a 95% confidence interval for the predicted value of y at those values.

```
> community_ds_median <- data.frame(
+   racepctblack = median(community_test_ds$racepctblack),
+   PctPersDenseHous = median(community_test_ds$PctPersDenseHous),
+   pctUrban = median(community_test_ds$pctUrban),
+   PctKidsBornNeverMar = median(community_test_ds$PctKidsBornNeverMar),
+   HousVacant = median(community_test_ds$HousVacant),
+   pctWWage = median(community_test_ds$pctWWage),
+   MalePctDivorce = median(community_test_ds$MalePctDivorce),
+   pctWRetire = median(community_test_ds$pctWRetire))

> predict(community_mdl_bc2, newdata = community_ds_median, interval = 'confidence')
       fit      lwr      upr
1 5.820758 5.734793 5.906722
```

Compute and report a 95% prediction interval for a particular observation (100th).

```
> # ------------------------------------------------------------------
> # Week 7 Assignment Q5: Compute and report a 95% prediction interval
> # for a particular observation. Again, you'll choose particular values
> # of your predictors and compute prediction interval for those values.
> # ------------------------------------------------------------------

> community_ds_100 <- community_test_ds[100,]
> predict(community_mdl_bc2, newdata = community_ds_100, interval = 'confidence')
         fit      lwr      upr
624 5.634868 5.518743 5.750994
```

9. Conclusion

- **Best fitting model was produced based on iterative approach by going through several cycles of predictor variable selection, EDA, Model build and diagnostics.**
- **It also required the subject matter knowledge to evaluate the findings of automated results from several processes during the modelling exercise.**
- **Produced model was based on 8 variables with Adjusted R^2 value = 0.64.**

10. Appendix - Code

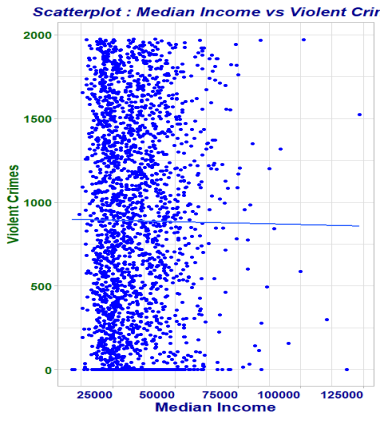
```
> # ------------------------------------------------------------------------
> # Initial Configuration
> # ------------------------------------------------------------------------
> library(ggplot2)
> library(tidyverse)
> library(okcupiddata)
> library(faraway)
> library(lubridate)
> library(stringr)
> library(NHANES)
> library(mdsr)
> library(rpart)
> library(partykit)
> library(randomForest)
> library(class)
> library(MASS)
> library(rms)
> library(naniar)
> library(lmtest)
> set.seed(1847)

> # ------------------------------------------------------------------------
> # Reading the Dataset and Metadata Files
> # ------------------------------------------------------------------------

> setwd("C:/Users/cool_/Google Drive/Education/Ashish/MCAS-UND/Semester2/Datasets")

> # Reading Community Violence Dataset
> file_ds <- file.path( "CommunityViolenceDataset.csv")
> community_all <- read.csv(file_ds, header = TRUE, sep = ",", na.strings = c("NA", "?"))

#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
# STAGE 4 : EDA
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------

> community_all$ViolentCrimesPerPop <- as.double(community_all$ViolentCrimesPerPop)
> community_all$NumKindsDrugsSeiz   <- as.integer(community_all$NumKindsDrugsSeiz)

> community_ds <- community_all

> attach(community_ds)

> glimpse(community_ds)

Observations: 2,215
Variables: 147
$ communityname         <fct> BerkeleyHeightstownship, Marpletownship, Tigardc...
$ state                 <fct> NJ, PA, OR, NY, MN, MO, MA, IN, ND, TX, TX, CA, ...
$ countyCode            <int> 39, 45, NA, 35, 7, NA, 21, NA, 17, NA, NA, NA, N...
$ communityCode         <int> 5320, 47616, NA, 29443, 5068, NA, 50250, NA, 257...
$ fold                  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
$ population            <int> 11980, 23123, 29344, 16656, 11245, 140494, 28700...
$ householdsize         <dbl> 3.10, 2.82, 2.43, 2.40, 2.76, 2.45, 2.60, 2.45, ...
$ racepctblack          <dbl> 1.37, 0.80, 0.74, 1.70, 0.53, 2.51, 1.60, 14.20,...
$ racePctWhite          <dbl> 91.78, 95.57, 94.33, 97.35, 89.16, 95.65, 96.57,...
$ racePctAsian          <dbl> 6.50, 3.44, 3.43, 0.50, 1.17, 0.90, 1.47, 0.40, ...
```

```
$ racePctHisp           <dbl> 1.88, 0.85, 2.35, 0.70, 0.52, 0.95, 1.10, 0.63, ...
$ agePct12t21           <dbl> 12.47, 11.01, 11.36, 12.55, 24.46, 18.09, 11.17,...
$ agePct12t29           <dbl> 21.44, 21.30, 25.88, 25.20, 40.53, 32.89, 27.41,...
$ agePct16t24           <dbl> 10.93, 10.48, 11.01, 12.19, 28.69, 20.04, 12.76,...
$ agePct65up            <dbl> 11.33, 17.18, 10.28, 17.57, 12.65, 13.26, 14.42,...
$ numbUrban             <int> 11980, 23123, 29344, 0, 0, 140494, 28700, 59449,...
$ pctUrban              <dbl> 100.00, 100.00, 100.00, 0.00, 0.00, 100.00, 100....
$ medIncome             <int> 75122, 47917, 35669, 20580, 17390, 21577, 42805,...
$ pctWWage              <dbl> 89.24, 78.99, 82.00, 68.15, 69.33, 75.78, 79.47,...
$ pctWFarmSelf          <dbl> 1.55, 1.11, 1.15, 0.24, 0.55, 1.00, 0.39, 0.67, ...
$ pctWInvInc            <dbl> 70.20, 64.11, 55.73, 38.95, 42.82, 41.15, 47.70,...
$ pctWSocSec            <dbl> 23.62, 35.50, 22.25, 39.48, 32.16, 29.31, 30.23,...
$ pctWPubAsst           <dbl> 1.03, 2.75, 2.94, 11.71, 11.21, 7.12, 5.41, 8.81...
$ pctWRetire            <dbl> 18.39, 22.85, 14.56, 18.33, 14.43, 14.09, 17.23,...
$ medFamInc             <int> 79584, 55323, 42112, 26501, 24018, 27705, 50394,...
$ perCapInc             <int> 29711, 20148, 16946, 10810, 8483, 11878, 18193, ...
$ whitePerCap           <int> 30233, 20191, 17103, 10909, 9009, 12029, 18276, ...
$ blackPerCap           <int> 13600, 18137, 16644, 9984, 887, 7382, 17342, 982...
$ indianPerCap          <int> 5725, 0, 21606, 4941, 4425, 10264, 21482, 6634, ...
$ AsianPerCap           <int> 27101, 20074, 15528, 3541, 3352, 10753, 12639, 8...
$ OtherPerCap           <int> 5115, 5250, 5954, 2451, 3000, 7192, 21852, 7428,...
$ HispPerCap            <int> 22838, 12222, 8405, 4391, 1328, 8104, 22594, 618...
$ NumUnderPov           <int> 227, 885, 1389, 2831, 2855, 23223, 1126, 10320, ...
$ PctPopUnderPov        <dbl> 1.96, 3.98, 4.75, 17.23, 29.99, 17.78, 4.01, 17....
$ PctLess9thGrade       <dbl> 5.81, 5.61, 2.80, 11.05, 12.15, 8.76, 4.49, 10.0...
$ PctNotHSGrad          <dbl> 9.90, 13.72, 9.09, 33.68, 23.06, 23.03, 13.89, 2...
$ PctBSorMore           <dbl> 48.18, 29.89, 30.13, 10.81, 25.28, 20.66, 27.01,...
$ PctUnemployed         <dbl> 2.70, 2.43, 4.01, 9.86, 9.08, 5.72, 4.85, 8.19, ...
$ PctEmploy             <dbl> 64.55, 61.96, 69.80, 54.74, 52.44, 59.02, 65.42,...
$ PctEmplManu           <dbl> 14.65, 12.26, 15.95, 31.22, 6.89, 14.31, 14.02, ...
$ PctEmplProfServ       <dbl> 28.82, 29.28, 21.52, 27.43, 36.54, 26.83, 27.17,...
$ PctOccupManu          <dbl> 5.49, 6.39, 8.79, 26.76, 10.94, 14.72, 8.50, 21....
$ PctOccupMgmtProf      <dbl> 50.73, 37.64, 32.48, 22.71, 27.80, 23.42, 32.78,...
$ MalePctDivorce        <dbl> 3.67, 4.23, 10.10, 10.98, 7.51, 11.40, 5.97, 13....
$ MalePctNevMarr        <dbl> 26.38, 27.99, 25.78, 28.15, 50.66, 33.32, 36.05,...
$ FemalePctDiv          <dbl> 5.22, 6.45, 14.76, 14.47, 11.64, 14.46, 9.06, 16...
$ TotalPctDiv           <dbl> 4.47, 5.42, 12.55, 12.91, 9.73, 13.04, 7.64, 14....
$ PersPerFam            <dbl> 3.22, 3.11, 2.95, 2.98, 2.98, 2.89, 3.14, 2.95, ...
$ PctFam2Par            <dbl> 91.43, 86.91, 78.54, 64.02, 58.59, 71.94, 79.53,...
$ PctKids2Par           <dbl> 90.17, 85.33, 78.85, 62.36, 55.20, 69.79, 79.76,...
$ PctYoungKids2Par      <dbl> 95.78, 96.82, 92.37, 65.38, 66.51, 79.76, 92.05,...
$ PctTeen2Par           <dbl> 95.81, 86.46, 75.72, 67.43, 79.17, 75.33, 77.12,...
$ PctWorkMomYoungKids   <dbl> 44.56, 51.14, 66.08, 59.59, 61.22, 62.96, 65.16,...
$ PctWorkMom            <dbl> 58.88, 62.43, 74.19, 70.27, 68.94, 70.52, 72.81,...
$ NumKidsBornNeverMar   <int> 31, 43, 164, 561, 402, 1511, 263, 2368, 751, 353...
$ PctKidsBornNeverMar   <dbl> 0.36, 0.24, 0.88, 3.84, 4.70, 1.58, 1.18, 4.66, ...
$ NumImmig              <int> 1277, 1920, 1468, 339, 196, 2091, 2637, 517, 147...
$ PctImmigRecent        <dbl> 8.69, 5.21, 16.42, 13.86, 46.94, 21.33, 11.38, 1...
$ PctImmigRec5          <dbl> 13.00, 8.65, 23.98, 13.86, 56.12, 30.56, 16.27, ...
$ PctImmigRec8          <dbl> 20.99, 13.33, 32.08, 15.34, 67.86, 38.02, 23.93,...
$ PctImmigRec10         <dbl> 30.93, 22.50, 35.63, 15.34, 69.90, 45.48, 27.76,...
$ PctRecentImmig        <dbl> 0.93, 0.43, 0.82, 0.28, 0.82, 0.32, 1.05, 0.11, ...
$ PctRecImmig5          <dbl> 1.39, 0.72, 1.20, 0.28, 0.98, 0.45, 1.49, 0.20, ...
$ PctRecImmig8          <dbl> 2.24, 1.11, 1.61, 0.31, 1.18, 0.57, 2.20, 0.25, ...
$ PctRecImmig10         <dbl> 3.30, 1.87, 1.78, 0.31, 1.22, 0.68, 2.55, 0.29, ...
$ PctSpeakEnglOnly      <dbl> 85.68, 87.79, 93.11, 94.98, 94.64, 96.87, 89.98,...
$ PctNotSpeakEnglWell   <dbl> 1.37, 1.81, 1.14, 0.56, 0.39, 0.60, 0.60, 0.28, ...
$ PctLargHouseFam       <dbl> 4.81, 4.25, 2.97, 3.93, 5.23, 3.08, 5.08, 3.85, ...
$ PctLargHouseOccup     <dbl> 4.17, 3.34, 2.05, 2.56, 3.11, 1.92, 3.46, 2.55, ...
$ PersPerOccupHous      <dbl> 2.99, 2.70, 2.42, 2.37, 2.35, 2.28, 2.55, 2.36, ...
$ PersPerOwnOccHous     <dbl> 3.00, 2.83, 2.69, 2.51, 2.55, 2.37, 2.89, 2.42, ...
$ PersPerRentOccHous    <dbl> 2.84, 1.96, 2.06, 2.20, 2.12, 2.16, 2.09, 2.27, ...
$ PctPersOwnOccup       <dbl> 91.46, 89.03, 64.18, 58.18, 58.13, 57.81, 64.62,...
```

```
$ PctPersDenseHous      <dbl> 0.39, 1.01, 2.03, 1.21, 2.94, 2.11, 1.47, 1.90, ...
$ PctHousLess3BR        <dbl> 11.06, 23.60, 47.46, 45.66, 55.64, 53.19, 47.35,...
$ MedNumBR              <int> 3, 3, 3, 3, 2, 2, 3, 2, 2, 2, 2, 2, 2, 3, 3, 3, ...
$ HousVacant            <int> 64, 240, 544, 669, 333, 5119, 566, 2051, 1562, 5...
$ PctHousOccup          <dbl> 98.37, 97.15, 95.68, 91.19, 92.45, 91.81, 95.11,...
$ PctHousOwnOcc         <dbl> 91.01, 84.88, 57.79, 54.89, 53.57, 55.50, 56.96,...
$ PctVacantBoarded      <dbl> 3.12, 0.00, 0.92, 2.54, 3.90, 2.09, 1.41, 6.39, ...
$ PctVacMore6Mos        <dbl> 37.50, 18.33, 7.54, 57.85, 42.64, 26.22, 34.45, ...
$ MedYrHousBuilt        <int> 1959, 1958, 1976, 1939, 1958, 1966, 1956, 1954, ...
$ PctHousNoPhone        <dbl> 0.00, 0.31, 1.55, 7.00, 7.45, 6.13, 0.69, 8.42, ...
$ PctWOFullPlumb        <dbl> 0.28, 0.14, 0.12, 0.87, 0.82, 0.31, 0.28, 0.49, ...
$ OwnOccLowQuart        <int> 215900, 136300, 74700, 36400, 30600, 37700, 1551...
$ OwnOccMedVal          <int> 262600, 164200, 90400, 49600, 43200, 53900, 1790...
$ OwnOccHiQuart         <int> 326900, 199900, 112000, 66500, 59500, 73100, 215...
$ OwnOccQrange          <int> 111000, 63600, 37300, 30100, 28900, 35400, 60400...
$ RentLowQ              <int> 685, 467, 370, 195, 202, 215, 463, 186, 241, 192...
$ RentMedian            <int> 1001, 560, 428, 250, 283, 280, 669, 253, 321, 28...
$ RentHighQ             <int> 1001, 672, 520, 309, 362, 349, 824, 325, 387, 36...
$ RentQrange            <int> 316, 205, 150, 114, 160, 134, 361, 139, 146, 177...
$ MedRent               <int> 1001, 627, 484, 333, 332, 340, 736, 338, 355, 35...
$ MedRentPctHousInc     <dbl> 23.8, 27.6, 24.1, 28.7, 32.2, 26.4, 24.4, 26.3, ...
$ MedOwnCostPctInc      <dbl> 21.1, 20.7, 21.7, 20.6, 23.2, 17.3, 20.8, 15.1, ...
$ MedOwnCostPctIncNoMtg <dbl> 14.0, 12.5, 11.6, 14.5, 12.9, 11.7, 12.5, 12.2, ...
$ NumInShelters         <int> 11, 0, 16, 0, 2, 327, 0, 21, 125, 43, 1, 20, 28,...
$ NumStreet             <int> 0, 0, 0, 0, 0, 4, 0, 0, 15, 4, 0, 49, 2, 0, 0, 0...
$ PctForeignBorn        <dbl> 10.66, 8.30, 5.00, 2.04, 1.74, 1.49, 9.19, 0.87,...
$ PctBornSameState      <dbl> 53.72, 77.17, 44.77, 88.71, 73.75, 64.35, 77.30,...
$ PctSameHouse85        <dbl> 65.29, 71.27, 36.60, 56.70, 42.22, 42.29, 63.45,...
$ PctSameCity85         <dbl> 78.09, 90.22, 61.26, 90.17, 60.34, 70.61, 82.23,...
$ PctSameState85        <dbl> 89.14, 96.12, 82.85, 96.24, 89.02, 85.66, 93.53,...
$ LemasSwornFT          <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, 198, NA, NA,...
$ LemasSwFTPerPop       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 183.53, NA, ...
$ LemasSwFTFieldOps     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, 187, NA, NA,...
$ LemasSwFTFieldPerPop  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 173.33, NA, ...
$ LemasTotalReq         <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, 73432, NA, N...
$ LemasTotReqPerPop     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 68065.1, NA,...
$ PolicReqPerOffic      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 370.9, NA, N...
$ PolicPerPop           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 183.5, NA, N...
$ RacialMatchCommPol    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 89.32, NA, N...
$ PctPolicWhite         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 78.28, NA, N...
$ PctPolicBlack         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 11.11, NA, N...
$ PctPolicHisp          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 10.61, NA, N...
$ PctPolicAsian         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.00, NA, NA...
$ PctPolicMinor         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 21.72, NA, N...
$ OfficAssgnDrugUnits   <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, 13, NA, NA, ...
$ NumKindsDrugsSeiz     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, 12, NA, NA, ...
$ PolicAveOTWorked      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 60.2, NA, NA...
$ LandArea              <dbl> 6.5, 10.6, 10.6, 5.2, 11.5, 70.4, 10.9, 39.2, 30...
$ PopDens               <dbl> 1845.9, 2186.7, 2780.9, 3217.7, 974.2, 1995.7, 2...
$ PctUsePubTrans        <dbl> 9.63, 3.84, 4.37, 3.31, 0.38, 0.97, 9.62, 0.70, ...
$ PolicCars             <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, 100, NA, NA,...
$ PolicOperBudg         <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, 9315474, NA,...
$ LemasPctPolicOnPatr   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 94.44, NA, N...
$ LemasGangUnitDeploy   <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, 10, NA, NA, ...
$ LemasPctOfficDrugUn   <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, ...
$ PolicBudgPerPop       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, 86346.3, NA,...
$ murders               <int> 0, 0, 3, 0, 0, 7, 0, 8, 0, 29, 1, 12, 3, 16, 0, ...
$ murdPerPop            <dbl> 0.00, 0.00, 8.30, 0.00, 0.00, 4.63, 0.00, 13.13,...
$ rapes                 <int> 0, 1, 6, 10, NA, 77, 4, 34, 35, 141, 29, 21, 36,...
$ rapesPerPop           <dbl> 0.00, 4.25, 16.60, 57.86, NA, 50.98, 13.53, 55.7...
$ robberies             <int> 1, 5, 56, 10, 4, 136, 9, 98, 16, 453, 71, 309, 5...
$ robbbPerPop           <dbl> 8.20, 21.26, 154.95, 57.86, 32.04, 90.05, 30.44,...
$ assaults              <int> 4, 24, 14, 33, 14, 449, 54, 128, 41, 1043, 131, ...
```

```
$ assaultPerPop          <dbl> 32.81, 102.05, 38.74, 190.93, 112.14, 297.29, 18...
$ burglaries             <int> 14, 57, 274, 225, 91, 2094, 110, 608, 425, 2397,...
$ burglPerPop            <dbl> 114.85, 242.37, 758.14, 1301.78, 728.93, 1386.46...
$ larcenies              <int> 138, 376, 1797, 716, 1060, 7690, 288, 2250, 3149...
$ larcPerPop             <dbl> 1132.08, 1598.78, 4972.19, 4142.56, 8490.87, 509...
$ autoTheft              <int> 16, 26, 136, 47, 91, 454, 144, 125, 206, 1070, 1...
$ autoTheftPerPop        <dbl> 131.26, 110.55, 376.30, 271.93, 728.93, 300.60, ...
$ arsons                 <int> 2, 1, 22, NA, 5, 134, 17, 9, 8, 18, 6, 20, 9, 46...
$ arsonsPerPop           <dbl> 16.41, 4.25, 60.87, NA, 40.05, 88.72, 57.50, 14....
$ ViolentCrimesPerPop    <dbl> 41.02, 127.56, 218.59, 306.64, NA, 442.95, 226.6...
$ nonViolPerPop          <dbl> 1394.59, 1955.95, 6167.51, NA, 9988.79, 6867.42,...
```

