Predicting the Proportion of People in Poverty by Neighborhood in Denver using Housing and Crime Demographics

Michael Ingram

Introduction

Predicting the poverty levels by neighborhood is important information for Denver policy makers. With Denver growing at an unprecedented rate, new neighborhoods are being constructed or rezoned. Being able to accurately predict the poverty levels of neighborhoods would be able to help with a multitude of policy making decisions. In addition, identifying the most significant factors that contribute to poverty levels could help policy makers and those in real estate to raise or lower the poverty levels of neighborhoods. This could contribute to the raising or lowering of housing prices in a neighborhood.

The data set used in the analysis is observational and came from combining information from two different sources. The first source that most of the demographics came from was a 2014 Denver housing data set. The second data set was a crime data set from: https://www.denvergov.org/opendata/dataset/city-and-county-of-denver-crime. Both data sets were found through the data 2 policy project available on the Auraria Library website. 50 variables were selected from the housing data set. While from the crime data set, the number of incidents that occurred in each neighborhood was summed. This gives a total of 51 predictor variables. Something to note is that the number of neighborhoods from the crime data set was less than the number of neighborhoods in the housing data set so missing values were filled in with the mean number of incidents per neighborhood. Table 1 shows the predictor variables used.

Table 1.

Predictor Variables		
1. Households	18. Housing Vacancy	35. Population Non Latino

2. Housing Units	19. Multifamily Units	36. Population Non Latino White
3. Persons per Household	20. Overcrowded Housing Units	37. Population Other Race, Single Race Selected
4. Total Population	21. People Over 25 with College Associates or Better	38. Population White
5. Crime Incidents	22. People Over 25 with High School Only	39. Population of 2 or More Races
6. One Person Housing	23. People Over 25 with less than 12th Grade Education	40. Population over 65
7. Adults Non English Speaking	24. People Over 25 with Some College No Degree	41. Population under 5
8. Births to Teen Mothers	25. Population 18-24	42. Single Family Units
9. Births to Unwed Mothers	26. Population 25-34	43. Renters Spending More Than 30% of Income on Housing
10. Births to Women Less with than a 12th Grade Education	27. Population 35-44	44. Single Mothers with Children in Poverty
11. Children Living With Single Parents	28. Population 45-54	45. Unemployed in Civilian Labor Force
12. Families with Children	29. Population 5-17	46. Births African American

13. Families without Children	30. Population 55-64	47. Births Asian and Pacific Islander
14. Foreign Born	31. Population African American	48. Births Latino
15. Households with Income more than 125k	32. Population Asian and Pacific Islander	49. Births Native American
16. Households with Income less than 60k	33. Population Latino	50. Births Non Latino White
17. Households with Income between 60k and 125k	34. Population Native American	51. Births Other Races

The predictor variables Households, Housing Units, Total Population and Crime Incidents were count variables. All other variables started as count variables but were transformed into proportions by dividing by either Households, Housing Units or Total Population. The one exception to this is that the Persons Per Household variable came as a proportion in the original data set. The data set was split into a training and test data set in order to do prediction. This data set contained a total of n=262 observations which were split 75/25 into 196 and 66 observations.

Methods

Four regression models were built in this analysis. I originally started with ordinary least squares regression. In order to meet the assumptions of ordinary least squares, many of the predictors had to be removed. This led me to try three other methods of linear regression to have models that would include more of the data and see if they would improve on the original model.

Ordinary Least Squares

I started by doing some exploratory data analysis on the data. Figure 1 shows the correlation between the different predictor variables. In the figure, the dark red circles correspond to strong negative correlation and the dark blue circles correspond to strong positive correlation. The number on the axis indicates the corresponding predictor variable from Table 1. Figure 1 shows that many of the predictor variables are highly correlated. This may indicate a problem in the model so to further analyze I look into the collinearity between predictor variables. In order to calculate collinearity, I fit an ordinary least squares model with all 51 predictors included and then analyze the variance inflation factors.

