Linear Models Final Project

Hunter Kempf - East Section

5/8/2019

Contents

Introduction

This report will take you through the data science process of fitting a linear regression model from Exploratory Data Analysis to Predicting results for new data.

The dataset being used is the Communities and Crime Data Set from UCI. The data combines socio-economic data from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI UCR. More information about it can be found here

Using this data a linear model will be fit to best predict Violent Crimes per 100k people using all other useful columns and two variable selection methods. Some other investigations will be made to see possible transformations that will improve model performance or unusual points that should be removed to improve the model performance. This analysis only focuses on linear models and does not use any other more advanced modeling techniques such as Decision trees, Random Forrests or Neural Nets some or all of which may prove better at prediction than a linear model for this data.

Step 1: Exploratory Data Analysis

The first step in the data science process is to understand the data set you are working with. This section will include some plots and other information about the dataset to give a broad overview about the dataset and reasons for any restrictions to the dataset

```
## Observations: 1,994
## Variables: 128
## $ state                 <int> 8, 53, 24, 34, 42, 6, 44, 6, 21, 29, 6, ...
## $ county                <chr> "?", "?", "?", "5", "95", "?", "7", "?",...
## $ community             <chr> "?", "?", "?", "81440", "6096", "?", "41...
## $ communityname         <chr> "Lakewoodcity", "Tukwilacity", "Aberdeen...
## $ fold                  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ population            <dbl> 0.19, 0.00, 0.00, 0.04, 0.01, 0.02, 0.01...
## $ householdsize         <dbl> 0.33, 0.16, 0.42, 0.77, 0.55, 0.28, 0.39...
## $ racepctblack          <dbl> 0.02, 0.12, 0.49, 1.00, 0.02, 0.06, 0.00...
## $ racePctWhite          <dbl> 0.90, 0.74, 0.56, 0.08, 0.95, 0.54, 0.98...
## $ racePctAsian          <dbl> 0.12, 0.45, 0.17, 0.12, 0.09, 1.00, 0.06...
## $ racePctHisp           <dbl> 0.17, 0.07, 0.04, 0.10, 0.05, 0.25, 0.02...
## $ agePct12t21           <dbl> 0.34, 0.26, 0.39, 0.51, 0.38, 0.31, 0.30...
## $ agePct12t29           <dbl> 0.47, 0.59, 0.47, 0.50, 0.38, 0.48, 0.37...
## $ agePct16t24           <dbl> 0.29, 0.35, 0.28, 0.34, 0.23, 0.27, 0.23...
## $ agePct65up            <dbl> 0.32, 0.27, 0.32, 0.21, 0.36, 0.37, 0.60...
## $ numbUrban             <dbl> 0.20, 0.02, 0.00, 0.06, 0.02, 0.04, 0.02...
## $ pctUrban              <dbl> 1.00, 1.00, 0.00, 1.00, 0.90, 1.00, 0.81...
## $ medIncome             <dbl> 0.37, 0.31, 0.30, 0.58, 0.50, 0.52, 0.42...
## $ pctWWage              <dbl> 0.72, 0.72, 0.58, 0.89, 0.72, 0.68, 0.50...
## $ pctWFarmSelf          <dbl> 0.34, 0.11, 0.19, 0.21, 0.16, 0.20, 0.23...
## $ pctWInvInc            <dbl> 0.60, 0.45, 0.39, 0.43, 0.68, 0.61, 0.68...
## $ pctWSocSec            <dbl> 0.29, 0.25, 0.38, 0.36, 0.44, 0.28, 0.61...
## $ pctWPubAsst           <dbl> 0.15, 0.29, 0.40, 0.20, 0.11, 0.15, 0.21...
## $ pctWRetire            <dbl> 0.43, 0.39, 0.84, 0.82, 0.71, 0.25, 0.54...
## $ medFamInc             <dbl> 0.39, 0.29, 0.28, 0.51, 0.46, 0.62, 0.43...
## $ perCapInc             <dbl> 0.40, 0.37, 0.27, 0.36, 0.43, 0.72, 0.47...
## $ whitePerCap           <dbl> 0.39, 0.38, 0.29, 0.40, 0.41, 0.76, 0.44...
## $ blackPerCap           <dbl> 0.32, 0.33, 0.27, 0.39, 0.28, 0.77, 0.40...
## $ indianPerCap          <dbl> 0.27, 0.16, 0.07, 0.16, 0.00, 0.28, 0.24...
## $ AsianPerCap           <dbl> 0.27, 0.30, 0.29, 0.25, 0.74, 0.52, 0.86...
## $ OtherPerCap           <chr> "0.36", "0.22", "0.28", "0.36", "0.51", ...
## $ HispPerCap            <dbl> 0.41, 0.35, 0.39, 0.44, 0.48, 0.60, 0.36...
## $ NumUnderPov           <dbl> 0.08, 0.01, 0.01, 0.01, 0.00, 0.01, 0.01...
## $ PctPopUnderPov        <dbl> 0.19, 0.24, 0.27, 0.10, 0.06, 0.12, 0.11...
## $ PctLess9thGrade       <dbl> 0.10, 0.14, 0.27, 0.09, 0.25, 0.13, 0.29...
## $ PctNotHSGrad          <dbl> 0.18, 0.24, 0.43, 0.25, 0.30, 0.12, 0.41...
## $ PctBSorMore           <dbl> 0.48, 0.30, 0.19, 0.31, 0.33, 0.80, 0.36...
## $ PctUnemployed         <dbl> 0.27, 0.27, 0.36, 0.33, 0.12, 0.10, 0.28...
## $ PctEmploy             <dbl> 0.68, 0.73, 0.58, 0.71, 0.65, 0.65, 0.54...
## $ PctEmplManu           <dbl> 0.23, 0.57, 0.32, 0.36, 0.67, 0.19, 0.44...
## $ PctEmplProfServ       <dbl> 0.41, 0.15, 0.29, 0.45, 0.38, 0.77, 0.53...
## $ PctOccupManu          <dbl> 0.25, 0.42, 0.49, 0.37, 0.42, 0.06, 0.33...
## $ PctOccupMgmtProf      <dbl> 0.52, 0.36, 0.32, 0.39, 0.46, 0.91, 0.49...
## $ MalePctDivorce        <dbl> 0.68, 1.00, 0.63, 0.34, 0.22, 0.49, 0.25...
## $ MalePctNevMarr        <dbl> 0.40, 0.63, 0.41, 0.45, 0.27, 0.57, 0.34...
## $ FemalePctDiv          <dbl> 0.75, 0.91, 0.71, 0.49, 0.20, 0.61, 0.28...
```

```
## $ TotalPctDiv           <dbl> 0.75, 1.00, 0.70, 0.44, 0.21, 0.58, 0.28...
## $ PersPerFam            <dbl> 0.35, 0.29, 0.45, 0.75, 0.51, 0.44, 0.42...
## $ PctFam2Par            <dbl> 0.55, 0.43, 0.42, 0.65, 0.91, 0.62, 0.77...
## $ PctKids2Par           <dbl> 0.59, 0.47, 0.44, 0.54, 0.91, 0.69, 0.81...
## $ PctYoungKids2Par      <dbl> 0.61, 0.60, 0.43, 0.83, 0.89, 0.87, 0.79...
## $ PctTeen2Par           <dbl> 0.56, 0.39, 0.43, 0.65, 0.85, 0.53, 0.74...
## $ PctWorkMomYoungKids   <dbl> 0.74, 0.46, 0.71, 0.85, 0.40, 0.30, 0.57...
## $ PctWorkMom            <dbl> 0.76, 0.53, 0.67, 0.86, 0.60, 0.43, 0.62...
## $ NumIlleg              <dbl> 0.04, 0.00, 0.01, 0.03, 0.00, 0.00, 0.00...
## $ PctIlleg              <dbl> 0.14, 0.24, 0.46, 0.33, 0.06, 0.11, 0.13...
## $ NumImmig              <dbl> 0.03, 0.01, 0.00, 0.02, 0.00, 0.04, 0.01...
## $ PctImmigRecent        <dbl> 0.24, 0.52, 0.07, 0.11, 0.03, 0.30, 0.00...