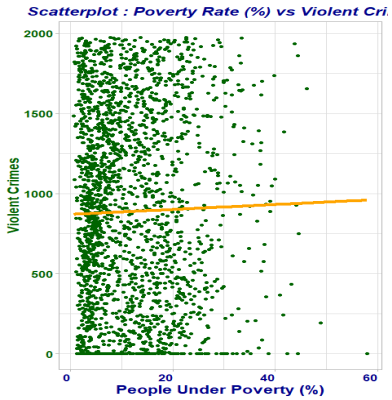
```
> # ------------------------------------------------------------------------
> # Plot 1 : Scatterplot : Median Income vs Violent Crime
> # ------------------------------------------------------------------------

> summary(community_ds$medIncome)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   8866   23817   31441   33985   41481  123625

> ggplot(data = community_ds, mapping = aes(x = medIncome, y = ViolentCrimesPerPop)) + ge
om_point(mapping = aes(color = "darkblue")) +
+   geom_point(color='blue') +   geom_smooth(method = "lm", se = FALSE)  +
+   ggtitle("Scatterplot : Median Income vs Violent Crime") + xlab("Median Income")  + yl
ab("Violent Crimes") +
+   scale_fill_brewer(palette = "Pastel1") +
+   theme_light() +
+   theme(plot.title   = element_text(color = "dark blue", size = 16, face = "bold.italic
", hjust = 0.5),
+         axis.title.x = element_text(color = "darkblue", size = 16, face = "bold"),
+         axis.title.y = element_text(color = "darkgreen", size = 14, face = "bold"),
+         axis.text.x  = element_text(color = "darkblue", size = 12, face = "bold", angle
= 0, hjust = 1),
+         axis.text.y  = element_text(color = "darkgreen", size = 12, face = "bold"),
+         legend.title = element_text(color = "blue", size = 12, face = "bold"),
+         legend.text  = element_text(color = "blue", size = 12, face = "bold"),
+         legend.background = element_rect(fill = "lightblue", size = 0.5, linetype = "so
lid"),
+         legend.position = "none",
+         strip.text.x = element_text(size = 12, colour = "yellow", face = "bold"))
```



As income grows, violent crime per population comes down.

```
> # ------------------------------------------------------------------------
> # Plot 2 : Scatterplot : People Under Poverty vs Violent Crime
> # ------------------------------------------------------------------------
```

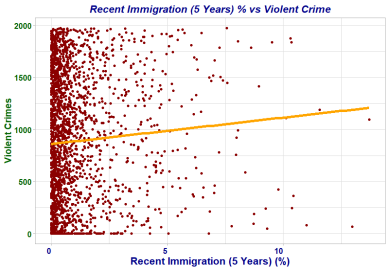
```
> summary(community_ds$PctPopUnderPov)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.64    4.51    9.33   11.62   16.91   58.00

> ggplot(data = community_ds, mapping = aes(x = PctPopUnderPov, y = ViolentCrimesPerPop))
+ geom_point(mapping = aes(color = "darkblue")) +
+   geom_point(color='dark green') +
+   geom_smooth(method = "lm", se = FALSE, colour="orange", size = 2)  +
+   ggtitle("Scatterplot : Poverty Rate (%) vs Violent Crime") + xlab("People Under Pover
ty (%)")  + ylab("Violent Crimes") +
+   #scale_x_continuous(breaks = seq(15,75,5), lim = c(15, 75)) +
+   #scale_y_continuous(breaks = seq(0, 72, 6), lim = c(0, 72)) +
+   scale_fill_brewer(palette = "Pastel1") +
+   theme_light() +
+   theme(plot.title   = element_text(color = "dark blue", size = 16, face = "bold.italic
", hjust = 0.5),
+         axis.title.x = element_text(color = "darkblue", size = 16, face = "bold"),
+         axis.title.y = element_text(color = "darkgreen", size = 14, face = "bold"),
+         axis.text.x  = element_text(color = "darkblue", size = 12, face = "bold", angle
= 0, hjust = 1),
+         axis.text.y  = element_text(color = "darkgreen", size = 12, face = "bold"),
+         legend.title = element_text(color = "blue", size = 12, face = "bold"),
+         legend.text  = element_text(color = "blue", size = 12, face = "bold"),
+         legend.background = element_rect(fill = "lightblue", size = 0.5, linetype = "so
lid"),
+         legend.position = "none",
+         strip.text.x = element_text(size = 12, colour = "yellow", face = "bold"))
```



As Poverty Rate grows, violent crime per population goes up.

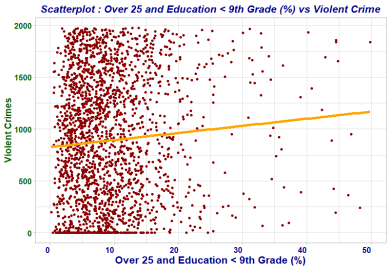
```
> # ------------------------------------------------------------------------
> # Plot 3 : Scatterplot : Recent Immigration (3 Years) % vs Violent Crime
> # ------------------------------------------------------------------------

> summary(community_ds$PctRecentImmig)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.170   0.500   1.099   1.310  13.710

> ggplot(data = community_ds, mapping = aes(x = PctRecentImmig, y = ViolentCrimesPerPop))
+ geom_point(mapping = aes(color = "darkblue")) +
+   geom_point(color='dark red') +
+   geom_smooth(method = "lm", se = FALSE, colour="orange", size = 2)  +
+   ggtitle("Recent Immigration (5 Years) % vs Violent Crime") + xlab("Recent Immigration
(5 Years) (%)")  + ylab("Violent Crimes") +
+   #scale_x_continuous(breaks = seq(15,75,5), lim = c(15, 75)) +
+   #scale_y_continuous(breaks = seq(0, 72, 6), lim = c(0, 72)) +
+   scale_fill_brewer(palette = "Pastel1") +
+   theme_light() +
```

```
+   theme(plot.title   = element_text(color = "dark blue", size = 16, face = "bold.italic
", hjust = 0.5),
+         axis.title.x = element_text(color = "darkblue", size = 16, face = "bold"),
+         axis.title.y = element_text(color = "darkgreen", size = 14, face = "bold"),
+         axis.text.x  = element_text(color = "darkblue", size = 12, face = "bold", angle
= 0, hjust = 1),
+         axis.text.y  = element_text(color = "darkgreen", size = 12, face = "bold"),
+         legend.title = element_text(color = "blue", size = 12, face = "bold"),
+         legend.text  = element_text(color = "blue", size = 12, face = "bold"),
+         legend.background = element_rect(fill = "lightblue", size = 0.5, linetype = "so
lid"),
+         legend.position = "none",
+         strip.text.x = element_text(size = 12, colour = "yellow", face = "bold"))
```



As Immigrated Count over Population grows, violent crime per population goes up.

```
> # ------------------------------------------------------------------------------
> # Plot 4 : Scatterplot : Over 25 and Education < 9th Grade (%) vs Violent Crime
> # ------------------------------------------------------------------------------

> summary(community_ds$PctLess9thGrade)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.200   4.640   7.740   9.187  11.835  49.890

> ggplot(data = community_ds, mapping = aes(x = PctLess9thGrade, y = ViolentCrimesPerPop)
) + geom_point(mapping = aes(color = "darkblue")) +
+   geom_point(color='dark red') +
+   geom_smooth(method = "lm", se = FALSE, colour="orange", size = 2)  +
+   ggtitle("Scatterplot : Over 25 and Education < 9th Grade (%) vs Violent Crime") + xla
b("Over 25 and Education < 9th Grade (%)")  + ylab("Violent Crimes") +
+   #scale_x_continuous(breaks = seq(15,75,5), lim = c(15, 75)) +
+   #scale_y_continuous(breaks = seq(0, 72, 6), lim = c(0, 72)) +
+   scale_fill_brewer(palette = "Pastel1") +
+   theme_light() +
+   theme(plot.title   = element_text(color = "dark blue", size = 16, face = "bold.italic
", hjust = 0.5),
+         axis.title.x = element_text(color = "darkblue", size = 16, face = "bold"),
+         axis.title.y = element_text(color = "darkgreen", size = 14, face = "bold"),
+         axis.text.x  = element_text(color = "darkblue", size = 12, face = "bold", angle
= 0, hjust = 1),
+         axis.text.y  = element_text(color = "darkgreen", size = 12, face = "bold"),
+         legend.title = element_text(color = "blue", size = 12, face = "bold"),
+         legend.text  = element_text(color = "blue", size = 12, face = "bold"),
+         legend.background = element_rect(fill = "lightblue", size = 0.5, linetype = "so
lid"),
+         legend.position = "none",
+         strip.text.x = element_text(size = 12, colour = "yellow", face = "bold"))
```