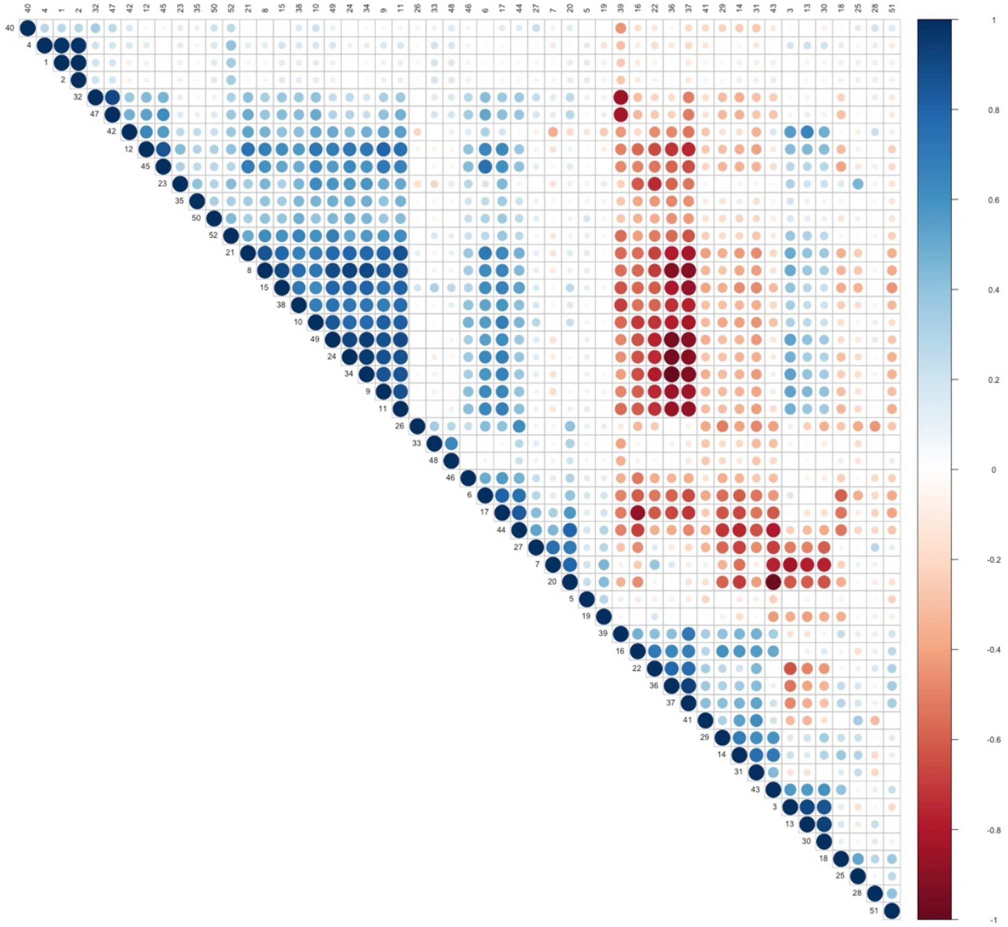


Figure 1

Before I could look into collinearity, I had another problem fitting the ordinary least squares model. The problem was that I had singularities in my data. Singularities are when certain column vectors are not independent from the others. This causes a problem when trying to obtain $\hat{\beta}$ coefficient estimates because the matrix $\hat{\beta} = (X^T X)^{-1} X^T y$ is not invertible. I remove the predictor variables causing the singularity since the information in those variables is already contained in other predictors and therefore will not change the model other than to fix the singularity issue. The predictors removed for singularities are 39. Population of 2 or more races, 40. Population over 65, 41. Population under 5, 17. Households with income between 60k and 125k, 35. Population Non Latino.

After those predictors are removed, I calculate the variance inflation factors to check for multicollinearity problems. All predictors except crime incidents which came from another data set had large variance inflation factors where large is defined as greater than 10. The variance inflation factors indicated that there was a big multicollinearity problem with the data and the solution to this problem was to remove the highly collinear variables.

Before the removal of predictors with high variance inflation factors, I used backward stepwise AIC model selection to see which predictors' removal would improve the model. Backward stepwise AIC tests the full model against another model where a predictor variable has been removed. If the removal of a predictor decreases the model's AIC then that predictor is removed and the process repeats until there is no longer any improvement. Running backward stepwise AIC on the full model left me with variables 1. Households, 3. Persons Per Household, 4. Total Population, 6. One Person Housing, 12. Families With Children, 13. Families Without Children, 14. Foreign Born, 16. Households with Income less than 60k,

18. Housing Vacancy, 19. Multifamily Units, 20. Overcrowded Housing Units, 21. People Over 25 with College Associates or Better, 22. People Over 25 with High School Only, 23. People Over 25 with less than 12th Grade Education, 24. People Over 25 with Some College No Degree, 25. Population 18-24, 26. Population 25-34, 27. Population 35-44_prop, 28. Population 45-54, 29. Population 5-17, 30. Population 55-64, 34. Population Native American, 36. Population Non Latino White , 42. Single Family Units, 43. Renters Spending More Than 30% of Income on Housing, 44. Single Mothers with Children in Poverty, 45. Unemployed in Civilian Labor Force, 46. Births African American, 47. Births Asian and Pacific Islander, 48. Births Latino, 50. Births Non Latino White.

After the backward stepwise AIC model selection, 31 variables remained of the original 51. I also ran a both directional stepwise AIC model selection and the same results from the backward directional were given. I refit the model with the 31 suggested variables and recalculated the variance inflation factors. The variance inflation factors had deceased from the previous variance inflation factors of the full model. Starting with the highest variance inflation factor, 21. People Over 25 with College Associates or Better, I removed the predictor with the highest variance inflation factor from the model, refit the model and then calculated the new inflation factors. I continued to exorcise the highest variance inflation factor from the model until no predictors with variance inflation factors over 10 remained in the model.

I performed another backward stepwise AIC on the remaining predictors to see if the model could be further improved. After removing predictors with high variance inflation factors and running backward stepwise AIC a second time the predictors leftover were 12. Families with Children, 13. Families without Children, 18. Housing Vacancy, 20. Overcrowded Housing Units,

24. People Over 25 with Some College No Degree, 25. Population 18-24, 42. Single Family Units, 43. Renters Spending More Than 30% of Income on Housing, 44. Single Mothers with Children in Poverty, and 50. Births Non Latino White.

Now that model selection is almost complete I do some hypothesis testing to verify results. First thing to do is test the null hypothesis, $H_0: \widehat{\beta_i} = 0$. This hypothesis test is testing for whether the corresponding regression coefficient differs from zero. In other words, is significant to the model. Table 2 shows the results of the hypothesis test. For $\alpha_{crit} = 0.05$, the hypothesis test shows to reject the null hypothesis, all coefficients are significant to the model.