## $ PctImmigRec5          <dbl> 0.27, 0.62, 0.06, 0.20, 0.07, 0.35, 0.02...
## $ PctImmigRec8          <dbl> 0.37, 0.64, 0.15, 0.30, 0.20, 0.43, 0.02...
## $ PctImmigRec10         <dbl> 0.39, 0.63, 0.19, 0.31, 0.27, 0.47, 0.10...
## $ PctRecentImmig        <dbl> 0.07, 0.25, 0.02, 0.05, 0.01, 0.50, 0.00...
## $ PctRecImmig5          <dbl> 0.07, 0.27, 0.02, 0.08, 0.02, 0.50, 0.01...
## $ PctRecImmig8          <dbl> 0.08, 0.25, 0.04, 0.11, 0.04, 0.56, 0.01...
## $ PctRecImmig10         <dbl> 0.08, 0.23, 0.05, 0.11, 0.05, 0.57, 0.03...
## $ PctSpeakEnglOnly      <dbl> 0.89, 0.84, 0.88, 0.81, 0.88, 0.45, 0.73...
## $ PctNotSpeakEnglWell   <dbl> 0.06, 0.10, 0.04, 0.08, 0.05, 0.28, 0.05...
## $ PctLargHouseFam       <dbl> 0.14, 0.16, 0.20, 0.56, 0.16, 0.25, 0.12...
## $ PctLargHouseOccup     <dbl> 0.13, 0.10, 0.20, 0.62, 0.19, 0.19, 0.13...
## $ PersPerOccupHous      <dbl> 0.33, 0.17, 0.46, 0.85, 0.59, 0.29, 0.42...
## $ PersPerOwnOccHous     <dbl> 0.39, 0.29, 0.52, 0.77, 0.60, 0.53, 0.54...
## $ PersPerRentOccHous    <dbl> 0.28, 0.17, 0.43, 1.00, 0.37, 0.18, 0.24...
## $ PctPersOwnOccup       <dbl> 0.55, 0.26, 0.42, 0.94, 0.89, 0.39, 0.65...
## $ PctPersDenseHous      <dbl> 0.09, 0.20, 0.15, 0.12, 0.02, 0.26, 0.03...
## $ PctHousLess3BR        <dbl> 0.51, 0.82, 0.51, 0.01, 0.19, 0.73, 0.46...
## $ MedNumBR              <dbl> 0.5, 0.0, 0.5, 0.5, 0.5, 0.0, 0.5, 0.0, ...
## $ HousVacant            <dbl> 0.21, 0.02, 0.01, 0.01, 0.01, 0.02, 0.01...
## $ PctHousOccup          <dbl> 0.71, 0.79, 0.86, 0.97, 0.89, 0.84, 0.89...
## $ PctHousOwnOcc         <dbl> 0.52, 0.24, 0.41, 0.96, 0.87, 0.30, 0.57...
## $ PctVacantBoarded      <dbl> 0.05, 0.02, 0.29, 0.60, 0.04, 0.16, 0.09...
## $ PctVacMore6Mos        <dbl> 0.26, 0.25, 0.30, 0.47, 0.55, 0.28, 0.49...
## $ MedYrHousBuilt        <dbl> 0.65, 0.65, 0.52, 0.52, 0.73, 0.25, 0.38...
## $ PctHousNoPhone        <dbl> 0.14, 0.16, 0.47, 0.11, 0.05, 0.02, 0.05...
## $ PctWOFullPlumb        <dbl> 0.06, 0.00, 0.45, 0.11, 0.14, 0.05, 0.05...
## $ OwnOccLowQuart        <dbl> 0.22, 0.21, 0.18, 0.24, 0.31, 0.94, 0.37...
## $ OwnOccMedVal          <dbl> 0.19, 0.20, 0.17, 0.21, 0.31, 1.00, 0.38...
## $ OwnOccHiQuart         <dbl> 0.18, 0.21, 0.16, 0.19, 0.30, 1.00, 0.39...
## $ RentLowQ              <dbl> 0.36, 0.42, 0.27, 0.75, 0.40, 0.67, 0.26...
## $ RentMedian            <dbl> 0.35, 0.38, 0.29, 0.70, 0.36, 0.63, 0.35...
## $ RentHighQ             <dbl> 0.38, 0.40, 0.27, 0.77, 0.38, 0.68, 0.42...
## $ MedRent               <dbl> 0.34, 0.37, 0.31, 0.89, 0.38, 0.62, 0.35...
## $ MedRentPctHousInc     <dbl> 0.38, 0.29, 0.48, 0.63, 0.22, 0.47, 0.46...
## $ MedOwnCostPctInc      <dbl> 0.46, 0.32, 0.39, 0.51, 0.51, 0.59, 0.44...
## $ MedOwnCostPctIncNoMtg <dbl> 0.25, 0.18, 0.28, 0.47, 0.21, 0.11, 0.31...
## $ NumInShelters         <dbl> 0.04, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00...
## $ NumStreet             <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00...
## $ PctForeignBorn        <dbl> 0.12, 0.21, 0.14, 0.19, 0.11, 0.70, 0.15...
## $ PctBornSameState      <dbl> 0.42, 0.50, 0.49, 0.30, 0.72, 0.42, 0.81...
## $ PctSameHouse85        <dbl> 0.50, 0.34, 0.54, 0.73, 0.64, 0.49, 0.77...
## $ PctSameCity85         <dbl> 0.51, 0.60, 0.67, 0.64, 0.61, 0.73, 0.91...
```

```
## $ PctSameState85        <dbl> 0.64, 0.52, 0.56, 0.65, 0.53, 0.64, 0.84...
## $ LemasSwornFT          <chr> "0.03", "?", "?", "?", "?", "?", "?", "?...
## $ LemasSwFTPerPop       <chr> "0.13", "?", "?", "?", "?", "?", "?", "?...
## $ LemasSwFTFieldOps     <chr> "0.96", "?", "?", "?", "?", "?", "?", "?...
## $ LemasSwFTFieldPerPop  <chr> "0.17", "?", "?", "?", "?", "?", "?", "?...
## $ LemasTotalReq         <chr> "0.06", "?", "?", "?", "?", "?", "?", "?...
## $ LemasTotReqPerPop     <chr> "0.18", "?", "?", "?", "?", "?", "?", "?...
## $ PolicReqPerOffic      <chr> "0.44", "?", "?", "?", "?", "?", "?", "?...
## $ PolicPerPop           <chr> "0.13", "?", "?", "?", "?", "?", "?", "?...
## $ RacialMatchCommPol    <chr> "0.94", "?", "?", "?", "?", "?", "?", "?...
## $ PctPolicWhite         <chr> "0.93", "?", "?", "?", "?", "?", "?", "?...
## $ PctPolicBlack         <chr> "0.03", "?", "?", "?", "?", "?", "?", "?...
## $ PctPolicHisp          <chr> "0.07", "?", "?", "?", "?", "?", "?", "?...
## $ PctPolicAsian         <chr> "0.1", "?", "?", "?", "?", "?", "?", "?"...
## $ PctPolicMinor         <chr> "0.07", "?", "?", "?", "?", "?", "?", "?...
## $ OfficAssgnDrugUnits   <chr> "0.02", "?", "?", "?", "?", "?", "?", "?...
## $ NumKindsDrugsSeiz     <chr> "0.57", "?", "?", "?", "?", "?", "?", "?...
## $ PolicAveOTWorked      <chr> "0.29", "?", "?", "?", "?", "?", "?", "?...
## $ LandArea              <dbl> 0.12, 0.02, 0.01, 0.02, 0.04, 0.01, 0.05...
## $ PopDens               <dbl> 0.26, 0.12, 0.21, 0.39, 0.09, 0.58, 0.08...
## $ PctUsePubTrans        <dbl> 0.20, 0.45, 0.02, 0.28, 0.02, 0.10, 0.06...
## $ PolicCars             <chr> "0.06", "?", "?", "?", "?", "?", "?", "?...
## $ PolicOperBudg         <chr> "0.04", "?", "?", "?", "?", "?", "?", "?...
## $ LemasPctPolicOnPatr   <chr> "0.9", "?", "?", "?", "?", "?", "?", "?"...
## $ LemasGangUnitDeploy   <chr> "0.5", "?", "?", "?", "?", "?", "?", "?"...
## $ LemasPctOfficDrugUn   <dbl> 0.32, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00...
## $ PolicBudgPerPop       <chr> "0.14", "?", "?", "?", "?", "?", "?", "?...
## $ ViolentCrimesPerPop   <dbl> 0.20, 0.67, 0.43, 0.12, 0.03, 0.14, 0.03...
```

Since many columns have missing values denoted as "?" we need to remove those and replace them with NA's that R can handle better. Also we will remove columns with a majority of values that are NA's or character columns that dont work well with linear models. A list of these removed columns, the number, percentage of NA values in them and their type is displayed below.