As Uneducated Rate grows, violent crime per population goes up.

```
> # ------------------------------------------------------------------------
> # Building first model based on selected 4 prediction Variables
> # ------------------------------------------------------------------------

> community_ds_s4 <- community_ds %>% dplyr::select (ViolentCrimesPerPop, medIncome, PctP
opUnderPov, PctLess9thGrade, PctRecentImmig) %>% na.omit()

> glimpse(community_ds_s4)

Observations: 1,994
Variables: 5
$ ViolentCrimesPerPop <dbl> 41.02, 127.56, 218.59, 306.64, 442.95, 226.63, 439...
$ medIncome           <int> 75122, 47917, 35669, 20580, 21577, 42805, 23221, 2...
$ PctPopUnderPov      <dbl> 1.96, 3.98, 4.75, 17.23, 17.78, 4.01, 17.98, 13.68...
$ PctLess9thGrade     <dbl> 5.81, 5.61, 2.80, 11.05, 8.76, 4.49, 10.09, 5.52, ...
$ PctRecentImmig      <dbl> 0.93, 0.43, 0.82, 0.28, 0.32, 1.05, 0.11, 0.47, 0....

> community_form <- as.formula(ViolentCrimesPerPop ~ medIncome + PctPopUnderPov + PctLess
9thGrade + PctRecentImmig)

> community_mdl_s4 <- lm(community_form, community_ds_s4)
> summary(community_mdl_s4)

Call:
lm(formula = community_form, data = community_ds_s4)

Residuals:
    Min      1Q  Median      3Q     Max
-1728.2  -255.2   -81.2   144.2  3901.4

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   329.390068  70.698436   4.659 3.39e-06 ***
medIncome      -0.004815   0.001420  -3.391 0.000709 ***
PctPopUnderPov 29.278363   2.347460  12.472  < 2e-16 ***
PctLess9thGrade  0.008532   2.380671   0.004 0.997141
PctRecentImmig  66.598040   7.823400   8.513  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 520.4 on 1989 degrees of freedom
Multiple R-squared:  0.285,    Adjusted R-squared:  0.2836
F-statistic: 198.2 on 4 and 1989 DF,  p-value: < 2.2e-16

#-------------------------------------------------------------------------
#-------------------------------------------------------------------------
#-------------------------------------------------------------------------
# STAGE 6: AUTOMATED MODEL SELECTION
#-------------------------------------------------------------------------
#-------------------------------------------------------------------------
```

```
#-------------------------------------------------------------------------------

# ---------------------------------------------------------------------
# Function for generating LM Formulae for any data-set
# The Dataset must contain first variable as response and rest
# as predictor variables
# ---------------------------------------------------------------------
> f_generate_formula <-function(arg_ds) {
+   ds_names <- names(arg_ds)
+   ncols <- length(ds_names)
+   ds_formula <- str_glue(ds_names[[1]], " ~ ")
+   for (i in 2:ncols) {
+     ds_formula <- str_glue(ds_formula, " " , ds_names[i] , " + ")
+   }
+   ds_formula <- str_sub(ds_formula, 1, str_length(ds_formula) - 2)
+   return(ds_formula)
+ }

> # ---------------------------------------------------------------------
> # Function for returning a list of variables from the argument
> # dataset which contain NAs
> # ---------------------------------------------------------------------

> f_get_NA_vars <-function(arg_ds) {
+   bad_cols <- c()
+   ds_names <- names(arg_ds)
+   ncols <- length(ds_names)
+   for (i in 2:ncols) {
       if (sum(is.na(arg_ds[[i]])) > 0 ) { bad_cols <- c(bad_cols, ds_names[[i]]) } }
+   return(bad_cols)
+ }

> # ---------------------------------------------------------------------
> # Creating the initial dataset with 1+124 Variables
> # ---------------------------------------------------------------------

> community_all %>% dplyr::select(ViolentCrimesPerPop, PctForeignBorn, PctBornSameState,
PctFam2Par, PctKids2Par, PctYoungKids2Par, PctTeen2Par, PctWorkMomYoungKids, PctWorkMom,
NumKidsBornNeverMar, PctKidsBornNeverMar, numbUrban, pctUrban, LandArea, PopDens, PctUseP
ubTrans, PolicCars, PolicOperBudg, LemasPctPolicOnPatr, LemasGangUnitDeploy, LemasPctOffi
cDrugUn, PolicBudgPerPop, PctLess9thGrade, PctNotHSGrad, PctBSorMore, PctUnemployed, PctE
mploy, PctEmplManu, PctEmplProfServ, PctOccupManu, PctOccupMgmtProf, PersPerFam, NumInShe
lters, NumStreet, householdsize, PctLargHouseFam, PctLargHouseOccup, PersPerOccupHous, Pe
rsPerOwnOccHous, PersPerRentOccHous, PctPersOwnOccup, PctPersDenseHous, PctHousLess3BR,
MedNumBR, HousVacant, PctHousOccup, PctHousOwnOcc, PctVacantBoarded, PctVacMore6Mos, MedY
rHousBuilt, PctHousNoPhone, PctWOFullPlumb, OwnOccLowQuart, OwnOccMedVal, OwnOccHiQuart,
OwnOccQrange, NumImmig, PctImmigRecent, PctImmigRec5, PctImmigRec8, PctImmigRec10, PctRec
entImmig, PctRecImmig5, PctRecImmig8, PctRecImmig10, medIncome, pctWWage, pctWFarmSelf, p
ctWInvInc, pctWSocSec, pctWPubAsst, pctWRetire, medFamInc, perCapInc, whitePerCap, blackP
erCap, indianPerCap, AsianPerCap, OtherPerCap, HispPerCap, PctSpeakEnglOnly, PctNotSpeakE
nglWell, MalePctDivorce, MalePctNevMarr, FemalePctDiv, TotalPctDiv, RentLowQ, RentMedian,
RentHighQ, RentQrange, MedRent, MedRentPctHousInc, MedOwnCostPctInc, edOwnCostPctIncNoMtg
, LemasSwornFT, LemasSwFTPerPop, LemasSwFTFieldOps, LemasSwFTFieldPerPop, LemasTotalReq,
LemasTotReqPerPop, PolicReqPerOffic, PolicPerPop, RacialMatchCommPol, PctPolicWhite, PctP
olicBlack, PctPolicHisp, PctPolicAsian, PctPolicMinor, OfficAssgnDrugUnits, NumKindsDrugs
Seiz, PolicAveOTWorked, population, racepctblack, racePctWhite, racePctAsian, racePctHisp
, agePct12t21, agePct12t29, agePct16t24, agePct65up, NumUnderPov, PctPopUnderPov, PctSame
House85, PctSameCity85, PctSameState85) -> community_ds_s124

> # ---------------------------------------------------------------------
> # Removing all variables which have either ? or NA
> # ---------------------------------------------------------------------
```

```
> bad_vars <- f_get_NA_vars(community_ds_s124)
> bad_vars
 [1] "PolicCars"             "PolicOperBudg"         "LemasPctPolicOnPatr"
 [4] "LemasGangUnitDeploy"   "PolicBudgPerPop"       "OtherPerCap"
 [7] "LemasSwornFT"          "LemasSwFTPerPop"       "LemasSwFTFieldOps"
[10] "LemasSwFTFieldPerPop"  "LemasTotalReq"         "LemasTotReqPerPop"
[13] "PolicReqPerOffic"      "PolicPerPop"           "RacialMatchCommPol"
[16] "PctPolicWhite"         "PctPolicBlack"         "PctPolicHisp"
[19] "PctPolicAsian"         "PctPolicMinor"         "OfficAssgnDrugUnits"
[22] "NumKindsDrugsSeiz"     "PolicAveOTWorked"

> # -------------- Removing all such bad variables and Rows with NAs.-------------------
> community_ds_s124 %>% dplyr::select(-bad_vars) %>% na.omit() -> community_ds_s124

> glimpse(community_ds_s124)
Observations: 1,994
Variables: 102
$ ViolentCrimesPerPop   <dbl> 41.02, 127.56, 218.59, 306.64, 442.95, 226.63, 4...
$ PctForeignBorn        <dbl> 10.66, 8.30, 5.00, 2.04, 1.49, 9.19, 0.87, 1.99,...
$ PctBornSameState      <dbl> 53.72, 77.17, 44.77, 88.71, 64.35, 77.30, 73.70,...
$ PctFam2Par            <dbl> 91.43, 86.91, 78.54, 64.02, 71.94, 79.53, 62.56,...
$ PctKids2Par           <dbl> 90.17, 85.33, 78.85, 62.36, 69.79, 79.76, 58.70,...
$ PctYoungKids2Par      <dbl> 95.78, 96.82, 92.37, 65.38, 79.76, 92.05, 69.89,...
$ PctTeen2Par           <dbl> 95.81, 86.46, 75.72, 67.43, 75.33, 77.12, 62.76,...
$ PctWorkMomYoungKids   <dbl> 44.56, 51.14, 66.08, 59.59, 62.96, 65.16, 63.08,...
$ PctWorkMom            <dbl> 58.88, 62.43, 74.19, 70.27, 70.52, 72.81, 72.44,...
$ NumKidsBornNeverMar   <int> 31, 43, 164, 561, 1511, 263, 2368, 751, 3537, 60...
$ PctKidsBornNeverMar   <dbl> 0.36, 0.24, 0.88, 3.84, 1.58, 1.18, 4.66, 1.64, ...
$ numbUrban             <int> 11980, 23123, 29344, 0, 140494, 28700, 59449, 74...
$ pctUrban              <dbl> 100.00, 100.00, 100.00, 0.00, 100.00, 100.00, 10...
$ LandArea              <dbl> 6.5, 10.6, 10.6, 5.2, 70.4, 10.9, 39.2, 30.9, 78...
$ PopDens               <dbl> 1845.9, 2186.7, 2780.9, 3217.7, 1995.7, 2643.5, ...
$ PctUsePubTrans        <dbl> 9.63, 3.84, 4.37, 3.31, 0.97, 9.62, 0.70, 1.41, ...
$ LemasPctOfficDrugUn   <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, ...
$ PctLess9thGrade       <dbl> 5.81, 5.61, 2.80, 11.05, 8.76, 4.49, 10.09, 5.52...
$ PctNotHSGrad          <dbl> 9.90, 13.72, 9.09, 33.68, 23.03, 13.89, 28.67, 1...
$ PctBSorMore           <dbl> 48.18, 29.89, 30.13, 10.81, 20.66, 27.01, 12.00,...
$ PctUnemployed         <dbl> 2.70, 2.43, 4.01, 9.86, 5.72, 4.85, 8.19, 4.18, ...
$ PctEmploy             <dbl> 64.55, 61.96, 69.80, 54.74, 59.02, 65.42, 56.59,...
$ PctEmplManu           <dbl> 14.65, 12.26, 15.95, 31.22, 14.31, 14.02, 27.00,...
$ PctEmplProfServ       <dbl> 28.82, 29.28, 21.52, 27.43, 26.83, 27.17, 21.54,...
$ PctOccupManu          <dbl> 5.49, 6.39, 8.79, 26.76, 14.72, 8.50, 21.92, 11....
$ PctOccupMgmtProf      <dbl> 50.73, 37.64, 32.48, 22.71, 23.42, 32.78, 18.02,...
$ PersPerFam            <dbl> 3.22, 3.11, 2.95, 2.98, 2.89, 3.14, 2.95, 3.00, ...
$ NumInShelters         <int> 11, 0, 16, 0, 327, 0, 21, 125, 43, 1, 20, 28, 28...
$ NumStreet             <int> 0, 0, 0, 0, 4, 0, 0, 15, 4, 0, 49, 2, 0, 1, 17, ...
$ householdsize         <dbl> 3.10, 2.82, 2.43, 2.40, 2.45, 2.60, 2.45, 2.46, ...
$ PctLargHouseFam       <dbl> 4.81, 4.25, 2.97, 3.93, 3.08, 5.08, 3.85, 2.59, ...
$ PctLargHouseOccup     <dbl> 4.17, 3.34, 2.05, 2.56, 1.92, 3.46, 2.55, 1.54, ...
$ PersPerOccupHous      <dbl> 2.99, 2.70, 2.42, 2.37, 2.28, 2.55, 2.36, 2.32, ...
$ PersPerOwnOccHous     <dbl> 3.00, 2.83, 2.69, 2.51, 2.37, 2.89, 2.42, 2.77, ...
$ PersPerRentOccHous    <dbl> 2.84, 1.96, 2.06, 2.20, 2.16, 2.09, 2.27, 1.91, ...
$ PctPersOwnOccup       <dbl> 91.46, 89.03, 64.18, 58.18, 57.81, 64.62, 65.29,...
$ PctPersDenseHous      <dbl> 0.39, 1.01, 2.03, 1.21, 2.11, 1.47, 1.90, 1.67, ...
$ PctHousLess3BR        <dbl> 11.06, 23.60, 47.46, 45.66, 53.19, 47.35, 56.30,...
$ MedNumBR              <int> 3, 3, 3, 3, 2, 3, 2, 2, 2, 2, 2, 2, 3, 3, 2, 3, ...
$ HousVacant            <int> 64, 240, 544, 669, 5119, 566, 2051, 1562, 5606, ...
$ PctHousOccup          <dbl> 98.37, 97.15, 95.68, 91.19, 91.81, 95.11, 92.22,...
$ PctHousOwnOcc         <dbl> 91.01, 84.88, 57.79, 54.89, 55.50, 56.96, 63.82,...
$ PctVacantBoarded      <dbl> 3.12, 0.00, 0.92, 2.54, 2.09, 1.41, 6.39, 0.45, ...
$ PctVacMore6Mos        <dbl> 37.50, 18.33, 7.54, 57.85, 26.22, 34.45, 56.36, ...
$ MedYrHousBuilt        <int> 1959, 1958, 1976, 1939, 1966, 1956, 1954, 1971, ...
$ PctHousNoPhone        <dbl> 0.00, 0.31, 1.55, 7.00, 6.13, 0.69, 8.42, 2.66, ...
```

```
$ PctWOFullPlumb         <dbl> 0.28, 0.14, 0.12, 0.87, 0.31, 0.28, 0.49, 0.19, ...
$ OwnOccLowQuart         <int> 215900, 136300, 74700, 36400, 37700, 155100, 263...
$ OwnOccMedVal           <int> 262600, 164200, 90400, 49600, 53900, 179000, 370...
$ OwnOccHiQuart          <int> 326900, 199900, 112000, 66500, 73100, 215500, 52...
$ OwnOccQrange           <int> 111000, 63600, 37300, 30100, 35400, 60400, 26100...
$ NumImmig               <int> 1277, 1920, 1468, 339, 2091, 2637, 517, 1474, 47...
$ PctImmigRecent         <dbl> 8.69, 5.21, 16.42, 13.86, 21.33, 11.38, 13.15, 2...
$ PctImmigRec5           <dbl> 13.00, 8.65, 23.98, 13.86, 30.56, 16.27, 22.82, ...
$ PctImmigRec8           <dbl> 20.99, 13.33, 32.08, 15.34, 38.02, 23.93, 28.24,...
$ PctImmigRec10          <dbl> 30.93, 22.50, 35.63, 15.34, 45.48, 27.76, 33.08,...
$ PctRecentImmig         <dbl> 0.93, 0.43, 0.82, 0.28, 0.32, 1.05, 0.11, 0.47, ...
$ PctRecImmig5           <dbl> 1.39, 0.72, 1.20, 0.28, 0.45, 1.49, 0.20, 0.67, ...
$ PctRecImmig8           <dbl> 2.24, 1.11, 1.61, 0.31, 0.57, 2.20, 0.25, 0.93, ...
$ PctRecImmig10          <dbl> 3.30, 1.87, 1.78, 0.31, 0.68, 2.55, 0.29, 1.07, ...
$ medIncome              <int> 75122, 47917, 35669, 20580, 21577, 42805, 23221,...
$ pctWWage               <dbl> 89.24, 78.99, 82.00, 68.15, 75.78, 79.47, 71.60,...
$ pctWFarmSelf           <dbl> 1.55, 1.11, 1.15, 0.24, 1.00, 0.39, 0.67, 2.93, ...
$ pctWInvInc             <dbl> 70.20, 64.11, 55.73, 38.95, 41.15, 47.70, 35.74,...
$ pctWSocSec             <dbl> 23.62, 35.50, 22.25, 39.48, 29.31, 30.23, 32.58,...
$ pctWPubAsst            <dbl> 1.03, 2.75, 2.94, 11.71, 7.12, 5.41, 8.81, 4.21,...
$ pctWRetire             <dbl> 18.39, 22.85, 14.56, 18.33, 14.09, 17.23, 22.59,...
$ medFamInc              <int> 79584, 55323, 42112, 26501, 27705, 50394, 28901,...
$ perCapInc              <int> 29711, 20148, 16946, 10810, 11878, 18193, 12161,...
$ whitePerCap            <int> 30233, 20191, 17103, 10909, 12029, 18276, 12599,...
$ blackPerCap            <int> 13600, 18137, 16644, 9984, 7382, 17342, 9820, 88...
$ indianPerCap           <int> 5725, 0, 21606, 4941, 10264, 21482, 6634, 5344, ...
$ AsianPerCap            <int> 27101, 20074, 15528, 3541, 10753, 12639, 8802, 8...
$ HispPerCap             <int> 22838, 12222, 8405, 4391, 8104, 22594, 6187, 517...
$ PctSpeakEnglOnly       <dbl> 85.68, 87.79, 93.11, 94.98, 96.87, 89.98, 97.43,...
$ PctNotSpeakEnglWell    <dbl> 1.37, 1.81, 1.14, 0.56, 0.60, 0.60, 0.28, 0.43, ...
$ MalePctDivorce         <dbl> 3.67, 4.23, 10.10, 10.98, 11.40, 5.97, 13.28, 7....
$ MalePctNevMarr         <dbl> 26.38, 27.99, 25.78, 28.15, 33.32, 36.05, 28.34,...
$ FemalePctDiv           <dbl> 5.22, 6.45, 14.76, 14.47, 14.46, 9.06, 16.33, 9....
$ TotalPctDiv            <dbl> 4.47, 5.42, 12.55, 12.91, 13.04, 7.64, 14.94, 8....
$ RentLowQ               <int> 685, 467, 370, 195, 215, 463, 186, 241, 192, 234...
$ RentMedian             <int> 1001, 560, 428, 250, 280, 669, 253, 321, 281, 30...
$ RentHighQ              <int> 1001, 672, 520, 309, 349, 824, 325, 387, 369, 37...
$ RentQrange             <int> 316, 205, 150, 114, 134, 361, 139, 146, 177, 142...
$ MedRent                <int> 1001, 627, 484, 333, 340, 736, 338, 355, 353, 38...
$ MedRentPctHousInc      <dbl> 23.8, 27.6, 24.1, 28.7, 26.4, 24.4, 26.3, 25.2, ...
$ MedOwnCostPctInc       <dbl> 21.1, 20.7, 21.7, 20.6, 17.3, 20.8, 15.1, 20.7, ...
$ MedOwnCostPctIncNoMtg  <dbl> 14.0, 12.5, 11.6, 14.5, 11.7, 12.5, 12.2, 12.8, ...
$ population             <int> 11980, 23123, 29344, 16656, 140494, 28700, 59459...
$ racepctblack           <dbl> 1.37, 0.80, 0.74, 1.70, 2.51, 1.60, 14.20, 0.35,...
$ racePctWhite           <dbl> 91.78, 95.57, 94.33, 97.35, 95.65, 96.57, 84.87,...
$ racePctAsian           <dbl> 6.50, 3.44, 3.43, 0.50, 0.90, 1.47, 0.40, 1.25, ...
$ racePctHisp            <dbl> 1.88, 0.85, 2.35, 0.70, 0.95, 1.10, 0.63, 0.73, ...
$ agePct12t21            <dbl> 12.47, 11.01, 11.36, 12.55, 18.09, 11.17, 15.31,...
$ agePct12t29            <dbl> 21.44, 21.30, 25.88, 25.20, 32.89, 27.41, 27.93,...
$ agePct16t24            <dbl> 10.93, 10.48, 11.01, 12.19, 20.04, 12.76, 14.78,...
$ agePct65up             <dbl> 11.33, 17.18, 10.28, 17.57, 13.26, 14.42, 14.60,...
$ NumUnderPov            <int> 227, 885, 1389, 2831, 23223, 1126, 10320, 9603, ...
$ PctPopUnderPov         <dbl> 1.96, 3.98, 4.75, 17.23, 17.78, 4.01, 17.98, 13....
$ PctSameHouse85         <dbl> 65.29, 71.27, 36.60, 56.70, 42.29, 63.45, 54.85,...
$ PctSameCity85          <dbl> 78.09, 90.22, 61.26, 90.17, 70.61, 82.23, 85.55,...
$ PctSameState85         <dbl> 89.14, 96.12, 82.85, 96.24, 85.66, 93.53, 91.51,...

> #------------------ Building the formula -----------------------------
> #------------------ Converting all factors to Numbers ----------------

> community_formula <- f_generate_formula(community_ds_s124)

> # ------------------------------------------------------------------------
```

```
> # Model Selection : fastbw() : Modelling Begins with sanitized data-set
> # ------------------------------------------------------------------------

> community_ols_s124 <- ols(ViolentCrimesPerPop ~  PctForeignBorn +  PctBornSameState +
PctFam2Par +  PctKids2Par +  PctYoungKids2Par +  PctTeen2Par +  PctWorkMomYoungKids +  Pc
tWorkMom +  NumKidsBornNeverMar +  PctKidsBornNeverMar +  numbUrban +  pctUrban +  LandAr
ea +  PopDens +  PctUsePubTrans +  LemasPctOfficDrugUn +  PctLess9thGrade +  PctNotHSGrad
+  PctBSorMore +  PctUnemployed +  PctEmploy +  PctEmplManu +  PctEmplProfServ +  PctOccu
pManu +  PctOccupMgmtProf +  PersPerFam +  NumInShelters +  NumStreet +  householdsize +
PctLargHouseFam +  PctLargHouseOccup +  PersPerOccupHous +  PersPerOwnOccHous +  PersPerR
entOccHous +  PctPersOwnOccup +  PctPersDenseHous +  PctHousLess3BR +  MedNumBR +  HousVa
cant +  PctHousOccup +  PctHousOwnOcc +  PctVacantBoarded +  PctVacMore6Mos +  MedYrHousB
uilt +  PctHousNoPhone +  PctWOFullPlumb +  OwnOccLowQuart +  OwnOccMedVal +  OwnOccHiQua
rt                 +  NumImmig +  PctImmigRecent +  PctImmigRec5 +  PctImmigRec8 +  PctIm
migRec10 +  PctRecentImmig +  PctRecImmig5 +  PctRecImmig8 +  PctRecImmig10 +  medIncome
+  pctWWage +  pctWFarmSelf +  pctWInvInc +  pctWSocSec +  pctWPubAsst +  pctWRetire +  m
edFamInc +  perCapInc +  whitePerCap +  blackPerCap +  indianPerCap +  AsianPerCap +  His
pPerCap +  PctSpeakEnglOnly +  PctNotSpeakEnglWell +  MalePctDivorce +  MalePctNevMarr +
FemalePctDiv +  TotalPctDiv +  RentLowQ +  RentMedian +  RentHighQ +  MedRent +  MedRentP
ctHousInc +  MedOwnCostPctInc +  MedOwnCostPctIncNoMtg +  population +  racepctblack +  r
acePctWhite +  racePctAsian +  racePctHisp +  agePct12t21 +  agePct12t29 +  agePct16t24 +
agePct65up +  NumUnderPov +  PctPopUnderPov )

> fastbw(community_ols_s124, rule = "p", sls = 0.05)

 Deleted              Chi-Sq d.f. P      Residual d.f. P      AIC    R2
 RentMedian           0.00   1    0.9809  0.00     1   0.9809  -2.00 0.675
 PctImmigRec5         0.00   1    0.9804  0.00     2   0.9994  -4.00 0.675
 PctRecentImmig       0.00   1    0.9664  0.00     3   1.0000  -6.00 0.675
 whitePerCap          0.00   1    0.9648  0.00     4   1.0000  -8.00 0.675
 PctTeen2Par          0.00   1    0.9620  0.01     5   1.0000  -9.99 0.675
 racePctHisp          0.00   1    0.9609  0.01     6   1.0000 -11.99 0.675
 MedRentPctHousInc    0.00   1    0.9452  0.01     7   1.0000 -13.99 0.675
 agePct16t24          0.01   1    0.9368  0.02     8   1.0000 -15.98 0.675
 OwnOccMedVal         0.02   1    0.8899  0.04     9   1.0000 -17.96 0.675
 NumUnderPov          0.02   1    0.8872  0.06    10   1.0000 -19.94 0.675
 PctUsePubTrans       0.03   1    0.8677  0.09    11   1.0000 -21.91 0.675
 population           0.04   1    0.8361  0.13    12   1.0000 -23.87 0.675
 LandArea             0.03   1    0.8536  0.16    13   1.0000 -25.84 0.675
 PctRecImmig10        0.05   1    0.8270  0.21    14   1.0000 -27.79 0.675
 PctRecImmig8         0.03   1    0.8738  0.24    15   1.0000 -29.76 0.675
 indianPerCap         0.06   1    0.8097  0.30    16   1.0000 -31.70 0.675
 PctImmigRec8         0.06   1    0.8077  0.35    17   1.0000 -33.65 0.675
 PctBornSameState     0.09   1    0.7620  0.45    18   1.0000 -35.55 0.675
 PctEmplProfServ      0.09   1    0.7648  0.54    19   1.0000 -37.46 0.675
 NumStreet            0.11   1    0.7430  0.64    20   1.0000 -39.36 0.675
 racePctWhite         0.11   1    0.7390  0.75    21   1.0000 -41.25 0.675
 racePctAsian         0.07   1    0.7850  0.83    22   1.0000 -43.17 0.675
 PctOccupManu         0.20   1    0.6516  1.03    23   1.0000 -44.97 0.675
 PctSpeakEnglOnly     0.19   1    0.6620  1.22    24   1.0000 -46.78 0.675
 blackPerCap          0.21   1    0.6438  1.44    25   1.0000 -48.56 0.675
 PctImmigRec10        0.21   1    0.6442  1.65    26   1.0000 -50.35 0.675
 PctOccupMgmtProf     0.32   1    0.5709  1.97    27   1.0000 -52.03 0.675
 PctUnemployed        0.36   1    0.5461  2.34    28   1.0000 -53.66 0.675
 MedYrHousBuilt       0.30   1    0.5863  2.63    29   1.0000 -55.37 0.675
 PctHousOccup         0.36   1    0.5484  2.99    30   1.0000 -57.01 0.675
 PctWOFullPlumb       0.35   1    0.5520  3.35    31   1.0000 -58.65 0.675
 PctFam2Par           0.37   1    0.5406  3.72    32   1.0000 -60.28 0.675
 RentHighQ            0.59   1    0.4437  4.31    33   1.0000 -61.69 0.674
 PctYoungKids2Par     0.60   1    0.4383  4.91    34   1.0000 -63.09 0.674
 PctNotSpeakEnglWell  0.62   1    0.4323  5.53    35   1.0000 -64.47 0.674
 agePct65up           0.62   1    0.4298  6.15    36   1.0000 -65.85 0.674
 pctWSocSec           0.31   1    0.5789  6.46    37   1.0000 -67.54 0.674
```