Table 2

Hypothesis Test of Coefficients' Significance Results		
Coefficients	T-Statistics	P-Value
12. Families with Children	-3.657	0.000333
13. Families without Children	-3.012	0.002961
18. Housing Vacancy	2.383	0.018165
20. Overcrowded Housing Units	3.296	0.001175
24. People Over 25 with Some College No Degree	-2.632	0.009197
25. Population 18-24	2.615	0.009649
42. Single Family Units	4.918	1.93e-06
43. Renters Spending More Than 30% of Income on Housing	5.538	1.04e-07

44. Single Mothers with Children in Poverty	12.189	< 2e-16
50. Births Non Latino White	-3.576	0.000446

To verify the new reduced model is an improvement on the full model, I run a K=10 folds cross validation. Table 3 shows the results from cross validation of the 2 models.

Table 3

Cross Validation of Full and Reduced Model			
	Root Mean Square Error	R Squared	Mean Absolute Error
Full Model	0.0410	0.809	0.029
10 Predictor Model	0.0397	0.828	0.026

The table shows that the reduced model has improved R squared, Root Mean Square Error and Mean Absolute Error over the full model.

Now that variable selection has been determined, I check the assumptions of linear regression.

Figure 2.

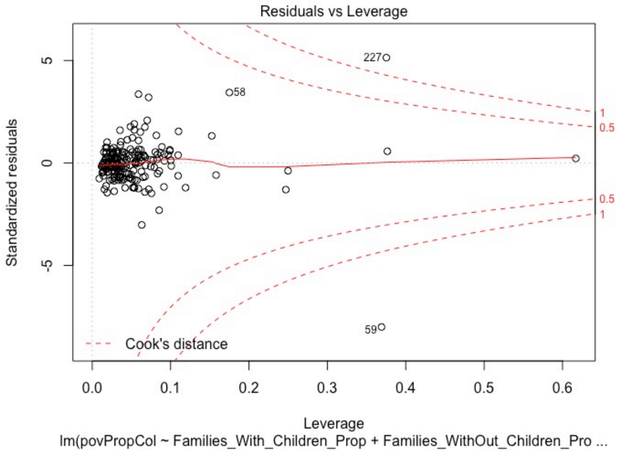
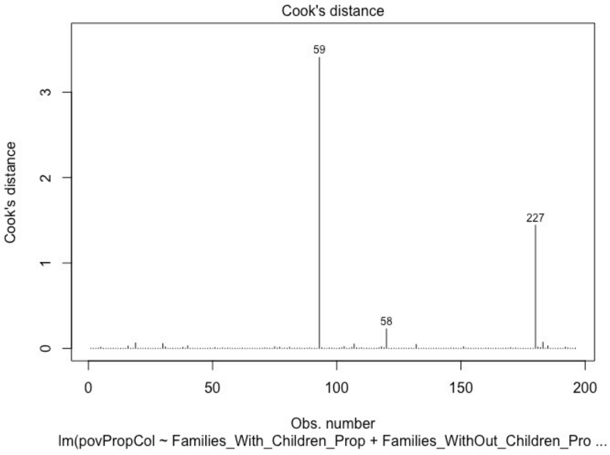


Figure 3.



Checking residual plots, component-residual plots and CERES plots, I find that assumptions look pretty good except for a few possible outliers and influential points. Figure 2 and 3 show the cook's distance, a statistics to quantify the influence of each observation.

Figure 2 is a residual vs. leverage plot where the red dashed lines indicate cook's distance. Observations outside the cook's distance lines are to be considered influential. In this case, observations 227 and 59 are influential. Figure 3 shows cook's distance by observations number. This figure shows observation 227 and 59 with a cook's distance much larger than the other observation as well as the observations being bigger than one. Physically looking at observation 59 and 227, I see that those neighborhoods are CU East Campus and University. These observations make sense as influential points since there is not a standard population living in these neighborhoods. Most of the CU East Campus and University are the school campus and do not have much housing outside of dorms. For this reason and the evidence provided by cook's distance I chose to remove these two influential points.

After the removal of the influential points, I recheck variance inflation factors, significance of each predictor, and cross validation. The only change being that hypothesis testing for the significance of Housing Vacancy now supports the null hypothesis

Figure 4. Component Residual Plot Before Removal of Influential Obs.

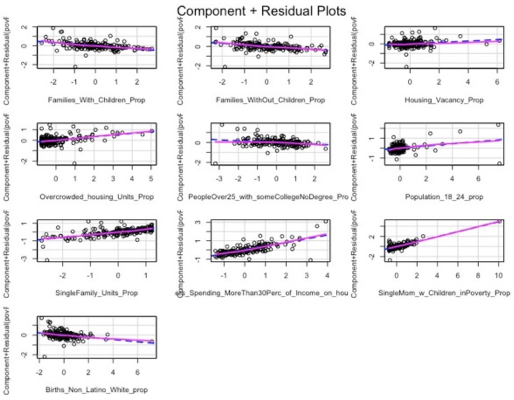
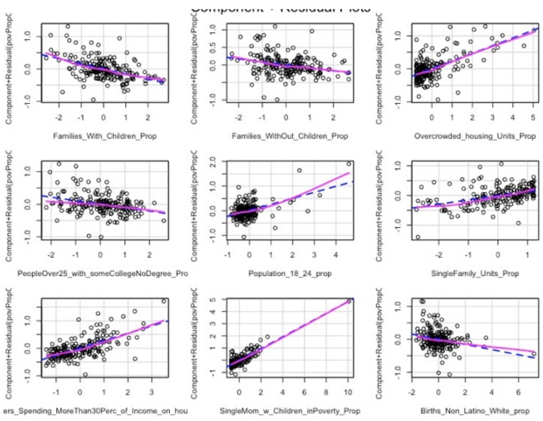


Figure 5. Component Residual Plot After Removal of Influential Obs.