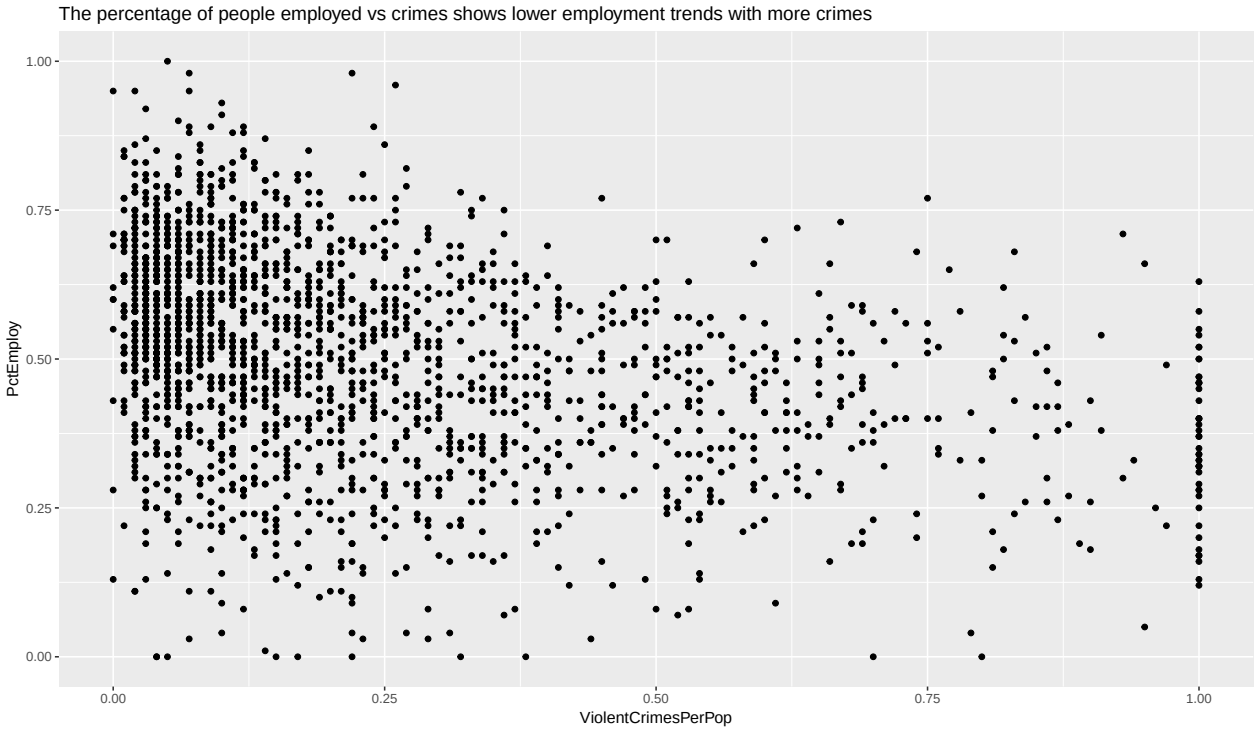
Removed Variables	Count of NAs	Percent NAs	Column Type
county	1174	0.5887663	integer
community	1177	0.5902708	integer
communityname	0	0.0000000	character
LemasSwornFT	1675	0.8400201	numeric
LemasSwFTPerPop	1675	0.8400201	numeric
LemasSwFTFieldOps	1675	0.8400201	numeric
LemasSwFTFieldPerPop	1675	0.8400201	numeric
LemasTotalReq	1675	0.8400201	numeric
LemasTotReqPerPop	1675	0.8400201	numeric
PolicReqPerOffic	1675	0.8400201	numeric
PolicPerPop	1675	0.8400201	numeric
RacialMatchCommPol	1675	0.8400201	numeric
PctPolicWhite	1675	0.8400201	numeric
PctPolicBlack	1675	0.8400201	numeric
PctPolicHisp	1675	0.8400201	numeric
PctPolicAsian	1675	0.8400201	numeric
PctPolicMinor	1675	0.8400201	numeric
OfficAssgnDrugUnits	1675	0.8400201	numeric
NumKindsDrugsSeiz	1675	0.8400201	numeric

Removed Variables	Count of NAs	Percent NAs	Column Type
PolicAveOTWorked	1675	0.8400201	numeric
PolicCars	1675	0.8400201	numeric
PolicOperBudg	1675	0.8400201	numeric
LemasPctPolicOnPatr	1675	0.8400201	numeric
LemasGangUnitDeploy	1675	0.8400201	numeric
PolicBudgPerPop	1675	0.8400201	numeric

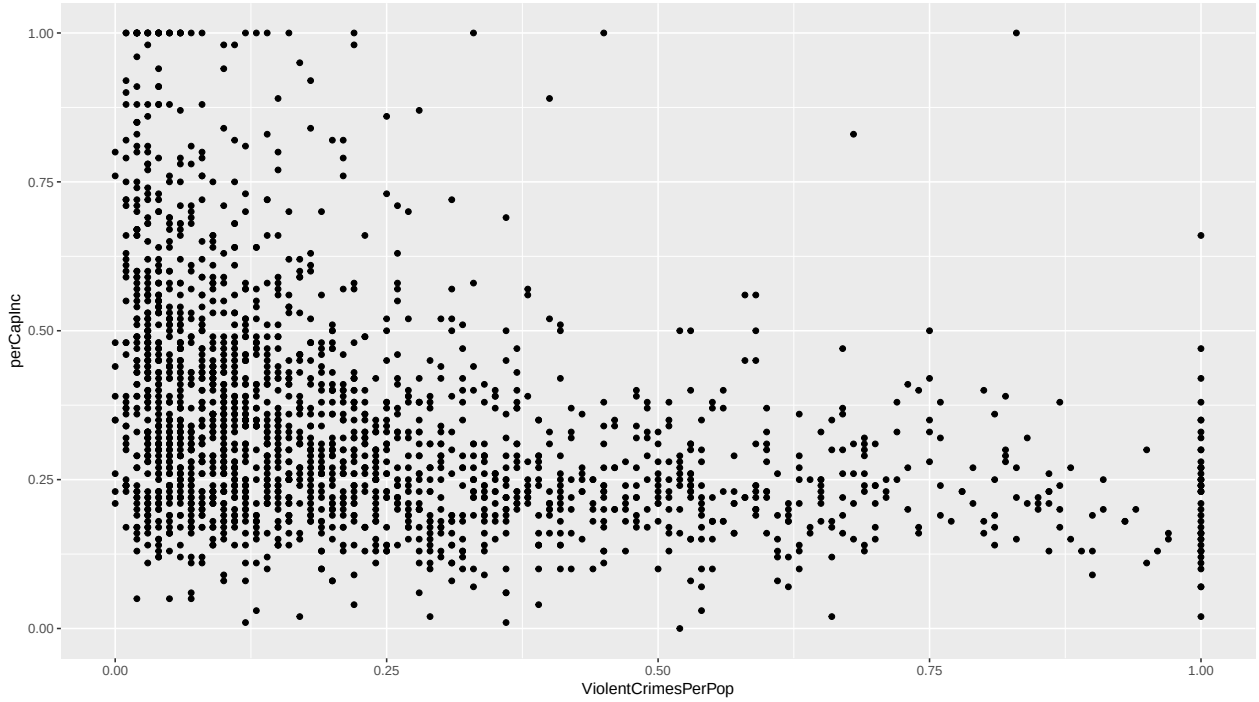
Also any rows with NAs will be omitted from the analysis. In our case this is only one row caused by the single NA value in the OtherPerCap variable.