PctLargHouseFam	0.67	1	0.4124	7.13	38	1.0000	-68.87	0.674
PctHousNoPhone	0.77	1	0.3792	7.90	39	1.0000	-70.10	0.674
PersPerFam	0.66	1	0.4181	8.56	40	1.0000	-71.44	0.674
PersPerOccupHous	0.58	1	0.4475	9.13	41	1.0000	-72.87	0.674
PctHousLess3BR	0.71	1	0.3989	9.85	42	1.0000	-74.15	0.673
MedNumBR	0.90	1	0.3423	10.75	43	1.0000	-75.25	0.673
perCapInc	0.95	1	0.3302	11.70	44	1.0000	-76.30	0.673
householdsize	0.91	1	0.3398	12.61	45	1.0000	-77.39	0.673
MedOwnCostPctInc	1.34	1	0.2462	13.95	46	1.0000	-78.05	0.673
OwnOccLowQuart	1.34	1	0.2472	15.29	47	1.0000	-78.71	0.673
pctWPubAsst	1.46	1	0.2271	16.75	48	1.0000	-79.25	0.672
PctBSorMore	1.38	1	0.2395	18.13	49	1.0000	-79.87	0.672
FemalePctDiv	1.64	1	0.2007	19.77	50	1.0000	-80.23	0.672
HispPerCap	1.71	1	0.1910	21.48	51	0.9999	-80.52	0.671
PctWorkMomYoungKids	1.90	1	0.1680	23.38	52	0.9998	-80.62	0.671
agePct12t21	1.99	1	0.1580	25.37	53	0.9995	-80.63	0.671
PctImmigRecent	2.36	1	0.1244	27.74	54	0.9989	-80.26	0.670
PctPopUnderPov	2.17	1	0.1411	29.90	55	0.9977	-80.10	0.670
NumKidsBornNeverMar	2.70	1	0.1003	32.60	56	0.9948	-79.40	0.670
OwnOccHiQuart	2.87	1	0.0905	35.47	57	0.9888	-78.53	0.669
medFamInc	1.86	1	0.1726	37.33	58	0.9841	-78.67	0.669
medIncome	1.27	1	0.2600	38.60	59	0.9817	-79.40	0.669
pctWFarmSelf	2.47	1	0.1163	41.07	60	0.9707	-78.93	0.668
PctRecImmig5	2.55	1	0.1102	43.62	61	0.9547	-78.38	0.668
PctForeignBorn	2.58	1	0.1082	46.20	62	0.9332	-77.80	0.667
NumInShelters	3.19	1	0.0740	49.39	63	0.8947	-76.61	0.667
NumImmig	3.42	1	0.0643	52.81	64	0.8397	-75.19	0.666
AsianPerCap	3.83	1	0.0504	56.64	65	0.7605	-73.36	0.665
PctLargHouseOccup	4.70	1	0.0301	61.34	66	0.6394	-70.66	0.665
numbUrban	4.85	1	0.0276	66.20	67	0.5047	-67.80	0.664
PctVacMore6Mos	5.82	1	0.0158	72.02	68	0.3463	-63.98	0.663
pctWInvInc	5.87	1	0.0154	77.89	69	0.2169	-60.11	0.662
MalePctNevMarr	4.55	1	0.0329	82.44	70	0.1467	-57.56	0.661
agePct12t29	4.59	1	0.0321	87.04	71	0.0949	-54.96	0.660
PopDens	3.03	1	0.0816	90.07	72	0.0735	-53.93	0.660

Approximate Estimates after Deleting Factors

	Coef	S.E.	Wald Z	P
Intercept	2.675e+03	3.215e+02	8.320	1.110e-16
PctKids2Par	-1.280e+01	3.086e+00	-4.148	3.353e-05
PctWorkMom	-5.848e+00	1.692e+00	-3.457	5.469e-04
PctKidsBornNeverMar	3.548e+01	7.972e+00	4.451	8.552e-06
pctUrban	9.432e-01	2.274e-01	4.148	3.352e-05
LemasPctOfficDrugUn	9.107e+00	3.076e+00	2.961	3.069e-03
PctLess9thGrade	-2.075e+01	4.558e+00	-4.552	5.320e-06
PctNotHSGrad	1.217e+01	3.301e+00	3.685	2.283e-04
PctEmploy	1.177e+01	2.796e+00	4.208	2.572e-05
PctEmplManu	-5.067e+00	1.214e+00	-4.175	2.978e-05
PersPerOwnOccHous	2.973e+02	1.030e+02	2.887	3.887e-03
PersPerRentOccHous	-3.454e+02	9.329e+01	-3.703	2.131e-04
PctPersOwnOccup	-3.817e+01	1.098e+01	-3.476	5.094e-04
PctPersDenseHous	2.388e+01	3.615e+00	6.605	3.970e-11
HousVacant	6.573e-03	1.476e-03	4.455	8.400e-06
PctHousOwnOcc	3.718e+01	1.044e+01	3.562	3.684e-04
PctVacantBoarded	1.038e+01	3.150e+00	3.294	9.873e-04
pctWWage	-1.721e+01	3.039e+00	-5.663	1.485e-08
pctWRetire	-1.234e+01	2.679e+00	-4.608	4.073e-06
MalePctDivorce	5.608e+01	1.504e+01	3.727	1.934e-04
TotalPctDiv	-3.591e+01	1.510e+01	-2.378	1.741e-02
RentLowQ	-7.254e-01	2.073e-01	-3.500	4.649e-04
MedRent	7.577e-01	1.765e-01	4.293	1.764e-05
MedOwnCostPctIncNoMtg	-3.439e+01	7.130e+00	-4.823	1.414e-06

```
racepctblack           8.614e+00 1.174e+00  7.336 2.194e-13

Factors in Final Model

 [1] PctKids2Par           PctWorkMom            PctKidsBornNeverMar
 [4] pctUrban              LemasPctOfficDrugUn   PctLess9thGrade
 [7] PctNotHSGrad          PctEmploy             PctEmplManu
[10] PersPerOwnOccHous     PersPerRentOccHous    PctPersOwnOccup
[13] PctPersDenseHous      HousVacant            PctHousOwnOcc
[16] PctVacantBoarded      pctWWage              pctWRetire
[19] MalePctDivorce        TotalPctDiv           RentLowQ
[22] MedRent               MedOwnCostPctIncNoMtg racepctblack

> community_mdl_s124_fb <- lm(ViolentCrimesPerPop ~ PctKids2Par + PctWorkMom + PctKidsBor
nNeverMar + pctUrban + LemasPctOfficDrugUn + PctLess9thGrade + PctNotHSGrad + PctEmploy +
PctEmplManu + PersPerOwnOccHous + PersPerRentOccHous + PctPersOwnOccup + PctPersDenseHous
+ HousVacant + PctHousOwnOcc + PctVacantBoarded + pctWWage + pctWRetire + MalePctDivorce
+ TotalPctDiv + RentLowQ + MedRent + MedOwnCostPctIncNoMtg + racepctblack)

> summary(community_mdl_s124_fb)

Call:
lm(formula = ViolentCrimesPerPop ~ PctKids2Par + PctWorkMom +
    PctKidsBornNeverMar + pctUrban + LemasPctOfficDrugUn + PctLess9thGrade +
    PctNotHSGrad + PctEmploy + PctEmplManu + PersPerOwnOccHous +
    PersPerRentOccHous + PctPersOwnOccup + PctPersDenseHous +
    HousVacant + PctHousOwnOcc + PctVacantBoarded + pctWWage +
    pctWRetire + MalePctDivorce + TotalPctDiv + RentLowQ + MedRent +
    MedOwnCostPctIncNoMtg + racepctblack)

Residuals:
     Min       1Q   Median       3Q      Max
-1594.59  -186.55   -41.23   125.12  2317.60

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)           2.675e+03  3.229e+02   8.282  < 2e-16 ***
PctKids2Par          -1.280e+01  3.100e+00  -4.129 3.79e-05 ***
PctWorkMom           -5.848e+00  1.700e+00  -3.441 0.000592 ***
PctKidsBornNeverMar   3.548e+01  8.008e+00   4.431 9.91e-06 ***
pctUrban              9.432e-01  2.284e-01   4.129 3.79e-05 ***
LemasPctOfficDrugUn   9.107e+00  3.090e+00   2.947 0.003244 **
PctLess9thGrade      -2.075e+01  4.578e+00  -4.531 6.22e-06 ***
PctNotHSGrad          1.217e+01  3.316e+00   3.669 0.000250 ***
PctEmploy             1.177e+01  2.809e+00   4.189 2.92e-05 ***
PctEmplManu          -5.067e+00  1.219e+00  -4.156 3.38e-05 ***
PersPerOwnOccHous     2.973e+02  1.034e+02   2.874 0.004096 **
PersPerRentOccHous   -3.454e+02  9.371e+01  -3.686 0.000234 ***
PctPersOwnOccup      -3.817e+01  1.103e+01  -3.460 0.000552 ***
PctPersDenseHous      2.388e+01  3.632e+00   6.575 6.21e-11 ***
HousVacant            6.573e-03  1.482e-03   4.434 9.74e-06 ***
PctHousOwnOcc         3.718e+01  1.049e+01   3.546 0.000401 ***
PctVacantBoarded      1.038e+01  3.165e+00   3.279 0.001060 **
pctWWage             -1.721e+01  3.053e+00  -5.638 1.97e-08 ***
pctWRetire           -1.234e+01  2.691e+00  -4.587 4.79e-06 ***
MalePctDivorce        5.608e+01  1.511e+01   3.711 0.000213 ***
TotalPctDiv          -3.591e+01  1.517e+01  -2.367 0.018023 *
RentLowQ             -7.254e-01  2.082e-01  -3.484 0.000504 ***
MedRent               7.577e-01  1.773e-01   4.273 2.02e-05 ***
MedOwnCostPctIncNoMtg -3.439e+01  7.163e+00  -4.801 1.70e-06 ***
racepctblack          8.614e+00  1.179e+00   7.303 4.07e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 360.8 on 1969 degrees of freedom
Multiple R-squared:  0.6597,   Adjusted R-squared:  0.6556
F-statistic: 159.1 on 24 and 1969 DF,  p-value: < 2.2e-16

> # ------------------------------------------------------------------------------
> # Model Selection : AIC : Modelling Begins with sanitized data-set
> # ------------------------------------------------------------------------------

> community_mdl_s124_AIC <- lm(community_formula, community_ds_s124)
> summary(community_mdl_s124_AIC)

Call:
lm(formula = community_formula, data = community_ds_s124)

Residuals:
     Min       1Q   Median       3Q      Max
-1691.66  -183.89   -39.82   131.35  2226.77

Coefficients: (2 not defined because of singularities)
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           4.987e+03  3.461e+03   1.441 0.149769
PctForeignBorn        1.345e+01  6.829e+00   1.969 0.049084 *
PctBornSameState     -2.046e-01  1.366e+00  -0.150 0.880991
PctFam2Par            4.676e+00  8.905e+00   0.525 0.599570
PctKids2Par          -1.514e+01  7.514e+00  -2.015 0.044042 *
PctYoungKids2Par      1.652e+00  2.402e+00   0.688 0.491672
PctTeen2Par           1.453e-01  2.189e+00   0.066 0.947075
PctWorkMomYoungKids   3.799e+00  2.767e+00   1.373 0.169827
PctWorkMom           -1.156e+01  4.012e+00  -2.881 0.004004 **
NumKidsBornNeverMar  -4.745e-03  4.205e-03  -1.128 0.259301
PctKidsBornNeverMar   4.072e+01  9.847e+00   4.136 3.69e-05 ***
numbUrban            -1.012e-03  2.049e-03  -0.494 0.621548
pctUrban              1.249e+00  4.814e-01   2.594 0.009569 **
LandArea             -1.801e-02  8.256e-02  -0.218 0.827320
PopDens              -1.299e-02  5.478e-03  -2.371 0.017863 *
PctUsePubTrans       -3.048e-01  3.083e+00  -0.099 0.921256
LemasPctOfficDrugUn   9.757e+00  3.247e+00   3.005 0.002693 **
PctLess9thGrade      -1.796e+01  6.341e+00  -2.832 0.004668 **
PctNotHSGrad          7.860e+00  4.929e+00   1.595 0.110977
PctBSorMore           2.762e+00  3.528e+00   0.783 0.433732
PctUnemployed        -4.644e+00  8.093e+00  -0.574 0.566144
PctEmploy             9.961e+00  4.815e+00   2.069 0.038708 *
PctEmplManu          -4.644e+00  2.197e+00  -2.113 0.034696 *
PctEmplProfServ      -1.120e+00  2.938e+00  -0.381 0.703023
PctOccupManu          2.321e+00  4.648e+00   0.499 0.617604
PctOccupMgmtProf      3.548e+00  4.682e+00   0.758 0.448610
PersPerFam           -4.712e+02  3.612e+02  -1.305 0.192180
NumInShelters         1.125e-01  6.132e-02   1.835 0.066734 .
NumStreet            -5.392e-02  1.444e-01  -0.373 0.708955
householdsize        -1.588e+02  1.173e+02  -1.355 0.175711
PctLargHouseFam       2.383e+01  3.197e+01   0.745 0.456153
PctLargHouseOccup    -4.060e+01  3.477e+01  -1.168 0.243154
PersPerOccupHous      6.152e+02  4.205e+02   1.463 0.143618
PersPerOwnOccHous     3.790e+02  2.806e+02   1.351 0.177016
PersPerRentOccHous   -3.518e+02  1.166e+02  -3.017 0.002591 **
PctPersOwnOccup      -4.740e+01  1.712e+01  -2.769 0.005683 **
PctPersDenseHous      1.959e+01  7.596e+00   2.579 0.009977 **
PctHousLess3BR        2.866e+00  2.070e+00   1.384 0.166430
MedNumBR              3.294e+01  2.655e+01   1.240 0.214952
HousVacant            2.245e-02  6.176e-03   3.636 0.000285 ***
PctHousOccup         -2.239e+00  2.658e+00  -0.842 0.399730
PctHousOwnOcc         4.273e+01  1.711e+01   2.497 0.012602 *
```

```
PctVacantBoarded       1.304e+01  3.621e+00   3.601 0.000325 ***
PctVacMore6Mos        -2.321e+00  9.145e-01  -2.538 0.011220 *
MedYrHousBuilt        -8.644e-01  1.626e+00  -0.531 0.595162
PctHousNoPhone         3.469e+00  5.653e+00   0.614 0.539457
PctWOFullPlumb        -1.296e+01  2.640e+01  -0.491 0.623427
OwnOccLowQuart         3.922e-04  1.335e-03   0.294 0.768941
OwnOccMedVal           2.607e-04  1.575e-03   0.165 0.868582
OwnOccHiQuart         -7.713e-04  6.769e-04  -1.139 0.254684
OwnOccQrange                  NA         NA      NA       NA
NumImmig               1.294e-03  6.796e-04   1.904 0.057040 .
PctImmigRecent         1.646e+00  2.416e+00   0.681 0.495731
PctImmigRec5          -4.788e-02  3.074e+00  -0.016 0.987575
PctImmigRec8          -7.520e-01  2.990e+00  -0.252 0.801444
PctImmigRec10          9.330e-01  1.944e+00   0.480 0.631337
PctRecentImmig        -2.540e+00  4.994e+01  -0.051 0.959435
PctRecImmig5          -3.072e+01  6.280e+01  -0.489 0.624779
PctRecImmig8           1.378e+01  5.757e+01   0.239 0.810793
PctRecImmig10         -8.130e+00  3.416e+01  -0.238 0.811912
medIncome             -1.191e-02  7.182e-03  -1.658 0.097515 .
pctWWage              -1.386e+01  5.908e+00  -2.346 0.019058 *
pctWFarmSelf           2.751e+01  1.572e+01   1.750 0.080229 .
pctWInvInc            -4.327e+00  2.597e+00  -1.666 0.095790 .
pctWSocSec             5.037e+00  5.937e+00   0.848 0.396272
pctWPubAsst            9.855e+00  6.060e+00   1.626 0.104043
pctWRetire            -1.184e+01  3.790e+00  -3.123 0.001816 **
medFamInc              8.367e-03  6.903e-03   1.212 0.225614
perCapInc             -7.826e-03  1.613e-02  -0.485 0.627599
whitePerCap           -5.273e-04  1.281e-02  -0.041 0.967179
blackPerCap           -5.375e-04  1.089e-03  -0.494 0.621674
indianPerCap          -1.256e-04  5.512e-04  -0.228 0.819749
AsianPerCap            1.912e-03  9.893e-04   1.933 0.053413 .
HispPerCap             2.703e-03  1.954e-03   1.383 0.166820
PctSpeakEnglOnly      -9.331e-01  3.423e+00  -0.273 0.785222
PctNotSpeakEnglWell   -9.108e+00  1.044e+01  -0.873 0.382936
MalePctDivorce         1.575e+02  6.873e+01   2.292 0.022019 *
MalePctNevMarr         1.136e+01  4.885e+00   2.325 0.020185 *
FemalePctDiv           1.016e+02  7.191e+01   1.412 0.158058
TotalPctDiv           -2.431e+02  1.393e+02  -1.746 0.081012 .
RentLowQ              -7.259e-01  2.650e-01  -2.740 0.006205 **
RentMedian            -4.936e-03  4.915e-01  -0.010 0.991988
RentHighQ             -1.998e-01  2.849e-01  -0.701 0.483179
RentQrange                    NA         NA      NA       NA
MedRent                9.541e-01  4.296e-01   2.221 0.026492 *
MedRentPctHousInc     -3.772e-01  5.219e+00  -0.072 0.942385
MedOwnCostPctInc      -4.225e+00  5.959e+00  -0.709 0.478439
MedOwnCostPctIncNoMtg -2.751e+01  8.939e+00  -3.077 0.002119 **
population             3.744e-04  2.083e-03   0.180 0.857366
racepctblack           7.683e+00  3.222e+00   2.384 0.017201 *
racePctWhite          -7.787e-01  3.034e+00  -0.257 0.797470
racePctAsian          -1.706e+00  5.091e+00  -0.335 0.737598
racePctHisp           -1.026e-01  2.864e+00  -0.036 0.971439
agePct12t21            1.522e+01  1.281e+01   1.188 0.234863
agePct12t29           -2.633e+01  1.266e+01  -2.080 0.037647 *
agePct16t24            1.343e+00  1.844e+01   0.073 0.941942
agePct65up            -8.130e+00  1.088e+01  -0.747 0.455116
NumUnderPov           -3.570e-04  2.447e-03  -0.146 0.884023
PctPopUnderPov        -9.583e+00  4.894e+00  -1.958 0.050380 .
PctSameHouse85        -8.237e-01  2.722e+00  -0.303 0.762196
PctSameCity85          1.218e+00  2.002e+00   0.608 0.543076
PctSameState85         9.838e-01  3.185e+00   0.309 0.757409
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

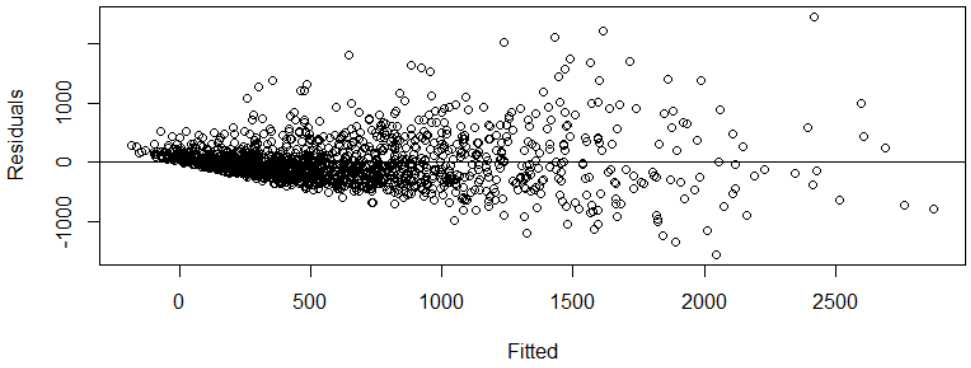
```
Residual standard error: 359.4 on 1894 degrees of freedom
Multiple R-squared:  0.6752,   Adjusted R-squared:  0.6583
F-statistic: 39.78 on 99 and 1894 DF,  p-value: < 2.2e-16