$H_0: \hat{\beta}_{Housing\ vacancy} = 0$ with a p-value of 0.110. Since the variable is no longer significant, it is removed from the model. After the removal of the predictor and influential points I recheck the residual plots, component residual plots and CERES plots. Figure 4 and 5 show the component residual plots before and after the removal of influential points, respectively. Both figures look like assumptions are being met and transformations are not necessary. Although influential points were removed, there does still seem to be some outliers affecting the component residual plots. However, these points will not be removed like the influential points.

Figure 6 is a plot of residuals vs. fitted values. This plot is useful for checking the linearity and homoscedasticity assumptions. In order to meet the assumptions, the red line should be straight and horizontal to meet the linearity assumption as well as there should be residuals evenly spread out from y=0 to meet the homoscedasticity assumption. Figure 6 is not perfect, but the residuals do appear to have a fairly even spread across y=0. The red line is not exactly straight but this plot does not have any severe causes for concern, so the linearity and homoscedasticity assumptions are met.

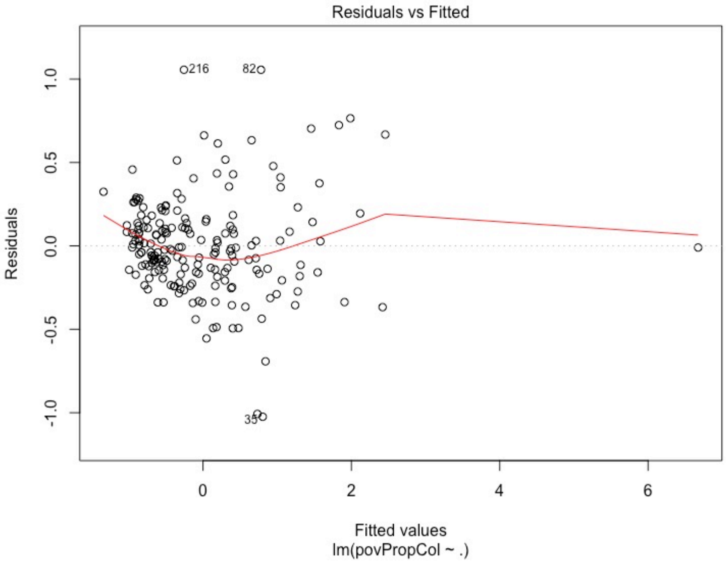


Figure 6.

Figures 7 and 8 are QQ plots to test the normality assumption. Figure 7 is from before the removal of the influential points and figure 8 is from after the removal. You can see from figure 8 that removing the influential points helps the normality assumption. Although the QQ plots show some heavy tail tendencies this is fairly common in observational data and is not a big enough problem for concern about meeting the normality assumption.

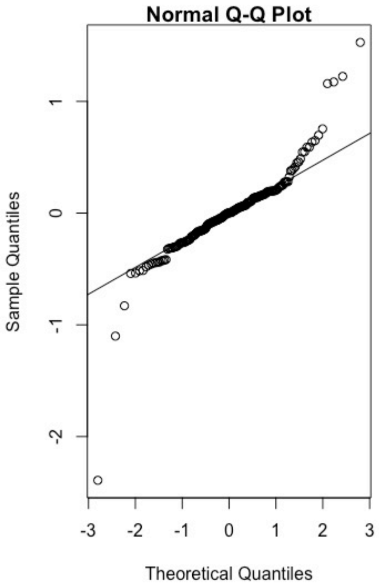


Figure 7. QQ Plot Before Removal of Influential Obs.

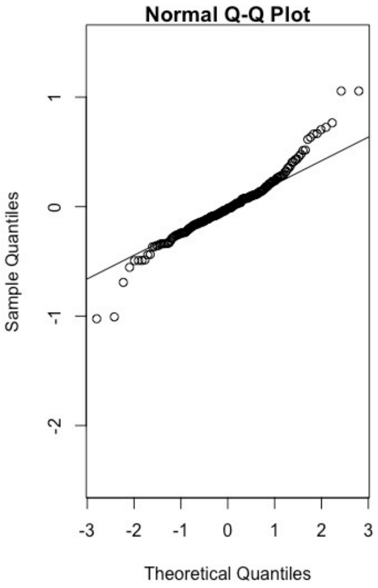


Figure 8. QQ Plot After Removal of Influential Obs.

The last assumption to check is for autocorrelation. This is done with a Durban Watson Test. This test a null hypothesis $H_{0:}$ No autocorrelation among residuals against the alternative hypothesis that there is autocorrelation among residuals. For the model I find the p value=0.082 for the Durban Watson test. This suggest there is evidence in favor of the Null Hypothesis. This means the assumption of no autocorrelation is met.

Therefore, the model with 9 predictors is shown to meet all assumptions of linear regression. In addition, all predictors are significant to the model and the model has been shown through cross validation to improve upon full model.

Ridge Regression

While the model with 9 predictors is shown to improve upon the full model, regressing only 9 predictors of the original 51 means that about 80% of the data has been removed. Although some of the information in the removed predictors is actually contained in the remaining predictors, it would be beneficial to run some models with the full data set that wasn't subject to the many assumptions of linear regression. To do this I run three other regression models, specifically, Ridge, Lasso, and Partial Least Squares Regression.