In Depth Single Variable Exploration

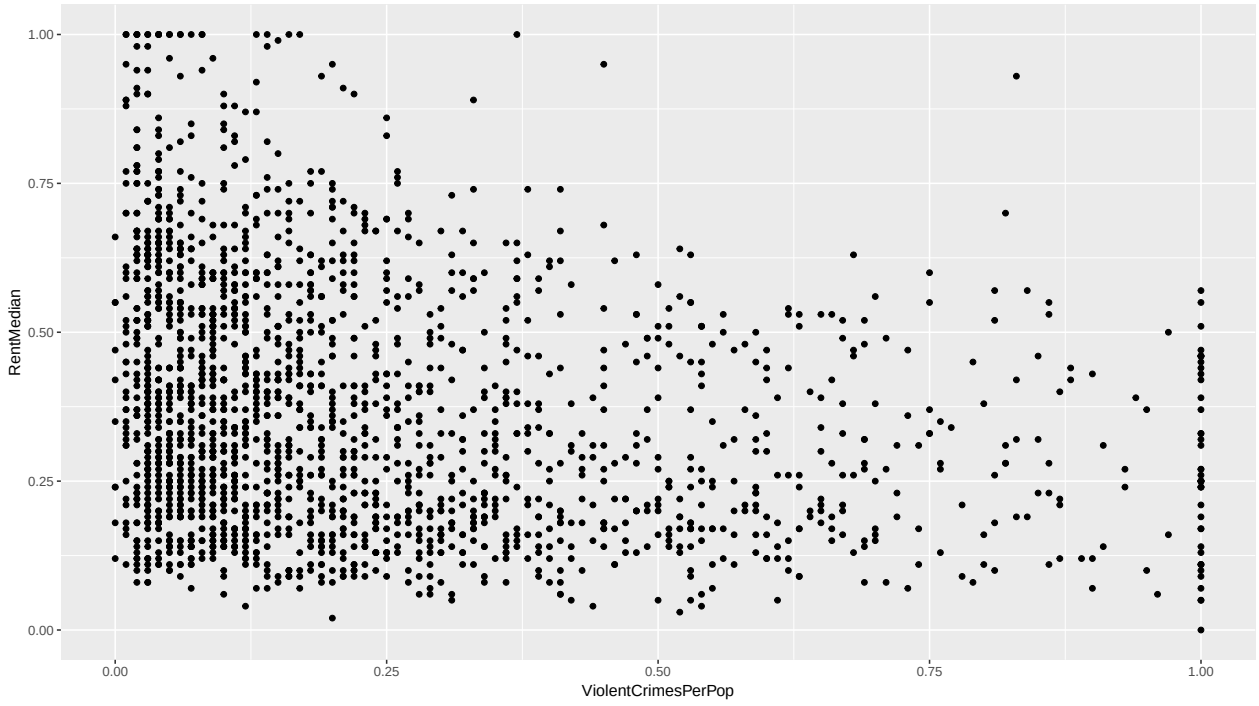
This section is a more in depth of an exploration of a few variables that I assume are related to the target variable our model will try to predict ViolentCrimesPerPop



Per capita income vs crimes shows lower income trends with more crimes



Lower Median Rent vs crime shows lower rent trends with more crimes



Step 2: Fit a linear Model

```
##
## Call:
## lm(formula = ViolentCrimesPerPop ~ PctEmploy + perCapInc + RentMedian +
##     PctHousOwnOcc, data = crime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5373 -0.1247 -0.0279  0.0694  0.8353
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.68226    0.01727  39.508  < 2e-16 ***
## PctEmploy     -0.32196    0.03031 -10.621  < 2e-16 ***
## perCapInc     -0.22313    0.04000  -5.578 2.76e-08 ***
## RentMedian     0.16209    0.03573   4.536 6.08e-06 ***
## PctHousOwnOcc -0.48321    0.02646 -18.264  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1966 on 1988 degrees of freedom
## Multiple R-squared:   0.29,  Adjusted R-squared:  0.2886
## F-statistic:   203 on 4 and 1988 DF,  p-value: < 2.2e-16
```

As you can see all of my variables are statistically significant from the p-values in the model summary. The R-squared and Adjusted R-squared are quite low though and point to the fact that the features chosen only explain around .28 of the variance in the ViolentCrimesPerPop variable.

The parameter estimates were similar to my intuition based on the EDA graphs for 3 of 4 variables. PctEmploy

has a negative slope meaning as employment decreases violent crimes increase. perCapInc has a negative slope meaning as income decreases violent crimes increase. RentMedian has a positive slope meaning as median rent increases violent crimes increase this is not what I expected based on the EDA graph.PctHousOwnOcc has a negative slope meaning as house owners as a percentage of the population decreases violent crimes increase.

Step 3: Perform model selection

Model selection will be done with both the fastbw and stepAIC algorithms

Final stepAIC() model

The stepAIC model used PctEmploy,PctHousOwnOcc that I used in my model but didnt use RentMedian (used RentLowQ and MedRent instead) also didnt use perCapInc (but used medIncome, pctWWage,medFamInc which all deal with income). This model makes sense to me and roughly matched my intuition for a 4 variable model. The variables I threw out on my own were the variables that were sparse (had mostly NA values) and to get the selection algorithms to work I had to also remove these variables. Since there were over 50 variables deleted I cant cover all of them. Most were deleted from multicoliniarity.

```
##
## Call:
## lm(formula = ViolentCrimesPerPop ~ state + fold + racepctblack +
##     racePctHisp + agePct12t29 + pctUrban + medIncome + pctWWage +
##     pctWFarmSelf + pctWInvInc + pctWRetire + medFamInc + whitePerCap +
##     indianPerCap + OtherPerCap + PctPopUnderPov + PctLess9thGrade +
##     PctEmploy + PctEmplManu + PctOccupManu + PctOccupMgmtProf +
##     MalePctDivorce + MalePctNevMarr + TotalPctDiv + PctKids2Par +
##     PctWorkMom + NumIlleg + PctIlleg + NumImmig + PctNotSpeakEnglWell +
##     PctLargHouseOccup + PersPerOccupHous + PersPerRentOccHous +
##     PctPersOwnOccup + PctPersDenseHous + HousVacant + PctHousOccup +
##     PctHousOwnOcc + PctVacantBoarded + PctVacMore6Mos + OwnOccLowQuart +
##     OwnOccMedVal + RentLowQ + MedRent + MedOwnCostPctIncNoMtg +
##     NumInShelters + NumStreet + PctForeignBorn + PctSameCity85 +
##     PctUsePubTrans + LemasPctOfficDrugUn, data = crime)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49810 -0.07040 -0.01311  0.05059  0.72570
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.6828592  0.1047842   6.517 9.13e-11 ***
## state                 -0.0007642  0.0002357  -3.242 0.001205 **
## fold                  -0.0017731  0.0010325  -1.717 0.086087 .
## racepctblack           0.2183584  0.0306686   7.120 1.51e-12 ***
## racePctHisp            0.0682280  0.0364392   1.872 0.061305 .
## agePct12t29           -0.2484167  0.0717968  -3.460 0.000552 ***
## pctUrban               0.0370719  0.0090822   4.082 4.65e-05 ***
## medIncome             -0.2706310  0.1561711  -1.733 0.083270 .
## pctWWage              -0.2674909  0.0607172  -4.406 1.11e-05 ***
## pctWFarmSelf           0.0467748  0.0190613   2.454 0.014218 *
## pctWInvInc            -0.1503688  0.0603273  -2.493 0.012766 *
```

```
## pctWRetire            -0.0792727  0.0317822  -2.494 0.012705 *
## medFamInc              0.3759525  0.1375010   2.734 0.006310 **
## whitePerCap           -0.2191710  0.0733787  -2.987 0.002854 **
## indianPerCap          -0.0343852  0.0189805  -1.812 0.070201 .
## OtherPerCap            0.0521395  0.0173481   3.005 0.002686 **
## PctPopUnderPov        -0.1401215  0.0471712  -2.970 0.003010 **
## PctLess9thGrade       -0.0590198  0.0347676  -1.698 0.089752 .
## PctEmploy              0.2126117  0.0588839   3.611 0.000313 ***
## PctEmplManu           -0.0573114  0.0294144  -1.948 0.051509 .
## PctOccupManu           0.0953577  0.0508539   1.875 0.060925 .
## PctOccupMgmtProf       0.1229338  0.0534147   2.301 0.021469 *
## MalePctDivorce         0.3452291  0.0923802   3.737 0.000192 ***
## MalePctNevMarr         0.1826948  0.0551118   3.315 0.000933 ***
## TotalPctDiv           -0.3203348  0.1010431  -3.170 0.001547 **
## PctKids2Par           -0.3468446  0.0749928  -4.625 3.99e-06 ***
## PctWorkMom            -0.1315314  0.0274434  -4.793 1.77e-06 ***
## NumIlleg              -0.1156617  0.0720890  -1.604 0.108782
## PctIlleg               0.1237491  0.0421422   2.936 0.003359 **
## NumImmig              -0.1448137  0.0636192  -2.276 0.022939 *
## PctNotSpeakEnglWell   -0.1549003  0.0588157  -2.634 0.008514 **
## PctLargHouseOccup     -0.1475370  0.0602397  -2.449 0.014407 *
## PersPerOccupHous       0.3850507  0.1076547   3.577 0.000356 ***
## PersPerRentOccHous    -0.2449376  0.0714758  -3.427 0.000623 ***
## PctPersOwnOccup       -0.8048420  0.2127810  -3.782 0.000160 ***
## PctPersDenseHous       0.2215173  0.0637016   3.477 0.000517 ***
## HousVacant             0.1528884  0.0471289   3.244 0.001198 **
## PctHousOccup          -0.0553692  0.0258513  -2.142 0.032331 *
## PctHousOwnOcc          0.6977220  0.2104963   3.315 0.000934 ***
## PctVacantBoarded       0.0566947  0.0203293   2.789 0.005342 **
## PctVacMore6Mos        -0.0695478  0.0235363  -2.955 0.003165 **
## OwnOccLowQuart        -0.4137020  0.1632949  -2.533 0.011372 *
## OwnOccMedVal           0.2727761  0.1566819   1.741 0.081850 .
## RentLowQ              -0.2122227  0.0542111  -3.915 9.36e-05 ***
## MedRent                0.2888072  0.0694428   4.159 3.34e-05 ***
## MedOwnCostPctIncNoMtg -0.0805908  0.0213697  -3.771 0.000167 ***
## NumInShelters          0.1172443  0.0616438   1.902 0.057324 .
## NumStreet              0.1712188  0.0455371   3.760 0.000175 ***
## PctForeignBorn         0.1436999  0.0435933   3.296 0.000997 ***
## PctSameCity85          0.0548008  0.0265757   2.062 0.039334 *
## PctUsePubTrans        -0.0320069  0.0204407  -1.566 0.117549
## LemasPctOfficDrugUn    0.0217812  0.0149237   1.460 0.144588
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1307 on 1941 degrees of freedom
## Multiple R-squared:  0.6937, Adjusted R-squared:  0.6857
## F-statistic:  86.2 on 51 and 1941 DF,  p-value: < 2.2e-16
```