> t1 <- stepAIC(community_mdl_s124_AIC, trace = 0)

> extractAIC(t1)
[1]    45.00 23479.43

> final_formula <- t1$call

> community_mdl_s124_AIC <- lm(final_formula)

> summary(community_mdl_s124_AIC)

Call:
lm(formula = final_formula)

Residuals:
     Min       1Q   Median       3Q      Max
-1639.76  -188.66   -38.47   128.55  2247.59

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)           3.035e+03  5.146e+02   5.898 4.32e-09 ***
PctForeignBorn        1.109e+01  3.838e+00   2.889 0.003907 **
PctKids2Par          -1.217e+01  3.523e+00  -3.455 0.000563 ***
PctWorkMom           -7.996e+00  1.955e+00  -4.091 4.47e-05 ***
NumKidsBornNeverMar  -5.020e-03  2.923e-03  -1.718 0.086030 .
PctKidsBornNeverMar   3.658e+01  8.419e+00   4.344 1.47e-05 ***
numbUrban            -6.307e-04  3.184e-04  -1.980 0.047798 *
pctUrban              1.147e+00  2.566e-01   4.470 8.26e-06 ***
PopDens              -1.295e-02  4.474e-03  -2.894 0.003845 **
LemasPctOfficDrugUn   9.398e+00  3.146e+00   2.987 0.002853 **
PctLess9thGrade      -2.035e+01  5.286e+00  -3.850 0.000122 ***
PctNotHSGrad          9.323e+00  3.976e+00   2.345 0.019133 *
PctEmploy             1.226e+01  3.430e+00   3.574 0.000360 ***
PctEmplManu          -3.314e+00  1.294e+00  -2.560 0.010530 *
NumInShelters         9.755e-02  4.364e-02   2.235 0.025519 *
PctLargHouseOccup    -1.894e+01  8.784e+00  -2.156 0.031238 *
PersPerOwnOccHous     4.824e+02  1.397e+02   3.453 0.000566 ***
PersPerRentOccHous   -3.326e+02  9.983e+01  -3.331 0.000881 ***
PctPersOwnOccup      -5.341e+01  1.231e+01  -4.340 1.50e-05 ***
PctPersDenseHous      2.308e+01  5.151e+00   4.480 7.90e-06 ***
HousVacant            2.130e-02  5.029e-03   4.236 2.38e-05 ***
PctHousOwnOcc         4.970e+01  1.188e+01   4.183 3.01e-05 ***
PctVacantBoarded      1.345e+01  3.393e+00   3.964 7.64e-05 ***
PctVacMore6Mos       -2.072e+00  8.037e-01  -2.578 0.010003 *
OwnOccHiQuart        -4.128e-04  2.321e-04  -1.778 0.075479 .
NumImmig              1.079e-03  5.713e-04   1.889 0.059095 .
PctImmigRecent        1.696e+00  1.132e+00   1.498 0.134197
PctRecImmig5         -2.737e+01  1.201e+01  -2.280 0.022733 *
medIncome            -1.159e-02  6.135e-03  -1.889 0.059092 .
pctWWage             -1.622e+01  4.124e+00  -3.934 8.65e-05 ***
pctWFarmSelf          2.915e+01  1.472e+01   1.980 0.047806 *
pctWInvInc           -4.702e+00  2.243e+00  -2.096 0.036228 *
pctWRetire           -1.128e+01  3.155e+00  -3.576 0.000357 ***
medFamInc             8.749e-03  5.306e-03   1.649 0.099336 .
AsianPerCap           1.997e-03  9.424e-04   2.119 0.034244 *
MalePctDivorce        5.701e+01  1.556e+01   3.664 0.000255 ***
MalePctNevMarr        1.140e+01  3.432e+00   3.321 0.000913 ***
TotalPctDiv          -4.469e+01  1.651e+01  -2.707 0.006853 **
RentLowQ             -6.641e-01  2.150e-01  -3.090 0.002032 **
```

```
MedRent                  6.901e-01  2.143e-01   3.221 0.001301 **
MedOwnCostPctIncNoMtg -3.405e+01  7.578e+00  -4.494 7.41e-06 ***
racepctblack             8.445e+00  1.261e+00   6.699 2.74e-11 ***
agePct12t21              8.812e+00  6.196e+00   1.422 0.155099
agePct12t29             -2.267e+01  6.882e+00  -3.294 0.001007 **
PctPopUnderPov          -5.878e+00  3.383e+00  -1.738 0.082454 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 356.5 on 1949 degrees of freedom
Multiple R-squared:  0.6711,   Adjusted R-squared:  0.6637
F-statistic:  90.4 on 44 and 1949 DF,  p-value: < 2.2e-16

> # ------------------------------------------------------------------------
> # Model Selection for limited predictor variables (8)
> # Creating the initial dataset with 1+8 Variables
> # ------------------------------------------------------------------------

> community_all %>% dplyr::select(ViolentCrimesPerPop, racepctblack, PctPersDenseHous,
pctUrban, PctKidsBornNeverMar, HousVacant, pctWWage, MalePctDivorce, pctWRetire) ->
community_ds_s8

> # ------------------------------------------------------------------------
> # Removing all variables which have either ? or NA
> # ------------------------------------------------------------------------

> bad_vars <- f_get_NA_vars(community_ds_s8)

> # -------------- Removing all such bad variables and Rows with NAs.-------------------
> community_ds_s8 %>% dplyr::select(-bad_vars) -> community_ds_s8

Error in -x : invalid argument to unary operator

> community_ds_s8 %>% na.omit() -> community_ds_s8

> #------------------ Building the formula -----------------------------
> community_formula <- f_generate_formula(community_ds_s8)

> # ------------------------------------------------------------------------
> # Split data into training and test samples
> # ------------------------------------------------------------------------
> n = dim(community_ds_s8)[1]
> train_index <- sample(1:n, 0.8*n, replace = F)
> test_index  <- setdiff(1:n, train_index)

> # ------------------------------------------------------------------------
> # Use the indices to define the training and test samples
> # ------------------------------------------------------------------------
> community_train_ds <- community_ds_s8[train_index,]
> community_test_ds  <- community_ds_s8[test_index,]

> attach(community_train_ds)

> # ------------------------------------------------------------------------------
> # Model Selection : fastbw() : Modelling Begins with sanitized data-set
> # ------------------------------------------------------------------------------
> community_ols_s8 <- ols(ViolentCrimesPerPop ~ racepctblack + PctPersDenseHous +
pctUrban + PctKidsBornNeverMar + HousVacant + pctWWage + MalePctDivorce + pctWRetire)

> fastbw(community_ols_s8, rule = "p", sls = 0.05)

No Factors Deleted
Factors in Final Model
```

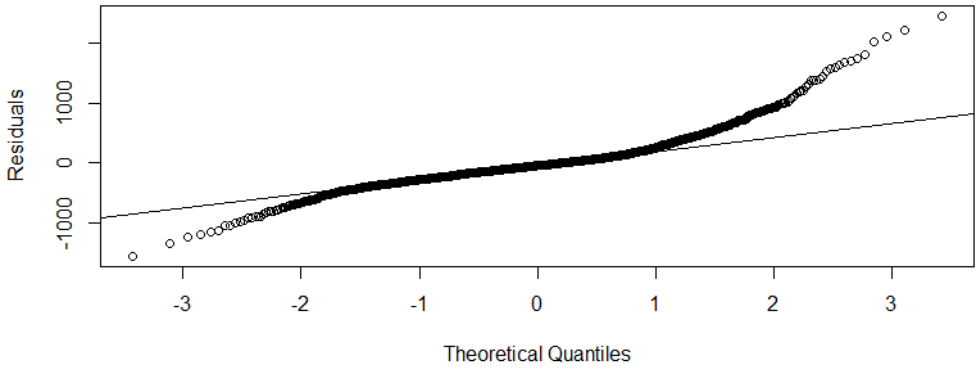
```
[1] racepctblack        PctPersDenseHous    pctUrban            PctKidsBornNeverMar
[5] HousVacant          pctWWage            MalePctDivorce      pctWRetire

> community_mdl_s8_fb <- lm(ViolentCrimesPerPop ~ racepctblack + PctPersDenseHous + pctUr
ban + PctKidsBornNeverMar + HousVacant + pctWWage + MalePctDivorce + pctWRetire)

> summary(community_mdl_s8_fb)

Call:
lm(formula = ViolentCrimesPerPop ~ racepctblack + PctPersDenseHous +
    pctUrban + PctKidsBornNeverMar + HousVacant + pctWWage +
    MalePctDivorce + pctWRetire)

Residuals:
     Min       1Q   Median       3Q      Max
-1564.71  -189.12   -36.95   120.44  2449.56

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         725.749454 175.840651   4.127 3.86e-05 ***
racepctblack          8.619682   1.195490   7.210 8.62e-13 ***
PctPersDenseHous     20.354836   1.922747  10.586  < 2e-16 ***
pctUrban              1.387255   0.225801   6.144 1.02e-09 ***
PctKidsBornNeverMar  70.231731   6.320571  11.112  < 2e-16 ***
HousVacant            0.008299   0.001520   5.461 5.49e-08 ***
pctWWage            -11.041426   1.714942  -6.438 1.60e-10 ***
MalePctDivorce       40.123894   3.866355  10.378  < 2e-16 ***
pctWRetire           -9.227563   2.739564  -3.368 0.000775 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 364.2 on 1586 degrees of freedom
Multiple R-squared:  0.6462,   Adjusted R-squared:  0.6444
F-statistic: 362.1 on 8 and 1586 DF,  p-value: < 2.2e-16

> # ------------------------------------------------------------------------
> # Model Selection : StepAIC() : Modelling Begins with sanitized data-set
> # ------------------------------------------------------------------------

> community_mdl1 <- lm(community_formula)
> summary(community_mdl1)

Call:
lm(formula = community_formula)

Residuals:
     Min       1Q   Median       3Q      Max
-1564.71  -189.12   -36.95   120.44  2449.56

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         725.749454 175.840651   4.127 3.86e-05 ***
racepctblack          8.619682   1.195490   7.210 8.62e-13 ***
PctPersDenseHous     20.354836   1.922747  10.586  < 2e-16 ***
pctUrban              1.387255   0.225801   6.144 1.02e-09 ***
PctKidsBornNeverMar  70.231731   6.320571  11.112  < 2e-16 ***
HousVacant            0.008299   0.001520   5.461 5.49e-08 ***
pctWWage            -11.041426   1.714942  -6.438 1.60e-10 ***
MalePctDivorce       40.123894   3.866355  10.378  < 2e-16 ***
pctWRetire           -9.227563   2.739564  -3.368 0.000775 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 364.2 on 1586 degrees of freedom
Multiple R-squared:  0.6462,   Adjusted R-squared:  0.6444
F-statistic: 362.1 on 8 and 1586 DF,  p-value: < 2.2e-16

> t1 <- stepAIC(community_mdl1, trace = 0)
> extractAIC(t1)
[1]     9.00 18822.39

> final_formula <- t1$call
> final_formula

lm(formula = ViolentCrimesPerPop ~ racepctblack + PctPersDenseHous +
    pctUrban + PctKidsBornNeverMar + HousVacant + pctWWage +
    MalePctDivorce + pctWRetire)

> community_mdl_s8_AIC <- lm(final_formula)
> summary(community_mdl_s8_AIC)

Call:
lm(formula = final_formula)

Residuals:
     Min       1Q   Median       3Q      Max
-1564.71  -189.12   -36.95   120.44  2449.56

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         725.749454 175.840651   4.127 3.86e-05 ***
racepctblack          8.619682   1.195490   7.210 8.62e-13 ***
PctPersDenseHous     20.354836   1.922747  10.586  < 2e-16 ***
pctUrban              1.387255   0.225801   6.144 1.02e-09 ***
PctKidsBornNeverMar  70.231731   6.320571  11.112  < 2e-16 ***
HousVacant            0.008299   0.001520   5.461 5.49e-08 ***
pctWWage            -11.041426   1.714942  -6.438 1.60e-10 ***
MalePctDivorce       40.123894   3.866355  10.378  < 2e-16 ***
pctWRetire           -9.227563   2.739564  -3.368 0.000775 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 364.2 on 1586 degrees of freedom
Multiple R-squared:  0.6462,   Adjusted R-squared:  0.6444
F-statistic: 362.1 on 8 and 1586 DF,  p-value: < 2.2e-16

#------------------------------------------------------------------------------
#------------------------------------------------------------------------------
#------------------------------------------------------------------------------
# STAGE 7 MODEL DIAGNOSTICS
#------------------------------------------------------------------------------
#------------------------------------------------------------------------------
#------------------------------------------------------------------------------

> # ------------------------------------------------------------------------------
> # checking for Multi-colinearity
> # Multicolinearity is not present here.
> # ------------------------------------------------------------------------------
> comm_vif <- vif(community_mdl_s8_AIC)

> comm_vif
       racepctblack    PctPersDenseHous            pctUrban PctKidsBornNeverMar
           3.394204            1.475989            1.212607            4.591688
        HousVacant            pctWWage      MalePctDivorce          pctWRetire
           1.090446            2.089594            1.425718            1.835077
```

```
> # ------------------------------------------------------------------------------
> # Fitted values vs. residuals plot Comparison for Constant Error Variance
> # As per plot, errors have constant variance and are not random.
> # Heteroscedasticity is present
> # ------------------------------------------------------------------------------

> fitted_values <- fitted(community_mdl_s8_AIC)
> residual_values <- residuals(community_mdl_s8_AIC)

> plot(fitted_values, residual_values, xlab = "Fitted", ylab = "Residuals")
> abline(community_mdl_s8_AIC)
```




```
> # ------------------------------------------------------------------------------
> # Formal Test for Constant Variance: BP Test
> # it also confirms that Heteoscedasticity is present
> # ------------------------------------------------------------------------------

> comm_bp <- bptest(community_mdl_s8_AIC)
> comm_bp

        studentized Breusch-Pagan test

data:  community_mdl_s8_AIC
BP = 278.76, df = 8, p-value < 2.2e-16

> # ------------------------------------------------------------------------------
> # checking for Normality
> # Trend line is short-tailed
> # Non-normality is present
> # ------------------------------------------------------------------------------

> qqnorm(residuals(community_mdl_s8_AIC), ylab = "Residuals", main="")
> qqline(residuals(community_mdl_s8_AIC))
```