Ridge regression is a form of penalized linear regression where the residual sums of squares is penalized using an L2 norm penalty. This increases the bias a little bit for a bigger decrease in the variance. This regression method is specifically used for data with multicollinearity. For ridge regression, a tuning parameter λ is calculated through cross validation which affects the amount the model is penalized. For my model, using a K=10 folds cross validation I find that $\lambda_{min} = 0.08776$. Note that $\lambda = 0$ indicates no penalty and therefore the ridge regression would be equal to ordinary least squares linear regression.

Lasso Regression

Lasso regression is another form of penalized linear regression where the residual sums of squares is penalized using an L1 norm instead of an L2 norm as in ridge regression. Although this method is very similar to ridge, the big difference is that Lasso can shrink coefficients to exactly zero while ridge cannot and therefore has variable selection built in to the model. Again, I used K=10 folds cross validation to find the best tuning parameter $\lambda_{min} = 0.00047$. The variables not shrunk to zero in the lasso regression model were 3. Persons per Household, 4. Total Population, 5. Crime Incidents, 12. Families with Children, 13. Families without Children, 16. Households with Income less than 60k, 17. Households with Income between 60k and 125k, 18. Housing Vacancy, 19. Multifamily Units, 22. People Over 25 with High School Only, 23. People Over 25 with less than 12th Grade Education, 24. People Over 25 with Some College No Degree, 25. Population 18-24, 32. Population Asian and Pacific Islander, 34. Population Native American, 39. Population of 2 or More Races, 40. Population over 65, 41. Population under 5, 43. Renters Spending More Than 30% of Income on Housing, 44. Single Mothers with Children in Poverty, 46. Births African American, 47. Births Asian per Pacific Islander, 49. Births Native American, and 50. Births Non Latino White.

The Lasso Regression model leaves 24 predictors variables in the model. Note that this includes all of the variables remaining from in the ordinary least squares except for 2, Overcrowded Housing and Single Family Units. This provides some validation to the predictors chosen for the ordinary least squares model since the Lasso regression also finds them important.

Partial Least Squares

Partial Least Squares is a regression method meant for high dimensional data. It transforms the predictors into principle components that are linear combinations of the original variables. The direction of the principle components is related to the response and components with most closely related to the response are given the most weight. The number of components used is determined through a plot of mean square prediction error vs. number of components shown in figure 9. The vertical line shows the best number of components to use as it is the smallest number of components with the smallest mean square prediction error. Using all 51 components would be the same as running the ordinary least squares linear regression model.

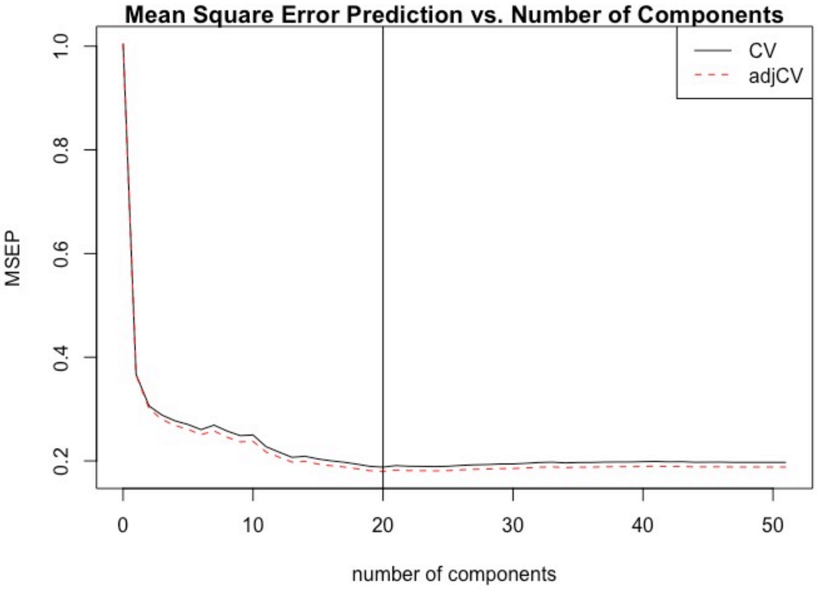


Figure 9.

Results

Table 4 shows the results of the ordinary least squares regression. Since the variables have been standardized interpretation of the magnitude is not relevant. However, we can interpret some interesting results from looking at the sign of the coefficients.

Table 4.

Ordinary Least Squares Results			
Predictor	Coefficient	Standard Error	P-Value
12. Families with Children	-0.158	0.036	1.89e-05
13. Families without Children	-0.085	0.040	0.035
20. Overcrowded Housing Units	0.233	0.040	1.83e-08
24. People Over 25 with Some College No Degree	-0.107	0.026	5.89e-05
25. Population 18-24	0.244	0.046	3.04e-07
42. Single Family Units	0.143	0.052	0.006
43. Renters Spending More Than 30% of Income on Housing	0.267	0.059	1.20e-05
44. Single Mothers with Children in Poverty	0.481	0.032	< 2e-16
50. Births Non Latino White	-0.079	0.026	0.003

One surprising result is that both families with children and families without children are included in the model with a negative coefficient. Physically, this means that as families with children and families without decreases the proportion of people in poverty increases. It seems like a strange result but one thing to note here is that families without children has a p-value of 0.035. While I was evaluating significant at the standard $\alpha_{crit} = 0.05$ if I were to evaluate at a stricter metric like $\alpha_{crit} = 0.025$ then families without children would no longer be significant to the model.