Final fastbw() model

The fastbw model selected PctEmploy and PctHousOwnOcc which I included in my model but did not include perCapInc (used pctWWage which deals with income also) and RentMedian (used RentLowQ + MedRent). Honestly the model from fastbw matched my intuition. I picked variables that are associated with income

and housing and this model chose the variables I did or very similar ones. It also added in variables related to Immigrants and Divorce. Most of these variables seem plausable to include in a violent crime model.Since there were over 50 variables deleted I cant cover all of them. Most were deleted from multicoliniarity.

```
##
## Call:
## lm(formula = ViolentCrimesPerPop ~ state + racepctblack + agePct12t29 +
##     pctUrban + pctWWage + pctWFarmSelf + pctWInvInc + OtherPerCap +
##     PctEmploy + MalePctDivorce + MalePctNevMarr + TotalPctDiv +
##     PctKids2Par + PctWorkMom + PctIlleg + NumImmig + PctNotSpeakEnglWell +
##     PersPerOccupHous + PersPerRentOccHous + PctPersOwnOccup +
##     PctPersDenseHous + HousVacant + PctHousOccup + PctHousOwnOcc +
##     OwnOccLowQuart + RentLowQ + MedRent + MedOwnCostPctIncNoMtg +
##     NumStreet + PctForeignBorn, data = crime)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.53762 -0.07057 -0.01625  0.04922  0.75609
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.4931754  0.0779274   6.329 3.06e-10 ***
## state                   -0.0008593  0.0002316  -3.710 0.000213 ***
## racepctblack             0.1710030  0.0265592   6.439 1.51e-10 ***
## agePct12t29             -0.2331328  0.0646268  -3.607 0.000317 ***
## pctUrban                 0.0400950  0.0085204   4.706 2.71e-06 ***
## pctWWage                -0.1757872  0.0535701  -3.281 0.001051 **
## pctWFarmSelf             0.0462460  0.0176910   2.614 0.009015 **
## pctWInvInc              -0.1477965  0.0488243  -3.027 0.002501 **
## OtherPerCap              0.0439945  0.0174623   2.519 0.011834 *
## PctEmploy                0.2128927  0.0515762   4.128 3.82e-05 ***
## MalePctDivorce           0.2792316  0.0839214   3.327 0.000893 ***
## MalePctNevMarr           0.1781789  0.0513057   3.473 0.000526 ***
## TotalPctDiv             -0.2016895  0.0926139  -2.178 0.029544 *
## PctKids2Par             -0.2713084  0.0691508  -3.923 9.03e-05 ***
## PctWorkMom              -0.1152584  0.0242061  -4.762 2.06e-06 ***
## PctIlleg                 0.1450430  0.0410857   3.530 0.000425 ***
## NumImmig                -0.1496359  0.0587229  -2.548 0.010905 *
## PctNotSpeakEnglWell     -0.1328309  0.0481308  -2.760 0.005838 **
## PersPerOccupHous         0.2442286  0.0743785   3.284 0.001043 **
## PersPerRentOccHous      -0.2336024  0.0689617  -3.387 0.000719 ***
## PctPersOwnOccup         -0.5912802  0.1968921  -3.003 0.002707 **
## PctPersDenseHous         0.1882265  0.0514134   3.661 0.000258 ***
## HousVacant               0.1452829  0.0324499   4.477 8.00e-06 ***
## PctHousOccup            -0.0448000  0.0217997  -2.055 0.040004 *
## PctHousOwnOcc            0.5138499  0.1948882   2.637 0.008439 **
## OwnOccLowQuart          -0.1436244  0.0378788  -3.792 0.000154 ***
## RentLowQ                -0.2124346  0.0531449  -3.997 6.64e-05 ***
## MedRent                  0.2812336  0.0661207   4.253 2.21e-05 ***
## MedOwnCostPctIncNoMtg -0.0930976  0.0204663  -4.549 5.72e-06 ***
## NumStreet                0.1980438  0.0430651   4.599 4.52e-06 ***
## PctForeignBorn           0.1418282  0.0410506   3.455 0.000562 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.1322 on 1962 degrees of freedom
## Multiple R-squared:  0.6828, Adjusted R-squared:  0.678
## F-statistic: 140.8 on 30 and 1962 DF,  p-value: < 2.2e-16
```

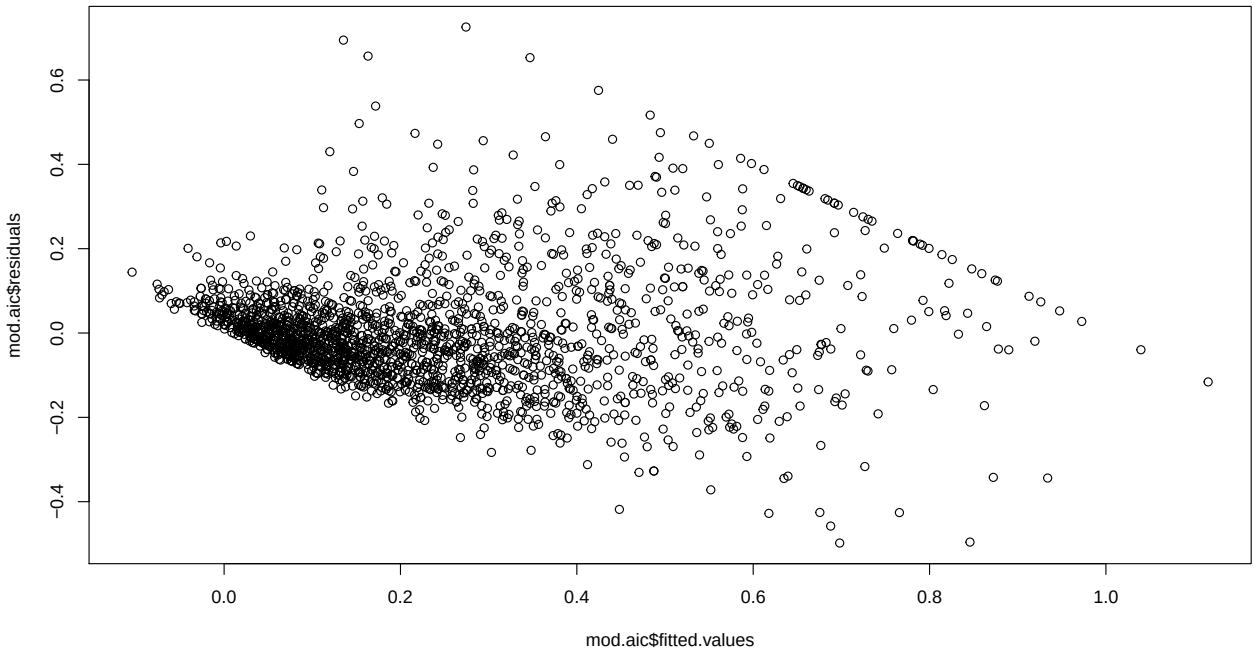
Final model choice

I would chose the model produced by stepAIC becuase it has a better adjusted R-squared (0.6857 vs. 0.678) although it does use 21 more predictor variables. Adjusted R-squared does penalize using more predictors so this is the most standardized way to compare the two models that use different numbers of predictors.