```
> # ------------------------------------------------------------------------------
> # checking for Shapiro Test: Formal test for normality
> # p-value is less than significant value so normality is not present.
> # ------------------------------------------------------------------------------

> shapiro.test(residuals(community_mdl_s8_AIC))

        Shapiro-Wilk normality test

data:  residuals(community_mdl_s8_AIC)
W = 0.90274, p-value < 2.2e-16

> # ------------------------------------------------------------------------------
> # checking for Durban-Watson Test : Formal Test for co-related errors
> # Co-relation is present
> # ------------------------------------------------------------------------------

> dwtest(community_mdl_s8_AIC)

        Durbin-Watson test

data:  community_mdl_s8_AIC
DW = 1.9559, p-value = 0.1886
alternative hypothesis: true autocorrelation is greater than 0
```

```
#------------------------------------------------------------------------------
#------------------------------------------------------------------------------
#------------------------------------------------------------------------------
# STAGE 8: MODEL DIAGNOSTICS: OUTLIERS AND INFLUENTIAL OBSERVATIONS
#------------------------------------------------------------------------------
#------------------------------------------------------------------------------
#------------------------------------------------------------------------------

> # ------------------------------------------------------------------
> # Calculating Leverages
> # ------------------------------------------------------------------

> # Generating Design Matrix
> X <-model.matrix(community_mdl_s8_AIC)

> #Finding row count and parameter count
> n <-dim(X)[1]
> p <-dim(X)[2]
```

```
> #Identifying  high leverage observations by hand.
> hatmat <- X%*%solve(t(X)%*%X)%*%t(X)
> community_ds_leverages <- diag(hatmat)
> community_ds[which(community_ds_leverages > 2*p/n),] -> comm_ds_influencers
> glimpse(comm_ds_influencers)

Observations: 118
Variables: 147
$ communityname          <fct> BerkeleyHeightstownship, Glendalecity, Arlington...
$ state                  <fct> NJ, CA, TX, MN, NY, AR, PA, MS, TX, TN, MI, NJ, ...
$ countyCode             <int> 39, NA, NA, 53, 57, NA, 101, NA, NA, NA, 161, 3,...
$ communityCode          <int> 5320, NA, NA, 51730, 2066, NA, 60000, NA, NA, NA...
$ fold                   <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
$ population             <int> 11980, 180038, 261721, 50889, 20714, 18540, 1585...
$ householdsize          <dbl> 3.10, 2.62, 2.60, 2.77, 2.36, 2.49, 2.63, 2.92, ...
$ racepctblack           <dbl> 1.37, 1.30, 8.41, 1.61, 1.46, 0.09, 39.86, 17.14...
$ racePctWhite           <dbl> 91.78, 74.02, 82.64, 95.66, 93.15, 99.39, 53.52,...
$ racePctAsian           <dbl> 6.50, 14.14, 3.92, 2.04, 0.56, 0.18, 2.74, 0.92,...
$ racePctHisp            <dbl> 1.88, 20.96, 8.91, 1.02, 11.61, 0.51, 5.63, 0.48...
$ agePct12t21            <dbl> 12.47, 12.04, 14.18, 13.13, 11.35, 14.13, 13.92,...
$ agePct12t29            <dbl> 21.44, 26.68, 32.78, 26.94, 23.13, 25.52, 28.02,...
$ agePct16t24            <dbl> 10.93, 12.37, 15.14, 12.19, 11.19, 12.37, 14.12,...
$ agePct65up             <dbl> 11.33, 11.54, 4.58, 4.77, 21.15, 15.18, 13.74, 7...
$ numbUrban              <int> 11980, 180038, 261763, 50889, 0, 0, 1585577, 218...
$ pctUrban               <dbl> 100.00, 100.00, 100.00, 100.00, 0.00, 0.00, 100....
$ medIncome              <int> 75122, 34372, 35048, 51314, 22166, 19985, 24603,...
$ pctWWage               <dbl> 89.24, 76.17, 90.25, 92.25, 65.00, 69.56, 70.12,...
$ pctWFarmSelf           <dbl> 1.55, 0.62, 0.81, 1.19, 0.32, 1.62, 0.35, 1.00, ...
$ pctWInvInc             <dbl> 70.20, 40.04, 39.01, 58.68, 47.13, 33.14, 32.15,...
$ pctWSocSec             <dbl> 23.62, 20.30, 11.05, 11.08, 43.79, 34.55, 31.63,...
$ pctWPubAsst            <dbl> 1.03, 11.27, 2.71, 2.41, 7.08, 8.77, 13.98, 3.80...
$ pctWRetire             <dbl> 18.39, 11.51, 9.12, 7.95, 21.60, 12.60, 18.20, 1...
$ medFamInc              <int> 79584, 39652, 41620, 59421, 29893, 24252, 30140,...
$ perCapInc              <int> 29711, 17966, 16239, 21908, 11991, 10444, 12091,...
$ whitePerCap            <int> 30233, 19362, 17420, 22294, 12228, 10448, 15027,...
$ blackPerCap            <int> 13600, 17693, 11615, 16470, 15778, 10101, 9061, ...
$ indianPerCap           <int> 5725, 20931, 13416, 10181, 0, 10092, 10146, 0, 1...
$ AsianPerCap            <int> 27101, 16696, 11175, 14316, 15243, 0, 8285, 1598...
$ OtherPerCap            <int> 5115, 9438, 7804, 1754, 6326, 7652, 5083, 10491,...
$ HispPerCap             <int> 22838, 11182, 9045, 9794, 7427, 9321, 6053, 8881...
$ NumUnderPov            <int> 227, 25484, 21272, 1681, 2756, 3398, 313374, 197...
$ PctPopUnderPov         <dbl> 1.96, 14.37, 8.21, 3.36, 13.50, 18.82, 20.27, 9....
$ PctLess9thGrade        <dbl> 5.81, 11.54, 3.71, 1.57, 12.54, 20.78, 11.29, 4....
$ PctNotHSGrad           <dbl> 9.90, 22.83, 12.19, 5.56, 31.33, 39.87, 35.69, 1...
$ PctBSorMore            <dbl> 48.18, 28.55, 29.99, 41.39, 12.33, 11.26, 15.22,...
$ PctUnemployed          <dbl> 2.70, 6.95, 4.99, 3.24, 8.97, 7.01, 9.62, 3.37, ...
$ PctEmploy              <dbl> 64.55, 60.04, 74.65, 78.05, 52.65, 55.48, 52.77,...
$ PctEmplManu            <dbl> 14.65, 14.43, 18.91, 19.44, 23.60, 33.52, 13.58,...
$ PctEmplProfServ        <dbl> 28.82, 24.78, 18.97, 22.51, 25.83, 19.75, 29.55,...
$ PctOccupManu           <dbl> 5.49, 9.07, 10.60, 7.08, 19.07, 27.97, 14.13, 7....
$ PctOccupMgmtProf       <dbl> 50.73, 33.09, 30.91, 39.55, 22.91, 17.54, 24.81,...
$ MalePctDivorce         <dbl> 3.67, 8.68, 10.73, 7.99, 9.34, 11.78, 10.62, 6.5...
$ MalePctNevMarr         <dbl> 26.38, 34.45, 30.21, 27.64, 29.70, 19.43, 42.11,...
$ FemalePctDiv           <dbl> 5.22, 13.33, 14.79, 10.17, 11.80, 13.59, 14.12, ...
$ TotalPctDiv            <dbl> 4.47, 11.12, 12.77, 9.09, 10.70, 12.75, 12.53, 9...
$ PersPerFam             <dbl> 3.22, 3.22, 3.14, 3.14, 2.96, 2.92, 3.26, 3.21, ...
$ PctFam2Par             <dbl> 91.43, 77.10, 78.19, 85.61, 65.74, 76.53, 53.45,...
$ PctKids2Par            <dbl> 90.17, 74.78, 75.99, 86.12, 62.69, 72.93, 47.22,...
$ PctYoungKids2Par       <dbl> 95.78, 86.01, 87.64, 93.55, 67.57, 77.38, 62.33,...
$ PctTeen2Par            <dbl> 95.81, 79.01, 76.69, 87.45, 70.40, 75.02, 56.66,...
$ PctWorkMomYoungKids    <dbl> 44.56, 52.45, 63.48, 66.82, 59.49, 69.52, 54.09,...
$ PctWorkMom             <dbl> 58.88, 59.33, 71.53, 73.32, 69.23, 73.04, 62.23,...
```

```
$ NumKidsBornNeverMar   <int> 31, 3034, 2898, 407, 529, 112, 138864, 567, 1198...
$ PctKidsBornNeverMar   <dbl> 0.36, 2.62, 1.89, 1.31, 3.28, 0.67, 11.53, 3.90,...
$ NumImmig              <int> 1277, 81352, 20006, 1330, 1250, 49, 104814, 389,...
$ PctImmigRecent        <dbl> 8.69, 30.33, 19.03, 13.98, 10.56, 0.00, 16.38, 3...
$ PctImmigRec5          <dbl> 13.00, 41.41, 30.13, 22.48, 18.96, 0.00, 24.30, ...
$ PctImmigRec8          <dbl> 20.99, 50.77, 44.66, 28.95, 21.60, 0.00, 33.12, ...
$ PctImmigRec10         <dbl> 30.93, 60.40, 56.86, 36.92, 25.28, 0.00, 42.58, ...
$ PctRecentImmig        <dbl> 0.93, 13.71, 1.45, 0.37, 0.64, 0.00, 1.08, 0.71,...
$ PctRecImmig5          <dbl> 1.39, 18.71, 2.30, 0.59, 1.14, 0.00, 1.61, 0.96,...
$ PctRecImmig8          <dbl> 2.24, 22.94, 3.41, 0.76, 1.30, 0.00, 2.19, 0.96,...
$ PctRecImmig10         <dbl> 3.30, 27.29, 4.35, 0.96, 1.53, 0.00, 2.81, 1.04,...
$ PctSpeakEnglOnly      <dbl> 85.68, 46.84, 87.44, 94.19, 78.98, 98.98, 86.31,...
$ PctNotSpeakEnglWell   <dbl> 1.37, 15.46, 2.20, 0.56, 2.49, 0.28, 2.63, 0.35,...
$ PctLargHouseFam       <dbl> 4.81, 6.98, 4.07, 3.30, 4.07, 2.28, 8.48, 3.92, ...
$ PctLargHouseOccup     <dbl> 4.17, 4.60, 2.80, 2.51, 2.62, 1.67, 5.42, 3.04, ...
$ PersPerOccupHous      <dbl> 2.99, 2.59, 2.58, 2.72, 2.33, 2.44, 2.56, 2.78, ...
$ PersPerOwnOccHous     <dbl> 3.00, 2.63, 2.89, 2.95, 2.45, 2.49, 2.75, 2.95, ...
$ PersPerRentOccHous    <dbl> 2.84, 2.56, 2.25, 2.08, 2.21, 2.34, 2.25, 2.41, ...
$ PctPersOwnOccup       <dbl> 91.46, 39.38, 58.00, 79.90, 53.56, 68.90, 66.55,...
$ PctPersDenseHous      <dbl> 0.39, 18.31, 4.95, 1.21, 1.79, 1.88, 4.69, 2.99,...
$ PctHousLess3BR        <dbl> 11.06, 75.17, 47.54, 40.32, 36.94, 43.88, 42.28,...
$ MedNumBR              <int> 3, 2, 3, 3, 3, 3, 3, 3, 3, 2, 2, 3, 2, 3, 2, 3, ...
$ HousVacant            <int> 64, 3510, 12116, 1255, 713, 472, 71824, 427, 339...
$ PctHousOccup          <dbl> 98.37, 95.13, 89.26, 93.60, 92.49, 94.03, 89.36,...
$ PctHousOwnOcc         <dbl> 91.01, 38.71, 51.84, 73.63, 50.94, 67.53, 61.95,...
$ PctVacantBoarded      <dbl> 3.12, 2.25, 1.81, 0.08, 2.10, 4.66, 21.96, 0.94,...
$ PctVacMore6Mos        <dbl> 37.50, 17.69, 27.45, 14.50, 59.05, 35.38, 59.26,...
$ MedYrHousBuilt        <int> 1959, 1962, 1978, 1979, 1939, 1972, 1939, 1976, ...
$ PctHousNoPhone        <dbl> 0.00, 1.53, 5.50, 0.29, 6.72, 10.95, 4.25, 2.60,...
$ PctWOFullPlumb        <dbl> 0.28, 0.46, 0.14, 0.14, 0.66, 0.51, 0.88, 0.37, ...
$ OwnOccLowQuart        <int> 215900, 249800, 64100, 97700, 53400, 30500, 2790...
$ OwnOccMedVal          <int> 262600, 343600, 82800, 127400, 71000, 41800, 494...
$ OwnOccHiQuart         <int> 326900, 469800, 110200, 167500, 91900, 60000, 77...
$ OwnOccQrange          <int> 111000, 220000, 46100, 69800, 38500, 29500, 4920...
$ RentLowQ              <int> 685, 511, 312, 506, 172, 153, 252, 292, 334, 187...
$ RentMedian            <int> 1001, 626, 382, 578, 240, 216, 358, 350, 397, 26...
$ RentHighQ             <int> 1001, 763, 469, 663, 303, 274, 483, 413, 483, 34...
$ RentQrange            <int> 316, 252, 157, 157, 131, 121, 231, 121, 149, 156...
$ MedRent               <int> 1001, 688, 444, 611, 337, 280, 452, 438, 469, 33...
$ MedRentPctHousInc     <dbl> 23.8, 30.5, 23.7, 22.9, 24.7, 26.0, 29.8, 25.6, ...
$ MedOwnCostPctInc      <dbl> 21.1, 25.3, 22.0, 21.7, 18.6, 17.5, 19.3, 18.1, ...
$ MedOwnCostPctIncNoMtg <dbl> 14.0, 11.6, 11.8, 11.5, 14.5, 12.3, 14.8, 12.3, ...
$ NumInShelters         <int> 11, 82, 47, 0, 2, 7, 3416, 0, 0, 354, 0, 5, 1344...
$ NumStreet             <int> 0, 17, 1, 0, 0, 0, 1069, 0, 4, 130, 0, 0, 16, 1,...
$ PctForeignBorn        <dbl> 10.66, 45.19, 7.64, 2.61, 6.03, 0.26, 6.61, 1.78...
$ PctBornSameState      <dbl> 53.72, 29.68, 51.54, 63.24, 81.86, 73.80, 73.58,...
$ PctSameHouse85        <dbl> 65.29, 37.69, 36.76, 43.01, 61.42, 52.41, 64.33,...
$ PctSameCity85         <dbl> 78.09, 73.29, 67.01, 76.47, 85.56, 83.68, 88.86,...
$ PctSameState85        <dbl> 89.14, 76.85, 84.09, 85.75, 93.76, 89.62, 92.65,...
$ LemasSwornFT          <int> NA, 204, 356, NA, NA, NA, 6523, NA, 160, 302, NA...
$ LemasSwFTPerPop       <dbl> NA, 113.73, 121.78, NA, NA, NA, 426.38, NA, 138....
$ LemasSwFTFieldOps     <int> NA, 189, 330, NA, NA, NA, 6519, NA, 138, 281, NA...
$ LemasSwFTFieldPerPop  <dbl> NA, 105.36, 112.89, NA, NA, NA, 426.12, NA, 119....
$ LemasTotalReq         <int> NA, 254080, 350000, NA, NA, NA, 5480855, NA, 734...
$ LemasTotReqPerPop     <dbl> NA, 141645.0, 119730.2, NA, NA, NA, 358261.4, NA...
$ PolicReqPerOffic      <dbl> NA, 1245.5, 983.1, NA, NA, NA, 840.2, NA, 459.1,...
$ PolicPerPop           <dbl> NA, 113.7, 121.8, NA, NA, NA, 426.4, NA, 138.2, ...
$ RacialMatchCommPol    <dbl> NA, 92.47, 95.56, NA, NA, NA, 79.54, NA, 92.83, ...
$ PctPolicWhite         <dbl> NA, 78.43, 87.08, NA, NA, NA, 73.48, NA, 92.50, ...
$ PctPolicBlack         <dbl> NA, 3.92, 6.74, NA, NA, NA, 23.18, NA, 2.50, 4.6...
$ PctPolicHisp          <dbl> NA, 11.76, 6.18, NA, NA, NA, 2.84, NA, 3.12, 0.0...
$ PctPolicAsian         <dbl> NA, 5.39, 0.00, NA, NA, NA, 0.00, NA, 0.00, 0.00...
$ PctPolicMinor         <dbl> NA, 21.08, 12.92, NA, NA, NA, 25.95, NA, 5.62, 4...
```

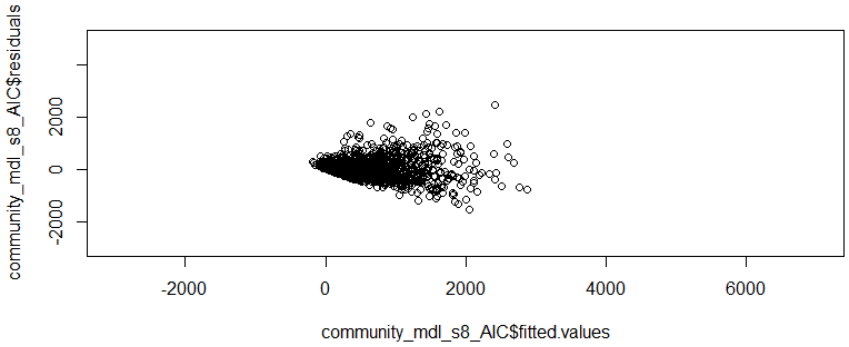
```
$ OfficAssgnDrugUnits   <int> NA, 12, 28, NA, NA, NA, 273, NA, 9, 31, NA, NA, ...
$ NumKindsDrugsSeiz     <int> NA, 11, 10, NA, NA, NA, 14, NA, 8, 8, NA, NA, 10...
$ PolicAveOTWorked      <dbl> NA, 16.9, 42.1, NA, NA, NA, 39.9, NA, 8.1, 41.3,...
$ LandArea              <dbl> 6.5, 31.7, 96.4, 34.1, 6.2, 30.1, 140.0, 24.7, 4...
$ PopDens               <dbl> 1845.9, 5677.3, 2716.3, 1491.7, 3353.7, 616.9, 1...
$ PctUsePubTrans        <dbl> 9.63, 4.15, 0.14, 2.64, 2.60, 0.23, 29.31, 0.24,...
$ PolicCars             <int> NA, 86, 98, NA, NA, NA, 822, NA, 127, 328, NA, N...
$ PolicOperBudg         <int> NA, 20900410, 20195376, NA, NA, NA, 287578496, N...
$ LemasPctPolicOnPatr   <dbl> NA, 92.65, 92.70, NA, NA, NA, 99.94, NA, 86.25, ...
$ LemasGangUnitDeploy   <int> NA, 5, 10, NA, NA, NA, 0, NA, 0, 0, NA, NA, 5, N...
$ LemasPctOfficDrugUn   <dbl> 0.00, 5.88, 7.87, 0.00, 0.00, 0.00, 4.19, 0.00, ...
$ PolicBudgPerPop       <dbl> NA, 116516.0, 69085.6, NA, NA, NA, 187978.5, NA,...
$ murders               <int> 0, 9, 7, 0, 0, 1, 439, 2, 5, 14, 0, 0, 22, 3, 1,...
$ murdPerPop            <dbl> 0.00, 5.02, 2.39, 0.00, 0.00, 4.65, 28.70, 8.76,...
$ rapes                 <int> 0, 30, 146, NA, 3, 9, 785, 0, 33, 102, NA, 2, 13...
$ rapesPerPop           <dbl> 0.00, 16.72, 49.94, NA, 14.98, 41.81, 51.31, 0.0...
$ robberies             <int> 1, 355, 710, 13, 3, 2, 11531, 14, 137, 596, 43, ...
$ robbbPerPop           <dbl> 8.20, 197.91, 242.88, 21.41, 14.98, 9.29, 753.74...
$ assaults              <int> 4, 277, 1396, 55, 4, 8, 6821, 12, 481, 2200, 50,...
$ assaultPerPop         <dbl> 32.81, 154.42, 477.55, 90.60, 19.97, 37.16, 445....
$ burglaries            <int> 14, 1596, 3977, 271, 128, 134, 15117, 169, NA, 2...
$ burglPerPop           <dbl> 114.85, 889.74, 1360.48, 446.39, 639.11, 622.47,...
$ larcenies             <int> 138, 4501, 11514, 1468, 314, 502, 39181, 435, 51...
$ larcPerPop            <dbl> 1132.08, 2509.23, 3938.78, 2418.09, 1567.81, 233...
$ autoTheft             <int> 16, 1447, 2452, 100, 17, 35, 23785, 12, 761, 160...
$ autoTheftPerPop       <dbl> 131.26, 806.68, 838.80, 164.72, 84.88, 162.59, 1...
$ arsons                <int> 2, 73, 97, 62, NA, 1, 2282, 5, 93, 173, 7, 2, 13...
$ arsonsPerPop          <dbl> 16.41, 40.70, 33.18, 102.13, NA, 4.65, 149.17, 2...
$ ViolentCrimesPerPop   <dbl> 41.02, 374.07, 772.77, NA, 49.93, 92.91, 1279.60...
$ nonViolPerPop         <dbl> 1394.59, 4246.34, 6171.23, 3131.33, NA, 3121.66,...