Other results made more sense such as population 18-24, renters spending more than 30% of income on housing, and single mothers with children in poverty. All of these had positive coefficients meaning that as population 18-24 or renters spending more than 30% of income on housing or single mothers with children in poverty increased so did poverty levels. These are unsurprising results since, for example, population age 18-24 tends to have the lowest wealth since they are just starting in the work force or renters spending more than 30% of income on housing don't tend to do so out of choice but rather because of the amount of total wealth they have relative to housing prices.

The ultimate goal of this project was prediction and Table 5 shows the prediction error results for the 4 models.

Table 5

Prediction Results				
	Root Mean Square Error	Mean Absolute Error	Adjusted R Squared	Expected Prediction Error

Ordinary Least Squares	0.555	0.374	0.699	0.308
Ridge	0.465	0.317	0.781	0.216
Lasso	0.447	0.310	0.797	0.200
Partial Least Squares	0.444	0.316	0.800	0.197

Between the 4 models, Partial Least squares had the best prediction results in three of the four metric, Root Mean Square Error, Adjusted R Square and Expected Prediction Error. Lasso was best in the Mean Absolute Error and it was also very close to Partial Least Squares in the other categories. These results make sense because partial least squares is meant for dealing with lots of predictors and low observations which is what we have here in this data set. However, the downside for partial least squares is that since it uses principle component you do not have the same inference capabilities of the other regression methods. Lasso would be best choice because it did as well as partial least squares but has the interpretability similar to ordinary least squares which allows us to see which coefficients are significant to the model.

Going Forward

It would be interesting to see if there is any significant interaction between population 18-24 and other variables such as overcrowded housing, renters spending more than 30% of income on housing and people over 25 with some but no college degree because to me, based on the results in the analysis and experience living in Denver, it seems the largest contributor to the poverty levels is people in the 18-24 age range and the high cost of living . In addition, more data would have to be collected to make any significant conclusions. This data set is a very small sample from Denver neighborhoods only. In order to make any real conclusions more observations would be needed. These results could definitely not be used for inference on anything other than the specific Denver neighborhoods that were sampled and would not be useful to make inference on other cities in the US.

Other analysis could provide more insight as well such as performing a Poisson regression with the variables in their original forms as counts instead of proportions. Adding spatial and temporal data could provide a lot more information such as where specifically in the neighborhoods are the highest poverty levels.

References

Weisberg, S. (2014). *Applied linear regression* (4th ed.). Hoboken, NJ: Wiley.

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. (2013). An introduction to statistical learning : with applications in R. New York :Springer,

Faraway, J. J. (2014). *Linear models with R* (2nd ed.). Boca Raton ; London ; New York: CRC Press.