Step 4: Apply diagnostics to the model

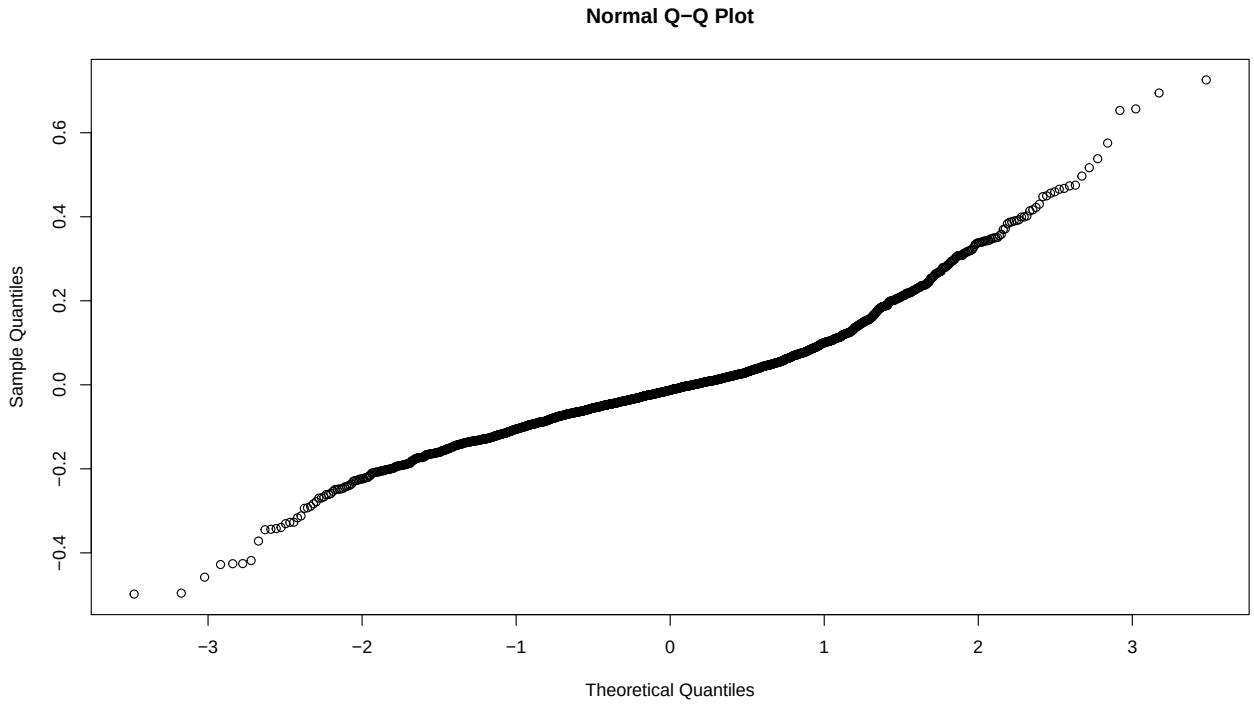
Fitted values vs residuals plot

There seems to be mostly random scattering of residuals but there seems to be a trendline from top left to bottom right this assumption of randomness in residuals vs fitted vals doesnt seem upheld



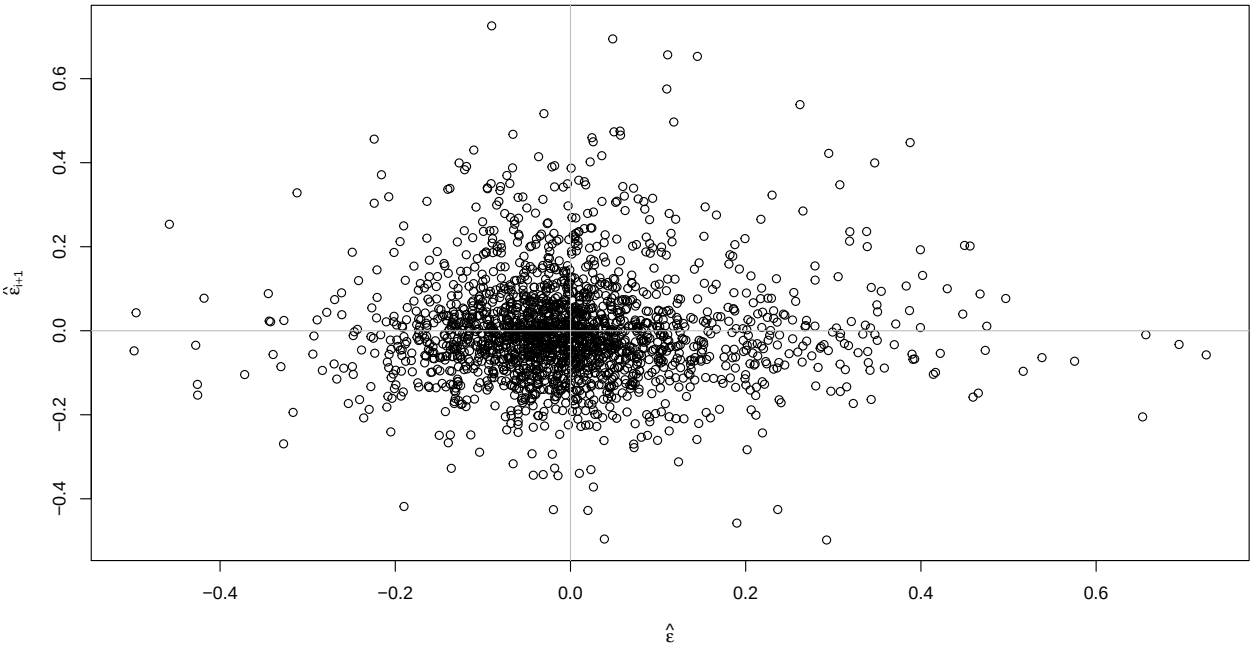
Q-Q plot

The line is mostly straight but has a pretty big bend up that shows the assumption of randomness does not seem to be upheld



Lagged residual plot

This shows a roughly random plot so the assumptions are held up and the lagged residuals are randomly distributed



Step 5: Investigate fit for individual observations

```
## [1] "Number of high leverage points: 135"
```

```
## [1] "Number of outlier points: 35"
## [1] "Number of problematic points: 1"
## [1] "Largest cook's distance: 0.0353570893245684"
## [1] "F statistic threshold: 0.987548659363356"
## [1] "Number of influential points: 0"
```

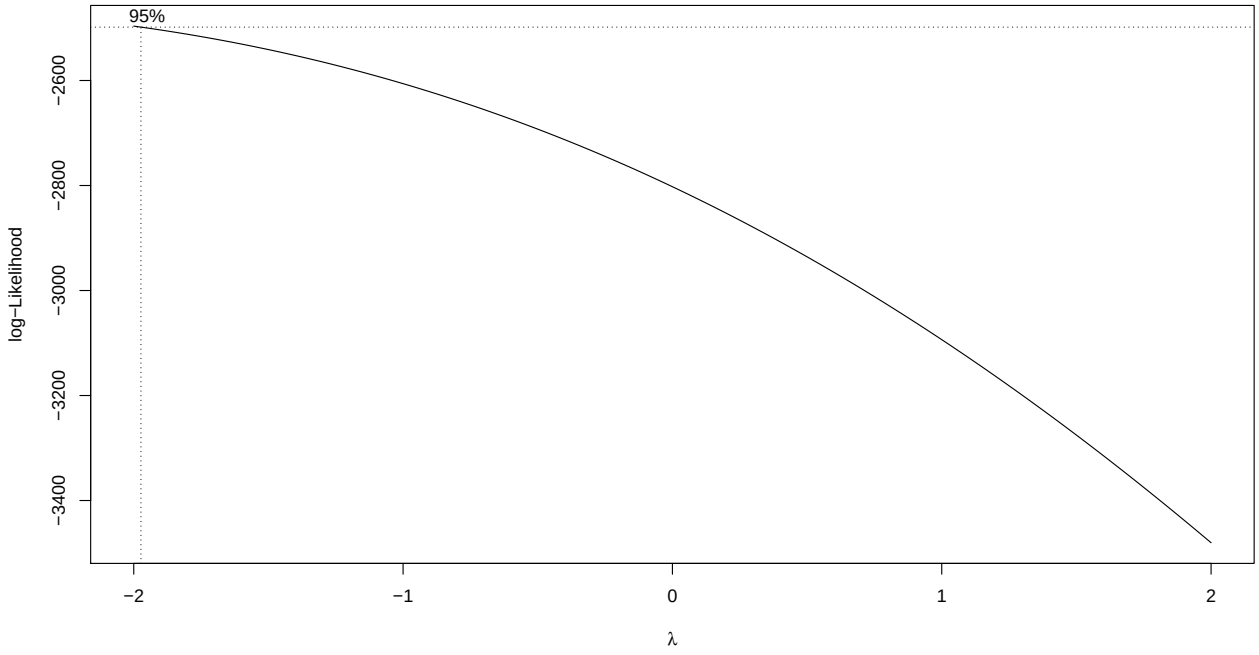
Discussion of results

While there are 35 outlier points, none of those points are considered influential. We only have one point that we would consider to be problematic (the cross section of the high leverage and outlier points). None of the points are even remotely close to being influential because the threshold is ~.99 and the largest cook value is .035. Basically this means that even though we have outlier points we can leave them in since they arent influential enough to affect our model materially

Step 6: Apply transformations to model as needed

A BoxCox transformation is needed. With the lambda value of -2 as shown by the peak in the chart

```
## [1] "R-squared value before Box-Cox transformation: 0.6937"
```




```
## [1] "Box-Cox transformation Optimal Lambda Value: -2"
## [1] "R-squared value after Box-Cox transformation: 0.7212"
```

Since the R squared value is improved materially the box cox transformation makes a positive impact on the model's ability to predict and should be used.