> # ------------------------------------------------------------------------
> # Collecting Outliers
> # ------------------------------------------------------------------------

> rs_comm <- data.frame(round(rstandard(community_mdl_s8_AIC),4))
> community_ds[which(abs(rs_comm) > 3),] ->community_ds_outliers

> glimpse(community_ds_outliers)
Observations: 29
Variables: 147
$ communityname         <fct> Rogerscity, GrandForkscity, Woostercity, Aberdee...
$ state                 <fct> AR, ND, OH, WA, CO, OH, MA, NJ, UT, OH, CA, WI, ...
$ countyCode            <int> NA, 35, 169, NA, NA, 165, 21, 35, NA, NA, NA, 13...
$ communityCode         <int> NA, 32060, 86548, NA, NA, 42364, 67945, 53280, N...
$ fold                  <int> 1, 1, 1, 1, 1, 2, 3, 3, 3, 3, 3, 3, 4, 5, 5, ...
$ population            <int> 24692, 49425, 22191, 16565, 10362, 10453, 26777,...
$ householdsize         <dbl> 2.54, 2.67, 2.58, 2.38, 2.40, 2.62, 2.85, 2.56, ...
$ racepctblack          <dbl> 0.06, 0.80, 3.49, 0.29, 0.18, 3.20, 4.00, 5.19, ...
$ racePctWhite          <dbl> 97.72, 95.49, 95.03, 93.76, 93.51, 95.56, 94.20,...
$ racePctAsian          <dbl> 0.77, 1.07, 1.17, 1.95, 0.23, 0.72, 1.11, 3.30, ...
$ racePctHisp           <dbl> 1.86, 1.19, 0.63, 2.55, 10.62, 0.47, 1.79, 12.01...
$ agePct12t21           <dbl> 12.84, 20.60, 18.31, 13.22, 15.23, 12.92, 13.21,...
$ agePct12t29           <dbl> 25.81, 38.87, 29.70, 24.76, 26.13, 27.69, 27.00,...
$ agePct16t24           <dbl> 11.93, 24.10, 18.07, 12.02, 13.20, 13.22, 13.11,...
$ agePct65up            <dbl> 14.62, 7.87, 12.58, 14.92, 15.01, 11.22, 11.96, ...
$ numbUrban             <int> 0, 49425, 0, 0, 0, 0, 25039, 18820, 86830, 39729...
$ pctUrban              <dbl> 0.00, 100.00, 0.00, 0.00, 0.00, 0.00, 93.51, 100...
$ medIncome             <int> 26198, 25456, 27148, 21762, 20189, 27095, 42044,...
$ pctWWage              <dbl> 73.97, 83.71, 77.40, 68.99, 70.72, 79.84, 83.33,...
$ pctWFarmSelf          <dbl> 1.36, 2.25, 0.45, 0.10, 3.75, 2.10, 1.22, 0.72, ...
$ pctWInvInc            <dbl> 42.10, 45.03, 48.20, 34.71, 40.14, 39.75, 45.54,...
$ pctWSocSec            <dbl> 31.04, 18.90, 28.30, 30.27, 32.16, 24.81, 27.73,...
```

```
$ pctWPubAsst          <dbl> 3.92, 4.64, 7.74, 11.64, 7.10, 5.12, 5.09, 2.99,...
$ pctWRetire           <dbl> 16.09, 9.83, 16.65, 18.52, 12.59, 16.84, 15.91, ...
$ medFamInc            <int> 31007, 32417, 32801, 28750, 24650, 31528, 47492,...
$ perCapInc            <int> 12779, 11902, 14283, 11816, 10189, 12464, 17313,...
$ whitePerCap          <int> 12784, 12096, 14618, 12066, 10577, 12614, 17160,...
$ blackPerCap          <int> 0, 10578, 8449, 1325, 18912, 8284, 17533, 17052,...
$ indianPerCap         <int> 13934, 6096, 13846, 6024, 9233, 12930, 9000, 139...
$ AsianPerCap          <int> 12182, 9944, 9267, 10750, 4133, 13034, 28003, 17...
$ OtherPerCap          <int> 9809, 5528, 4923, 8985, 4017, 3161, 24531, 10276...
$ HispPerCap           <int> 9511, 6559, 17914, 9490, 4789, 2993, 8328, 11168...
$ NumUnderPov          <int> 2120, 6526, 2603, 2965, 1691, 881, 1261, 685, 23...
$ PctPopUnderPov       <dbl> 8.72, 14.48, 12.69, 18.30, 16.91, 8.65, 4.78, 3....
$ PctLess9thGrade      <dbl> 8.18, 7.24, 6.41, 9.02, 9.71, 10.83, 7.57, 7.70,...
$ PctNotHSGrad         <dbl> 22.43, 14.24, 20.68, 26.57, 22.38, 25.00, 17.46,...
$ PctBSorMore          <dbl> 15.11, 29.28, 23.99, 12.41, 13.15, 16.09, 22.60,...
$ PctUnemployed        <dbl> 3.31, 5.08, 5.93, 10.96, 4.52, 1.93, 5.81, 3.76,...
$ PctEmploy            <dbl> 63.10, 66.75, 60.43, 49.82, 60.00, 64.82, 66.67,...
$ PctEmplManu          <dbl> 27.42, 6.38, 25.26, 24.97, 7.83, 17.82, 18.56, 1...
$ PctEmplProfServ      <dbl> 16.18, 35.22, 28.65, 21.11, 27.05, 21.10, 21.92,...
$ PctOccupManu         <dbl> 22.66, 11.28, 18.99, 21.09, 16.94, 15.17, 14.48,...
$ PctOccupMgmtProf     <dbl> 21.56, 30.59, 26.89, 21.68, 17.79, 24.30, 27.38,...
$ MalePctDivorce       <dbl> 9.43, 6.77, 9.29, 14.49, 10.06, 10.67, 6.47, 8.2...
$ MalePctNevMarr       <dbl> 20.49, 43.87, 32.73, 28.28, 27.09, 21.87, 32.66,...
$ FemalePctDiv         <dbl> 13.38, 9.78, 13.31, 15.93, 11.92, 15.60, 9.46, 1...
$ TotalPctDiv          <dbl> 11.54, 8.28, 11.46, 15.23, 11.07, 13.33, 8.04, 9...
$ PersPerFam           <dbl> 2.96, 3.07, 2.97, 2.99, 2.99, 3.04, 3.29, 3.09, ...
$ PctFam2Par           <dbl> 78.73, 75.26, 73.14, 66.09, 73.18, 74.48, 82.62,...
$ PctKids2Par          <dbl> 76.61, 76.33, 71.98, 64.05, 73.72, 73.30, 79.22,...
$ PctYoungKids2Par     <dbl> 88.72, 83.86, 76.73, 64.12, 82.43, 87.29, 93.37,...
$ PctTeen2Par          <dbl> 85.57, 78.54, 77.00, 82.52, 74.24, 74.25, 80.21,...
$ PctWorkMomYoungKids  <dbl> 69.07, 73.58, 55.53, 39.70, 59.14, 61.07, 64.27,...
$ PctWorkMom           <dbl> 75.45, 77.17, 68.31, 52.28, 68.06, 69.82, 71.62,...
$ NumKidsBornNeverMar  <int> 173, 629, 343, 498, 175, 76, 256, 247, 478, 403,...
$ PctKidsBornNeverMar  <dbl> 0.89, 1.95, 2.05, 3.63, 1.85, 0.94, 1.25, 2.04, ...
$ NumImmig             <int> 354, 1112, 496, 669, 312, 84, 2993, 2708, 4715, ...
$ PctImmigRecent       <dbl> 18.93, 33.63, 29.44, 7.03, 14.10, 0.00, 6.82, 10...
$ PctImmigRec5         <dbl> 23.73, 42.90, 39.72, 14.20, 14.10, 14.29, 9.52, ...
$ PctImmigRec8         <dbl> 27.68, 48.38, 44.76, 26.31, 21.47, 23.81, 9.99, ...
$ PctImmigRec10        <dbl> 29.94, 54.23, 50.81, 28.85, 23.72, 45.24, 13.40,...
$ PctRecentImmig       <dbl> 0.27, 0.76, 0.66, 0.28, 0.42, 0.00, 0.76, 1.48, ...
$ PctRecImmig5         <dbl> 0.34, 0.97, 0.89, 0.57, 0.42, 0.11, 1.06, 2.58, ...
$ PctRecImmig8         <dbl> 0.40, 1.09, 1.00, 1.06, 0.65, 0.19, 1.12, 4.90, ...
$ PctRecImmig10        <dbl> 0.43, 1.22, 1.14, 1.17, 0.71, 0.36, 1.50, 6.16, ...
$ PctSpeakEnglOnly     <dbl> 95.90, 94.35, 96.01, 93.20, 91.54, 97.50, 84.21,...
$ PctNotSpeakEnglWell  <dbl> 0.75, 0.30, 0.19, 0.56, 0.66, 0.25, 3.16, 3.75, ...
$ PctLargHouseFam      <dbl> 3.36, 3.60, 3.40, 4.43, 3.78, 2.96, 5.70, 5.13, ...
$ PctLargHouseOccup    <dbl> 2.47, 2.32, 2.38, 2.70, 2.36, 2.15, 4.30, 3.57, ...
$ PersPerOccupHous     <dbl> 2.51, 2.43, 2.39, 2.34, 2.32, 2.56, 2.81, 2.54, ...
$ PersPerOwnOccHous    <dbl> 2.58, 2.84, 2.60, 2.47, 2.43, 2.70, 3.01, 2.77, ...
$ PersPerRentOccHous   <dbl> 2.38, 2.04, 2.10, 2.17, 2.15, 2.39, 2.25, 2.22, ...
$ PctPersOwnOccup      <dbl> 65.60, 56.93, 62.62, 60.23, 63.92, 57.73, 78.56,...
$ PctPersDenseHous     <dbl> 2.80, 1.98, 1.15, 2.73, 2.50, 2.30, 1.68, 3.96, ...
$ PctHousLess3BR       <dbl> 41.45, 57.07, 44.90, 52.64, 51.36, 52.22, 37.94,...
$ MedNumBR             <int> 3, 2, 3, 2, 2, 2, 3, 2, 2, 3, 3, 3, 2, 3, 3, 3, ...
$ HousVacant           <int> 586, 1058, 416, 618, 474, 129, 360, 443, 773, 99...
$ PctHousOccup         <dbl> 94.31, 94.60, 95.39, 91.84, 90.11, 96.87, 96.31,...
$ PctHousOwnOcc        <dbl> 63.80, 48.71, 57.48, 57.03, 61.01, 54.68, 73.20,...
$ PctVacantBoarded     <dbl> 0.34, 0.85, 2.40, 5.02, 4.01, 0.00, 2.22, 0.23, ...
$ PctVacMore6Mos       <dbl> 26.45, 20.51, 27.16, 43.53, 58.44, 30.23, 36.11,...
$ MedYrHousBuilt       <int> 1976, 1968, 1959, 1942, 1955, 1965, 1965, 1951, ...
$ PctHousNoPhone       <dbl> 8.04, 2.77, 4.35, 7.90, 7.92, 6.96, 1.34, 1.61, ...
$ PctWOFullPlumb       <dbl> 0.40, 0.45, 0.38, 0.44, 0.48, 0.17, 0.15, 0.23, ...
$ OwnOccLowQuart       <int> 44800, 50800, 51700, 31200, 28700, 57200, 132500...
```

```
$ OwnOccMedVal          <int> 60300, 64700, 70300, 42200, 40200, 69800, 156300...
$ OwnOccHiQuart         <int> 75500, 81800, 95900, 59900, 56800, 93100, 184600...
$ OwnOccQrange          <int> 30700, 31000, 44200, 28700, 28100, 35900, 52100,...
$ RentLowQ              <int> 252, 234, 221, 177, 155, 272, 397, 522, 229, 380...
$ RentMedian            <int> 302, 320, 293, 249, 216, 333, 567, 615, 289, 438...
$ RentHighQ             <int> 373, 391, 361, 301, 297, 387, 689, 705, 424, 511...
$ RentQrange            <int> 121, 157, 140, 124, 142, 115, 292, 183, 195, 131...
$ MedRent               <int> 389, 367, 363, 314, 273, 410, 633, 678, 336, 509...
$ MedRentPctHousInc     <dbl> 24.8, 26.1, 23.6, 27.4, 23.5, 23.5, 24.7, 25.7, ...
$ MedOwnCostPctInc      <dbl> 19.6, 20.5, 18.8, 17.3, 20.4, 20.1, 22.2, 23.9, ...
$ MedOwnCostPctIncNoMtg <dbl> 11.7, 12.1, 12.1, 12.4, 13.4, 11.1, 13.2, 16.3, ...
$ NumInShelters         <int> 0, 70, 0, 21, 0, 0, 0, 9, 65, 0, 0, 0, 0, 0, 0, ...
$ NumStreet             <int> 0, 0, 0, 0, 0, 0, 0, 0, 9, 0, 2, 0, 0, 0, 0, 0, ...
$ PctForeignBorn        <dbl> 1.43, 2.25, 2.24, 4.04, 3.01, 0.80, 11.18, 14.39...
$ PctBornSameState      <dbl> 40.40, 62.70, 69.56, 64.10, 58.42, 75.12, 76.28,...
$ PctSameHouse85        <dbl> 39.82, 41.07, 45.18, 50.82, 52.32, 42.98, 65.39,...
$ PctSameCity85         <dbl> 70.19, 66.59, 76.13, 83.10, 78.65, 79.58, 82.69,...
$ PctSameState85        <dbl> 77.43, 80.46, 86.81, 90.53, 90.47, 91.81, 95.13,...
$ LemasSwornFT          <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ LemasSwFTPerPop       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ LemasSwFTFieldOps     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ LemasSwFTFieldPerPop  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ LemasTotalReq         <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ LemasTotReqPerPop     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ PolicReqPerOffic      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ PolicPerPop           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ RacialMatchCommPol    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ PctPolicWhite         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ PctPolicBlack         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ PctPolicHisp          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ PctPolicAsian         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ PctPolicMinor         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ OfficAssgnDrugUnits   <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ NumKindsDrugsSeiz     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ PolicAveOTWorked      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ LandArea              <dbl> 23.0, 15.0, 12.2, 11.0, 5.6, 9.6, 16.6, 2.9, 40....
$ PopDens               <dbl> 1073.9, 3304.7, 1813.5, 1510.9, 1853.0, 1087.9, ...
$ PctUsePubTrans        <dbl> 0.00, 0.95, 0.91, 3.12, 0.40, 0.14, 5.98, 5.04, ...
$ PolicCars             <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ PolicOperBudg         <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ LemasPctPolicOnPatr   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ LemasGangUnitDeploy   <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ LemasPctOfficDrugUn   <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, ...
$ PolicBudgPerPop       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
$ murders               <int> 0, 0, 0, 1, 0, 0, 0, 0, 1, 2, 2, 0, 0, 3, 0, 0, ...
$ murdPerPop            <dbl> 0.00, 0.00, 0.00, 5.84, 0.00, 0.00, 0.00, 0.00, ...
$ rapes                 <int> 9, 19, 8, 28, 2, 20, 7, 2, 36, 8, 4, 1, 4, 13, 0...
$ rapesPerPop           <dbl> 29.18, 37.70, 35.33, 163.47, 18.94, 170.98, 25.3...
$ robberies             <int> 10, 14, 7, 20, 1, 9, 19, 30, 18, 24, 96, 0, 3, 3...
$ robbbPerPop           <dbl> 32.42, 27.78, 30.91, 116.76, 9.47, 76.94, 68.89,...
$ assaults              <int> 44, 28, 27, 41, 14, 17, 47, 54, 88, 162, 101, 5,...
$ assaultPerPop         <dbl> 142.64, 55.55, 119.24, 239.36, 132.60, 145.34, 1...
$ burglaries            <int> 233, 234, 147, 198, 67, 118, 177, 243, 508, 316,...
$ burglPerPop           <dbl> 755.34, 464.26, 649.18, 1155.93, 634.59, 1008.81...
$ larcenies             <int> 1215, 2202, 718, 1234, 374, 410, 328, 717, 2889,...
$ larcPerPop            <dbl> 3938.79, 4368.79, 3170.82, 7204.16, 3542.34, 350...
$ autoTheft             <int> 48, 195, 29, 42, 14, 23, 175, 148, 155, 116, 328...
$ autoTheftPerPop       <dbl> 155.61, 386.88, 128.07, 245.20, 132.60, 196.63, ...
$ arsons                <int> 6, 14, 7, 27, 2, 1, 2, 4, 10, 9, 21, 1, 1, 2, 0,...
$ arsonsPerPop          <dbl> 19.45, 27.78, 30.91, 157.63, 18.94, 8.55, 7.25, ...
$ ViolentCrimesPerPop   <dbl> 204.23, 121.02, 185.48, 525.42, 161.02, 393.26, ...
$ nonViolPerPop         <dbl> 4869.19, 5247.70, 3978.98, 8762.92, 4328.47, 471...
```