Appendix R Code

Load and Clean Data

```r
houseDat<-read.csv("housingreduced.csv")
houseDat=houseDat[,-1]
## Fill Missing Crime incidents with mean of crime incidents.
for(i in 1:length(houseDat$Crime.Incidents)){

if(is.na(houseDat[i,52])==TRUE){
  houseDat[i,52]=mean(houseDat$Crime.Incidents, na.rm=TRUE)

}}

## Change People in poverty to propiortion of total population
povPropCol<-houseDat$Persons_in_Poverty/houseDat$Total_Population
houseDat<-cbind(houseDat, povPropCol)
houseDat$Persons_in_Poverty<-NULL

hist(povPropCol)

## One Person Households to proportion
One_Person_House_Prop<-houseDat$One_Person_Households/houseDat$Total_Population
houseDat<-cbind(houseDat, One_Person_House_Prop)
houseDat$One_Person_Households<-NULL

## Adults non English Speaking to proportion
Adults_Non_English_Speaking_Prop<-houseDat$Adults_Non_English_Speaking/houseDat$Total_Population
houseDat<-cbind(houseDat, Adults_Non_English_Speaking_Prop)
houseDat$Adults_Non_English_Speaking<-NULL

## Births to Teen Mothers to proportion
Births_to_Teen_Mom_Prop<-houseDat$Births_to_Teen_Mothers_19_and_under/houseDat$Total_Population
houseDat<-cbind(houseDat, Births_to_Teen_Mom_Prop)
houseDat$Births_to_Teen_Mothers_19_and_under<-NULL

## Births to Unwed Mother to proportion
Births_to_Unwed_Mom_Prop<-houseDat$Births_to_Unwed_Mothers/houseDat$Total_Population
houseDat<-cbind(houseDat, Births_to_Unwed_Mom_Prop)
houseDat$Births_to_Unwed_Mothers<-NULL

## Births to Women Less than 12th grade education to proportion
Births_to_Women_LessThan12thGradeEdu_Prop<-houseDat$Births_to_Women_With_Less_Than_12th_Grade_Education/houseDat$Total_Population
houseDat<-cbind(houseDat, Births_to_Women_LessThan12thGradeEdu_Prop)
houseDat$Births_to_Women_With_Less_Than_12th_Grade_Education<-NULL
```

```r
## Children Living with Single Parents to proportion
Children_Living_With_SingleParentProp<-houseDat$Children_Living_with_Single_P
arents/houseDat$Total_Population
houseDat<-cbind(houseDat, Children_Living_With_SingleParentProp)
houseDat$Children_Living_with_Single_Parents<-NULL

## Families with Children as a proportion of households
Families_With_Children_Prop<-houseDat$Families_with_Children/houseDat$Househo
lds
houseDat<-cbind(houseDat, Families_With_Children_Prop)
houseDat$Families_with_Children<-NULL

## Families w/o children as a proportion of households
Families_WithOut_Children_Prop<-houseDat$Families_without_Children/houseDat$H
ouseholds
houseDat<-cbind(houseDat, Families_WithOut_Children_Prop)
houseDat$Families_without_Children<-NULL

## Foreign Born to proportion
Foreign_Born_Prop<-houseDat$Foreign_Born/houseDat$Total_Population
houseDat<-cbind(houseDat, Foreign_Born_Prop)
houseDat$Foreign_Born<-NULL

## Households w/ income more than 125k to prop
Households_W_Income_more_than_125k_Prop<-houseDat$Households_with_Income_more
_than_USD_125000/houseDat$Households
houseDat<-cbind(houseDat, Households_W_Income_more_than_125k_Prop)
houseDat$Households_with_Income_more_than_USD_125000<-NULL

## Households w/ income less than 60k to prop
Households_W_Income_less_than_60k_Prop<-houseDat$Households_with_Income_less_
than_USD_60000/houseDat$Households
houseDat<-cbind(houseDat, Households_W_Income_less_than_60k_Prop)
houseDat$Households_with_Income_less_than_USD_60000<-NULL

## Households w/ income less than 60k to prop
Households_W_Income_between_60k_And_125k_Prop<-houseDat$Households_with_Incom
e_USD_60000_to_USD_124999/houseDat$Households
houseDat<-cbind(houseDat, Households_W_Income_between_60k_And_125k_Prop)
houseDat$Households_with_Income_USD_60000_to_USD_124999<-NULL

## Housing Vacancy as proportion of housing units
Housing_Vacancy_Prop<-houseDat$Housing_Vacancy/houseDat$Housing_Units
houseDat<-cbind(houseDat, Housing_Vacancy_Prop)
houseDat$Housing_Vacancy<-NULL

## Multi Family Units as proportion of housing units
Multifamiliy_Units_Prop<-houseDat$Multi_Family_Units/houseDat$Housing_Units
houseDat<-cbind(houseDat, Multifamiliy_Units_Prop)
```

```r
houseDat$Multi_Family_Units<-NULL

## Overcrowded housing Units as proportion of housing units
Overcrowded_housing_Units_Prop<-houseDat$Overcrowded_housing_units/houseDat$H
ousing_Units
houseDat<-cbind(houseDat, Overcrowded_housing_Units_Prop)
houseDat$Overcrowded_housing_units<-NULL

## PeopleOver25 with College Associates or Better to proportion
PeopleOver25_with_College_Associates_Prop<-houseDat$Persons_Age_25_with_Colle
ge_Degree_Associates_or_Better/houseDat$Total_Population
houseDat<-cbind(houseDat, PeopleOver25_with_College_Associates_Prop)
houseDat$Persons_Age_25_with_College_Degree_Associates_or_Better<-NULL

## PeopleOver25 with high school only to proportion
PeopleOver25_with_highschool_only_Prop<-houseDat$Persons_age_25_with_high_sch
ool_only_education/houseDat$Total_Population
houseDat<-cbind(houseDat, PeopleOver25_with_highschool_only_Prop)
houseDat$Persons_age_25_with_high_school_only_education<-NULL

## PeopleOver25 with less than 12th grade edu to proportion
PeopleOver25_with_lessThan12thGradeEdu_Prop<-houseDat$Persons_age_25_with_les
s_than_12th_grade_education/houseDat$Total_Population
houseDat<-cbind(houseDat, PeopleOver25_with_lessThan12thGradeEdu_Prop)
houseDat$Persons_age_25_with_less_than_12th_grade_education<-NULL

## PeopleOver25 with some College No Degree to proportion
PeopleOver25_with_someCollegeNoDegree_Prop<-houseDat$Persons_age_25_with_some
_college_but_no_degree/houseDat$Total_Population