Step 7: Report inferences and make predictions using your final model

Print out Parameter Estimates and P-values for each Predictor Variable

Predictor	Parameter.Estimate	p.Value
(Intercept)	0.3464917	5.52e-05
state	0.0005180	7.28e-03
fold	0.0009554	2.58e-01
racepctblack	-0.1483435	3.98e-09
racePctHisp	-0.1265353	2.30e-05
agePct12t29	0.1318808	2.49e-02
pctUrban	-0.0339363	5.26e-06
medIncome	0.3938940	2.08e-03
pctWWage	0.2236541	7.13e-06
pctWFarmSelf	-0.0356800	2.23e-02
pctWInvInc	0.2426663	9.55e-07
pctWRetire	0.0377636	1.47e-01
medFamInc	-0.4385685	1.00e-04
whitePerCap	0.1261845	3.57e-02
indianPerCap	0.0279301	7.22e-02
OtherPerCap	-0.0405804	4.29e-03
PctPopUnderPov	0.1347098	4.93e-04
PctLess9thGrade	0.0192416	4.99e-01
PctEmploy	-0.1814433	1.71e-04
PctEmplManu	0.0412859	8.64e-02
PctOccupManu	-0.0475778	2.53e-01
PctOccupMgmtProf	-0.1151838	8.46e-03
MalePctDivorce	-0.2595233	6.08e-04
MalePctNevMarr	-0.1009014	2.54e-02
TotalPctDiv	0.2055485	1.30e-02
PctKids2Par	0.3210593	1.85e-07
PctWorkMom	0.0939524	2.99e-05
NumIlleg	0.2367789	6.18e-05
PctIlleg	-0.0931542	6.96e-03
NumImmig	0.0405156	4.36e-01
PctNotSpeakEnglWell	0.1798780	1.91e-04
PctLargHouseOccup	0.1001643	4.23e-02
PersPerOccupHous	-0.2323755	8.40e-03
PersPerRentOccHous	0.1028660	7.87e-02
PctPersOwnOccup	0.6342727	2.76e-04
PctPersDenseHous	-0.1150366	2.74e-02
HousVacant	-0.1929227	6.15e-07
PctHousOccup	0.0228429	2.80e-01
PctHousOwnOcc	-0.5642649	1.07e-03
PctVacantBoarded	-0.0417596	1.21e-02
PctVacMore6Mos	0.0712099	2.23e-04
OwnOccLowQuart	0.2600471	5.17e-02
OwnOccMedVal	-0.1719652	1.80e-01
RentLowQ	0.1611702	2.87e-04
MedRent	-0.2145591	1.64e-04
MedOwnCostPctIncNoMtg	0.0608368	5.13e-04
NumInShelters	-0.0846801	9.33e-02

Predictor	Parameter.Estimate	p.Value
NumStreet	-0.0830508	2.59e-02
PctForeignBorn	-0.1424995	6.70e-05
PctSameCity85	-0.0445615	4.06e-02
PctUsePubTrans	0.0329417	4.90e-02
LemasPctOfficDrugUn	-0.0289814	1.77e-02

R-squared for model

```
## [1] "The R-squared value for this model is: 0.7212"
```

95% CI of slope for most important predictor

```
##                    2.5 %      97.5 %
## racepctblack -0.1975539 -0.09913309
```

95% CI of predictions at median values for each column

```
##         fit       lwr       upr
## 1 0.7431254 0.7312541 0.7549967
```

95% PI of predictions at median values for each column

```
##         fit       lwr       upr
## 1 0.7431254 0.5331433 0.9531075
```

The prediction and confidence intervals predict the same fit but the prediction interval is much wider to account for potential variation that any one sample might have.

Conclusion

In conclusion the most useful variables in an automated model selection are not that different from the variables chosen by EDA. These automated selection methods are able to chose variables much faster and with end results that are pretty good! This also showed the power of transformations such as the Box-Cox method which gave 3% improvement in R^2 for a linear model fit on the same variables. Finally we predicted the violent crimes per 100k for the median of each variable in our dataset and gave both a confidence and prediction interval that we would expect real world values to fit into. The major things left out of this report deal with testing models on data that they werent trained with. This analysis doesnt run a true test of accuracy and only gives R^2 and adjusted R^2 values which do not tell anything about overfitting that may be occuring since training and analysis are done on the same dataset.

Appendix

In the zip file is a .R file with all this code nicely formated for execution as well as the crime data file and column names csv file