```
> #Calculating F-Value Threshold

> num_df <- p
> den_df <- n-p


> F_thresh <- qf(0.5, num_df,den_df)
> F_thresh
[1] 0.9273739

> #calculate Cook's distances
> comm_cd <- cooks.distance(community_mdl_s8_AIC)

> #Identifying the outliers
> community_outliers <- which(comm_cd > F_thresh)

> community_outliers
143

> #To check the influence of outlier, recreating the data-set minus the outlier
> community_ds2 <- community_ds[-community_outliers,]

> #Generating the model without the outlier
> community_mdl_y <- lm(final_formula, community_ds2)
> summary(community_mdl_s8_AIC)

Call:
lm(formula = final_formula)

Residuals:
     Min       1Q   Median       3Q      Max
-1564.71  -189.12   -36.95   120.44  2449.56

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          725.749454 175.840651   4.127 3.86e-05 ***
racepctblack           8.619682   1.195490   7.210 8.62e-13 ***
PctPersDenseHous      20.354836   1.922747  10.586  < 2e-16 ***
pctUrban               1.387255   0.225801   6.144 1.02e-09 ***
PctKidsBornNeverMar   70.231731   6.320571  11.112  < 2e-16 ***
HousVacant             0.008299   0.001520   5.461 5.49e-08 ***
pctWWage             -11.041426   1.714942  -6.438 1.60e-10 ***
MalePctDivorce        40.123894   3.866355  10.378  < 2e-16 ***
pctWRetire            -9.227563   2.739564  -3.368 0.000775 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 364.2 on 1586 degrees of freedom
Multiple R-squared:  0.6462,   Adjusted R-squared:  0.6444
F-statistic: 362.1 on 8 and 1586 DF,  p-value: < 2.2e-16

> summary(community_mdl_y)

Call:
lm(formula = final_formula, data = community_ds2)

Residuals:
     Min       1Q   Median       3Q      Max
-1547.75  -197.45   -37.71   117.52  2459.12

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)          693.188110 163.017453   4.252 2.21e-05 ***
```

```
racepctblack           9.589080   1.081792   8.864  < 2e-16 ***
PctPersDenseHous      20.324564   1.745254  11.646  < 2e-16 ***
pctUrban               1.500037   0.207428   7.232 6.78e-13 ***
PctKidsBornNeverMar   64.906234   5.771493  11.246  < 2e-16 ***
HousVacant             0.009287   0.001486   6.248 5.08e-10 ***
pctWWage             -11.011010   1.577901  -6.978 4.06e-12 ***
MalePctDivorce        41.641982   3.590091  11.599  < 2e-16 ***
pctWRetire            -8.090349   2.540083  -3.185  0.00147 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 373.6 on 1984 degrees of freedom
  (221 observations deleted due to missingness)
Multiple R-squared:  0.6324,   Adjusted R-squared:  0.6309
F-statistic: 426.6 on 8 and 1984 DF,  p-value: < 2.2e-16

#-------------------------------------------------------------------------------------------
# CONCLUSION: Removal of Outlier has had no significant impact on the Model hence outlier
will not be removed.
#-------------------------------------------------------------------------------------------

#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
# STAGE 9: MODEL TRANSFORMATION
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------

> plot(community_mdl_s8_AIC$fitted.values, community_mdl_s8_AIC$residuals, xlim = c(-3000
, 7000), ylim = c(-3000, 5000))
```

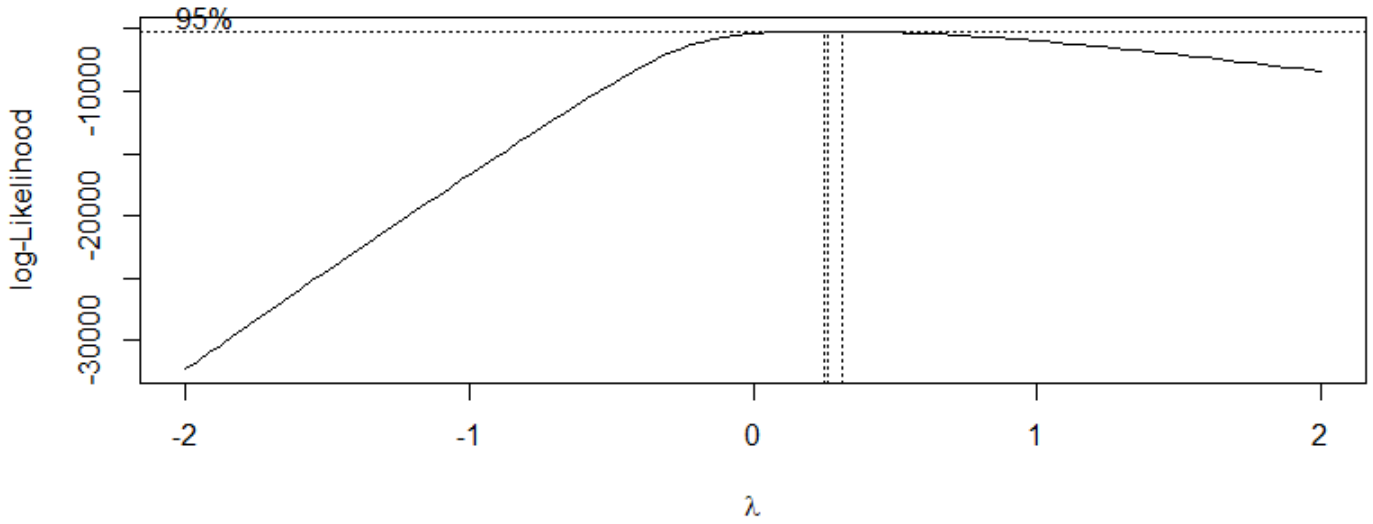



```
> #Try Box-Cox to remove heteroscedasticity
> bc <- boxcox(community_mdl_s8_AIC)
Error in boxcox.default(community_mdl_s8_AIC) : response variable must be positive

> # ------------------------------------------------
> # Varifying Response Variable. If it contains 0
> # then adding an offset to response variable
> # ------------------------------------------------

> summary(ViolentCrimesPerPop)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0   161.8   375.6   586.6   797.2  4877.1

> community_mdl_bc1 <- lm(ViolentCrimesPerPop + 0.01 ~ racepctblack + PctPersDenseHous +
pctUrban +   PctKidsBornNeverMar + HousVacant + pctWWage + MalePctDivorce + pctWRetire )

> summary(community_mdl_bc1)
```

```
Call:
lm(formula = ViolentCrimesPerPop + 0.01 ~ racepctblack + PctPersDenseHous +
    pctUrban + PctKidsBornNeverMar + HousVacant + pctWWage +
    MalePctDivorce + pctWRetire)

Residuals:
     Min       1Q   Median       3Q      Max
-1564.71  -189.12   -36.95   120.44  2449.56

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         725.759454 175.840651   4.127 3.86e-05 ***
racepctblack          8.619682   1.195490   7.210 8.62e-13 ***
PctPersDenseHous     20.354836   1.922747  10.586  < 2e-16 ***
pctUrban              1.387255   0.225801   6.144 1.02e-09 ***
PctKidsBornNeverMar  70.231731   6.320571  11.112  < 2e-16 ***
HousVacant            0.008299   0.001520   5.461 5.49e-08 ***
pctWWage            -11.041426   1.714942  -6.438 1.60e-10 ***
MalePctDivorce       40.123894   3.866355  10.378  < 2e-16 ***
pctWRetire           -9.227563   2.739564  -3.368 0.000775 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 364.2 on 1586 degrees of freedom
Multiple R-squared:  0.6462,   Adjusted R-squared:  0.6444
F-statistic: 362.1 on 8 and 1586 DF,  p-value: < 2.2e-16
```

```
> # ------------------------------------------------------------------------
> # Re-calculating the lambda
> # ------------------------------------------------------------------------
```

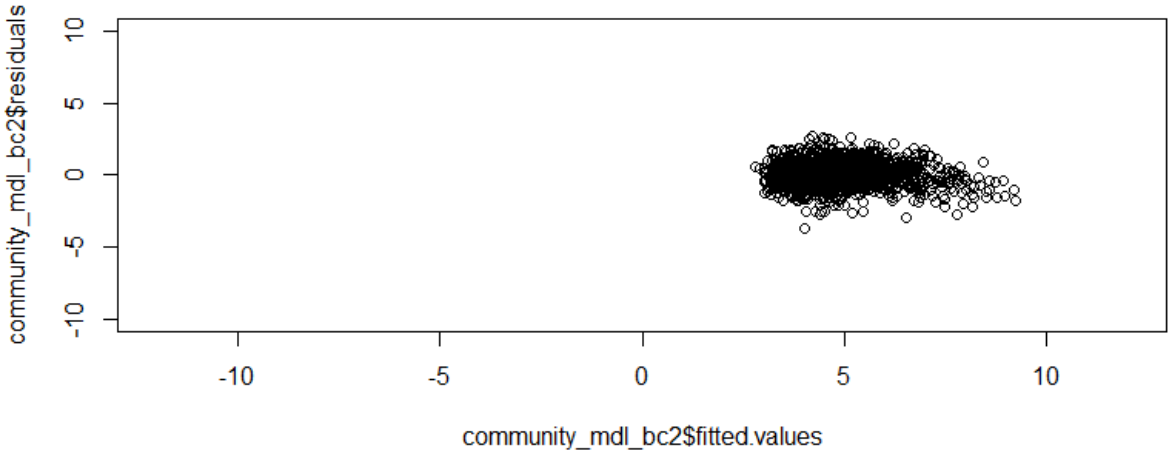
```
> bc <- boxcox(community_mdl_bc1)
```



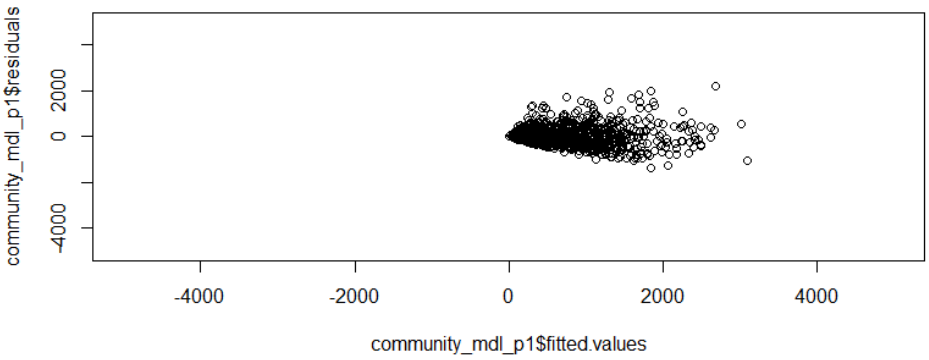

```
> lambda <- bc$x[which.max(bc$y)]
```

```
> lambda
[1] 0.3030303
```

```
> community_mdl_bc2 <- lm(((ViolentCrimesPerPop + 0.01)^lambda) ~ racepctblack + PctPersD
enseHous + pctUrban + PctKidsBornNeverMar + HousVacant + pctWWage + MalePctDivorce + pctW
Retire)

> plot(community_mdl_bc2$fitted.values, community_mdl_bc2$residuals, xlim = c(-12, 12), y
lim = c(-10, 10))
```




```
> bptest(community_mdl_s8_AIC)

        studentized Breusch-Pagan test

data:  community_mdl_s8_AIC
BP = 278.76, df = 8, p-value < 2.2e-16

> bptest(community_mdl_bc1)

        studentized Breusch-Pagan test

data:  community_mdl_bc1
BP = 278.76, df = 8, p-value < 2.2e-16

> bptest(community_mdl_bc2)

        studentized Breusch-Pagan test

data:  community_mdl_bc2
BP = 36.757, df = 8, p-value = 1.276e-05

> # ------------------------------------------------------------------------
> # Applying Polynomial Transformation
> # ------------------------------------------------------------------------

> community_mdl_p1 <- lm(ViolentCrimesPerPop ~ polym(racepctblack, PctPersDenseHous, pctU
rban, PctKidsBornNeverMar, HousVacant, pctWWage, MalePctDivorce, pctWRetire, degree = 2),
community_train_ds)

> plot(community_mdl_p1$fitted.values, community_mdl_p1$residuals, xlim = c(-5000, 5000),
+      ylim = c(-5000, 5000))
```



After studying all 3 models: OLS, BoxCox Transformed and Polynomial results and their residual vs fitted plots, CoxBox Transformation based plot seem to provide the best fitting linear model.

```
#------------------------------------------------------------------------------
#------------------------------------------------------------------------------
#------------------------------------------------------------------------------
# STAGE 10: INFERENCES AND PREDICTIONS
#------------------------------------------------------------------------------
#------------------------------------------------------------------------------
#------------------------------------------------------------------------------
```

```
> # ------------------------------------------------------------------
> # Week 7 Assignment Q1 & Q2:
> # ------------------------------------------------------------------
> a1 <- summary(community_mdl_bc2)
> a1

Call:
lm(formula = ((ViolentCrimesPerPop + 0.01)^lambda) ~ racepctblack +
    PctPersDenseHous + pctUrban + PctKidsBornNeverMar + HousVacant +
    pctWWage + MalePctDivorce + pctWRetire)

Residuals:
    Min      1Q  Median      3Q     Max
-4.7189 -0.7870 -0.0457  0.7369  5.4946

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          6.586e+00  5.619e-01  11.720  < 2e-16 ***
racepctblack         2.976e-02  3.820e-03   7.791 1.20e-14 ***
PctPersDenseHous     9.131e-02  6.144e-03  14.861  < 2e-16 ***
pctUrban             4.784e-03  7.216e-04   6.630 4.59e-11 ***
PctKidsBornNeverMar  1.427e-01  2.020e-02   7.066 2.37e-12 ***
HousVacant           1.879e-05  4.857e-06   3.868 0.000114 ***
pctWWage            -4.288e-02  5.480e-03  -7.825 9.20e-15 ***
MalePctDivorce       2.134e-01  1.236e-02  17.275  < 2e-16 ***
pctWRetire          -2.972e-02  8.755e-03  -3.394 0.000705 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.164 on 1586 degrees of freedom
Multiple R-squared:  0.6587,   Adjusted R-squared:  0.657
F-statistic: 382.7 on 8 and 1586 DF,  p-value: < 2.2e-16
```

```
> # ------------------------------------------------------------------
> # Week 7 Assignment Q3: Most Important Variable: PctPersDenseHou
```

```
> # ------------------------------------------------------------------

> a1$coefficients[3,]
    Estimate   Std. Error      t value     Pr(>|t|)
9.131342e-02 6.144488e-03 1.486103e+01 7.200200e-47

> v_coeff  <- a1$coefficients[3,1]
> v_stderr <- a1$coefficients[3,2]
> n <- dim(community_ds)[1]
> p <- length(community_mdl_bc2$coefficients)
> t_val <- qt(0.975, (n-p))

> v_coeff + (c(-1, 1) * t_val * v_stderr)
[1] 0.07926384 0.10336301

> # ------------------------------------------------------------------
> # Week 7 Assignment Q4: Compute and report a 95% confidence interval
> # for a prediction. In other words, choose particular values of your
> # predictors that are meaningful and compute a 95% confidence
> # interval for the predicted value of y at those values.
> # ------------------------------------------------------------------

> dim(community_test_ds)
[1] 399   9

> community_ds_median <- data.frame(
+   racepctblack = median(community_test_ds$racepctblack),
+   PctPersDenseHous = median(community_test_ds$PctPersDenseHous),
+   pctUrban = median(community_test_ds$pctUrban),
+   PctKidsBornNeverMar = median(community_test_ds$PctKidsBornNeverMar),
+   HousVacant = median(community_test_ds$HousVacant),
+   pctWWage = median(community_test_ds$pctWWage),
+   MalePctDivorce = median(community_test_ds$MalePctDivorce),
+   pctWRetire = median(community_test_ds$pctWRetire))

> predict(community_mdl_bc2, newdata = community_ds_median, interval = 'confidence')
       fit      lwr      upr
1 5.820758 5.734793 5.906722

> # ------------------------------------------------------------------
> # Week 7 Assignment Q5: Compute and report a 95% prediction interval
> # for a particular observation. Again, you'll choose particular values
> # of your predictors and compute prediction interval for those values.
> # ------------------------------------------------------------------

> community_ds_100 <- community_test_ds[100,]
> predict(community_mdl_bc2, newdata = community_ds_100, interval = 'confidence')
         fit      lwr      upr
624 5.634868 5.518743 5.750994
```