R code block

```
############### prework ####################
# read in libraries
library(tidyverse) # general dataframe manipulation
library(MASS) # statistical package
```

```
library(rms) # regression modeling package

# read in data file and column names file
crime<- read_csv("crime.txt",col_names=F)
colsnames <- read_csv("colnames.csv")
colnames(crime)<-colnames(colsnames)

############### Step 1: Exploratory Data Analysis ####################
# get basic view of raw dataset
glimpse(crime)

# function to clean up columns that had ? in them
convert_types <- function(x) {
    stopifnot(is.list(x))
    x[] <- rapply(x, utils::type.convert, classes = "character",
                  how = "replace", as.is = TRUE)
    return(x)
}

# replace question marks with NAs and convert all columns back to what they should be
crime[crime=="?"] <- NA
crime <- convert_types(crime)

# analyze the removed columns
crime.removed <- crime %>% dplyr::select(county,community,communityname,
                              c(LemasSwornFT:PolicAveOTWorked),
                              c(PolicCars:LemasGangUnitDeploy),PolicBudgPerPop)

# make table describing the columns removed from analysis
colSums(is.na(crime.removed))%>% data.frame()%>% rename("Count of NAs" = ".") %>%
  rownames_to_column("Removed Variables")%>%
  mutate(`Percent NAs` = `Count of NAs`/dim(crime.removed)[1],
         `Column Type` = sapply(crime.removed,class))

# remove columns with too many NA values
crime <- crime %>% dplyr::select(-county,-community,-communityname,
                          -c(LemasSwornFT:PolicAveOTWorked),
                          -c(PolicCars:LemasGangUnitDeploy),-PolicBudgPerPop)
# remove rows with NA values
crime <- crime %>% na.omit()

## In Depth Single Variable Exploration

ggplot(
  data = crime,
  aes(y = PctEmploy,x = ViolentCrimesPerPop)
) + geom_point() +
  ggtitle(paste("The percentage of people employed vs crimes shows lower",
                " employment trends with more crimes"))

ggplot(
  data = crime,
  aes(y = perCapInc,x = ViolentCrimesPerPop)
```

```
) + geom_point() +
  ggtitle(paste("Per capita income vs crimes shows lower income",
                           " trends with more crimes"))

ggplot(
  data = crime,
  aes(y = RentMedian,x = ViolentCrimesPerPop)
) + geom_point() +
  ggtitle(paste("Lower Median Rent vs crime shows lower rent",
                           "trends with more crimes"))

ggplot(
  data = crime,
  aes(y = PctHousOwnOcc,x = ViolentCrimesPerPop)
) + geom_point() +
  ggtitle(paste("Percentage of houses owned by the occupants shows",
                " lower homeownership trends with more crime"))


###### Step 2: Fit a linear Model ####################

# make lm with 4 predictors chosen in EDA
simple_4var_lm <- lm(ViolentCrimesPerPop ~ PctEmploy + perCapInc +
                       RentMedian + PctHousOwnOcc,data = crime)
summary(simple_4var_lm)

####### Step 3: Perform model selection ####################

### fastbw
# fit ols model with every variable
mod.ols <- ols( ViolentCrimesPerPop ~ ., data=crime)
# run fastbw variable selection
fastbw.parameters <- fastbw(mod.ols, rule="p", sls=0.05)
# print out variables saved
fastbw.parameters$coefficients

### stepAIC
mod <- lm(ViolentCrimesPerPop ~ ., data=crime)
summary(mod)
mod.aic <- stepAIC(mod)

# Final stepAIC model
summary(mod.aic)

# Final fastbw model
mod.fastbw <- lm(ViolentCrimesPerPop ~ state+racepctblack+agePct12t29+pctUrban +
pctWWage+pctWFarmSelf+pctWInvInc+OtherPerCap +
PctEmploy+MalePctDivorce+MalePctNevMarr+TotalPctDiv+
PctKids2Par+PctWorkMom+PctIlleg+NumImmig+
PctNotSpeakEnglWell+PersPerOccupHous+PersPerRentOccHous+PctPersOwnOccup+
PctPersDenseHous+HousVacant+PctHousOccup+PctHousOwnOcc+
OwnOccLowQuart+RentLowQ+MedRent+MedOwnCostPctIncNoMtg+
NumStreet+PctForeignBorn , data=crime)
```

```r
summary(mod.fastbw)



### Step 4: Apply diagnostics to the model ####################

## Fitted values vs residuals plot
plot(mod.aic$fitted.values, mod.aic$residuals)

## Q-Q plot
qqnorm(mod.aic$residuals)

## Lagged residual plot
n <- length(residuals(mod.aic))
plot(tail(residuals(mod.aic),n-1)~head(residuals(mod.aic),n-1),
     xlab = expression(hat(epsilon)), ylab=expression(hat(epsilon)[i+1]))
abline(h=0,v=0,col=grey(.75))

### Step 5: Investigate fit for individual observations ####################

X <- model.matrix(mod.aic)
n <- dim(X)[1]
p <- dim(X)[2]

hatv <- hatvalues(mod.aic)

# Leverage investigation

high.leverage <- hatv[hatv>2*p/n]
print(paste("Number of high leverage points:",length(high.leverage)))

# Outlier investigation

rstd <- rstandard(mod.aic)
outlier <- rstd[abs(rstd)>3]

print(paste("Number of outlier points:",length(outlier)))


rstd.hand <- mod.aic$residuals/(sqrt(1- hatv)*
                sqrt(sum(mod.aic$residuals^2)/mod.aic$df.residual))

# Problematic investigation

problematic <- which(hatv>2*p/n&abs(rstd)>3)
print(paste("Number of problematic points:",length(problematic)))

# Cooks distance influence investigation

cook <- cooks.distance(mod.aic)

print(paste("Largest cook's distance:",max(cook)))
```

```r
num.df <- p
den.df <- n-p
F.thresh <- qf(0.5,num.df,den.df)

print(paste("F statistic threshold:",F.thresh))

influential <- which(hatv>2*p/n&abs(rstd)>3&cook>F.thresh)
print(paste("Number of influential points:",length(influential)))

################ Step 6: Apply transformations to model as needed ####################

# make Violent Crimes Per Pop always positive for boxcox
crime.pos <- crime
crime.pos$ViolentCrimesPerPop <- crime.pos$ViolentCrimesPerPop + 1

mod.aic.pos <- lm(formula = ViolentCrimesPerPop ~ state + fold + racepctblack +
    racePctHisp + agePct12t29 + pctUrban + medIncome + pctWWage +
    pctWFarmSelf + pctWInvInc + pctWRetire + medFamInc + whitePerCap +
    indianPerCap + OtherPerCap + PctPopUnderPov + PctLess9thGrade +
    PctEmploy + PctEmplManu + PctOccupManu + PctOccupMgmtProf +
    MalePctDivorce + MalePctNevMarr + TotalPctDiv + PctKids2Par +
    PctWorkMom + NumIlleg + PctIlleg + NumImmig + PctNotSpeakEnglWell +
    PctLargHouseOccup + PersPerOccupHous + PersPerRentOccHous +
    PctPersOwnOccup + PctPersDenseHous + HousVacant + PctHousOccup +
    PctHousOwnOcc + PctVacantBoarded + PctVacMore6Mos + OwnOccLowQuart +
    OwnOccMedVal + RentLowQ + MedRent + MedOwnCostPctIncNoMtg +
    NumInShelters + NumStreet + PctForeignBorn + PctSameCity85 +
    PctUsePubTrans + LemasPctOfficDrugUn, data = crime.pos)

print(paste("R-squared value before Box-Cox transformation:",
            round(summary(mod.aic.pos)$r.squared,4)))

bc <- boxcox(mod.aic.pos, plotit=T)
lambda <- bc$x[which.max(bc$y)]
print(paste("Box-Cox transformation Optimal Lambda Value:",lambda))

# fit model with transformation

mod.aic.pos.transform <- lm(formula = ViolentCrimesPerPop^lambda ~ state + fold +
    racepctblack + racePctHisp + agePct12t29 + pctUrban + medIncome + pctWWage +
    pctWFarmSelf + pctWInvInc + pctWRetire + medFamInc + whitePerCap +
    indianPerCap + OtherPerCap + PctPopUnderPov + PctLess9thGrade +
    PctEmploy + PctEmplManu + PctOccupManu + PctOccupMgmtProf +
    MalePctDivorce + MalePctNevMarr + TotalPctDiv + PctKids2Par +
    PctWorkMom + NumIlleg + PctIlleg + NumImmig + PctNotSpeakEnglWell +
    PctLargHouseOccup + PersPerOccupHous + PersPerRentOccHous +
    PctPersOwnOccup + PctPersDenseHous + HousVacant + PctHousOccup +
    PctHousOwnOcc + PctVacantBoarded + PctVacMore6Mos + OwnOccLowQuart +
    OwnOccMedVal + RentLowQ + MedRent + MedOwnCostPctIncNoMtg +
    NumInShelters + NumStreet + PctForeignBorn + PctSameCity85 +
    PctUsePubTrans + LemasPctOfficDrugUn, data = crime.pos)

print(paste("R-squared value after Box-Cox transformation:",
```

```
            round(summary(mod.aic.pos.transform)$r.squared,4)))

#### Step 7: Report inferences and make predictions using your final model ##############

mod.coef <- data.frame(coef(summary(mod.aic.pos.transform)))

mod.coef$Predictor <- rownames(mod.coef)
rownames(mod.coef)<- NULL
mod.coef <- mod.coef[,c(5,1,4)]
colnames (mod.coef) <- c("Predictor", "Parameter Estimate", "p-Value")

output <- mod.coef
output$`p-Value` <- format(output$`p-Value`, digits = 3)
output%>% data.frame()

print(paste("The R-squared value for this model is:",
            round(summary(mod.aic.pos.transform)$r.squared,4)))

confint(mod.aic.pos.transform, 'racepctblack', level=0.95)

# create new data out of median of all columns used in model
mod.data <- crime.pos %>% dplyr::select(ViolentCrimesPerPop, state , fold ,
    racepctblack , racePctHisp , agePct12t29 , pctUrban , medIncome , pctWWage ,
    pctWFarmSelf , pctWInvInc , pctWRetire , medFamInc , whitePerCap ,
    indianPerCap , OtherPerCap , PctPopUnderPov , PctLess9thGrade ,
    PctEmploy , PctEmplManu , PctOccupManu , PctOccupMgmtProf ,
    MalePctDivorce , MalePctNevMarr , TotalPctDiv , PctKids2Par ,
    PctWorkMom , NumIlleg , PctIlleg , NumImmig , PctNotSpeakEnglWell ,
    PctLargHouseOccup , PersPerOccupHous , PersPerRentOccHous ,
    PctPersOwnOccup , PctPersDenseHous , HousVacant , PctHousOccup ,
    PctHousOwnOcc , PctVacantBoarded , PctVacMore6Mos , OwnOccLowQuart ,
    OwnOccMedVal , RentLowQ , MedRent , MedOwnCostPctIncNoMtg ,
    NumInShelters , NumStreet , PctForeignBorn , PctSameCity85 ,
    PctUsePubTrans , LemasPctOfficDrugUn)
newdata <-  data.frame(rbind(apply(mod.data, 2, median)))

# confidence interval of ViolentCrimesPerPop for new data
predict(mod.aic.pos.transform, newdata, interval="confidence")

# prediction interval of ViolentCrimesPerPop for new data
predict(mod.aic.pos.transform, newdata, interval = "predict")
```