houseDat<-cbind(houseDat, PeopleOver25_with_someCollegeNoDegree_Prop)
houseDat$Persons_age_25_with_some_college_but_no_degree<-NULL

## Population Age columns to Proportion
popAgeDf<-matrix(0,nrow = 264, ncol = 6)
PopNamesCol<-c("Population_18_24_prop","Population_25_34_prop","Population_35
_44_prop","Population_45_54_prop","Population_5_17_prop", "Population_55_64_p
rop" )
for (i in 4:9) {
  popAgeDf[,(i-3)]<-houseDat[,i]/houseDat$Total_Population

}
popAgeDf<-as.data.frame(popAgeDf)
names(popAgeDf)<-PopNamesCol
houseDat<-cbind(houseDat, popAgeDf)
houseDat$Population_18_24<-NULL
houseDat$Population_25_34<-NULL
houseDat$Population_35_44<-NULL
houseDat$Population_45_54<-NULL
houseDat$Population_55_64<-NULL
houseDat$Population_5_17<-NULL
```

```
## Population Race columns to Proportion
popRaceDf<-matrix(0,nrow = 264, ncol = 11)
PopRaceCol<-c("Population_African_American_prop", "Population_Asian_per_Pacif
ic_Islander_prop", "Population_Latino_prop", "Population_Native_American_prop
", "Population_Non_Latino_prop", "Population_Non_Latino_White_prop", "Populat
ion_Other_Race_Single_Race_Selected_prop", "Population_White_prop", "Populati
on_of_2_or_More_Races_prop", "Population_over_65_prop", "Population_under_5_p
rop")
for (i in 4:14) {
  popRaceDf[,(i-3)]<-houseDat[,i]/houseDat$Total_Population

}
popRaceDf<-as.data.frame(popRaceDf)
names(popRaceDf)<-PopRaceCol
houseDat<-cbind(houseDat, popRaceDf)
houseDat$Population_African_American<-NULL
houseDat$Population_Asian_per_Pacific_Islander<-NULL
houseDat$Population_Latino<-NULL
houseDat$Population_Native_American<-NULL
houseDat$Population_Non_Latino<-NULL
houseDat$Population_Non_Latino_White<-NULL
houseDat$Population_Other_Race_Single_Race_Selected<-NULL
houseDat$Population_White<-NULL
houseDat$Population_of_2_or_More_Races<-NULL
houseDat$Population_over_65<-NULL
houseDat$Population_under_5<-NULL

## Single Family Units as proportion of housing units
SingleFamily_Units_Prop<-houseDat$Single_Family_Units/houseDat$Housing_Units
houseDat<-cbind(houseDat, SingleFamily_Units_Prop)
houseDat$Single_Family_Units<-NULL

## Renters Spending More Than 30% income on Housingas prop of total household
s
Renters_Spending_MoreThan30Perc_of_Income_on_housing_Prop<-houseDat$Renters_S
pending_More_Than_30Percent_of_Income_on_Housing/houseDat$Households
houseDat<-cbind(houseDat, Renters_Spending_MoreThan30Perc_of_Income_on_housin
g_Prop)
houseDat$Renters_Spending_More_Than_30Percent_of_Income_on_Housing<-NULL


## Single Mothers w. Children in Poverty to proportion
SingleMom_w_Children_inPoverty_Prop<-houseDat$Single_Mothers_with_Children_in
_Poverty/houseDat$Total_Population
houseDat<-cbind(houseDat, SingleMom_w_Children_inPoverty_Prop)
houseDat$Single_Mothers_with_Children_in_Poverty<-NULL

## Unemployed and Births to proportion of total pop
```

```r
birthDf<-matrix(0,nrow = 264, ncol = 7)
birthCol<-c("Unemployed_in_Civilian_Labor_Force_prop", "Births_African_Americ
an_prop", "Births_Asian_per_Pacific_Islander_prop","Births_Latino_prop", "Bir
ths_Native_American_prop", "Births_Non_Latino_White_prop", "Births_Other_Race
_prop")
for (i in 5:11) {
  birthDf[,(i-4)]<-houseDat[,i]/houseDat$Total_Population

}
birthDf<-as.data.frame(birthDf)
names(birthDf)<-birthCol
houseDat<-cbind(houseDat, birthDf)
houseDat$Unemployed_in_Civilian_Labor_Force<-NULL
houseDat$Births_African_American<-NULL
houseDat$Births_Asian_per_Pacific_Islander<-NULL
houseDat$Births_Latino<-NULL
houseDat$Births_Native_American<-NULL
houseDat$Births_Non_Latino_White<-NULL
houseDat$Births_Other_Race<-NULL

## Look for NAs
for (i in 1:52) {print(which(is.na(houseDat[,i])))}
## Rows 182 and 78 were Nan remove them
houseDat<-houseDat[-182,]
houseDat<-houseDat[-78,]

## Train Test Split
set.seed(123) # Set Seed so that same sample can be reproduced in future
# Now Selecting 75% of data as sample from total 'n' rows of the data
smp_siz = floor(0.75*nrow(houseDat))  # creates a value for dividing the data
into train and test. In this case the value is defined as 75% of the number o
f rows in the dataset
smp_siz
train_ind = sample(seq_len(nrow(houseDat)),size = smp_siz)
Train =houseDat[train_ind,] #creates the training dataset
Test=houseDat[-train_ind,] # creates the testing data set

## Create standardized version of the data set
Train.Std<-scale(Train)
Train.Std<-as.data.frame(Train.Std)
Test.Std<-scale(Test)
Test.Std<-as.data.frame(Test.Std)
## check to make sure standardized
colMeans(Test.Std)
colMeans(Train.Std)
apply(Train.Std, 2, sd)
apply(Test.Std, 2, sd)

## Check Data Types
for (i in 1:52) {
  print(typeof(Train.Std[,i]))
}
for (i in 1:52) {
  print(typeof(Train.Std[,i]))
}
```

Linear Regression, Initial Fit, Collinearity and Diagnostics

```
mod1<-lm(povPropCol~., Train.Std)
## Removing NA columns becuase of singularities in lm

## Remove Singularities: Population of 2 or more races, Population over 65, P
op under 5, households w/ income between 60k and 125k, Population Non Latino
mod1<-lm(povPropCol~.-Households_W_Income_between_60k_And_125k_Prop  -Populat
ion_Non_Latino_prop -Population_of_2_or_More_Races_prop -Population_over_65_p
rop -Population_under_5_prop, Train.Std)
mod1.Sum<-summary(mod1)

## Save standardized train data set without singularities
Train2.Std<-Train.Std
Train2.Std$Households_W_Income_between_60k_And_125k_Prop<-NULL
Train2.Std$Population_Non_Latino_prop<-NULL
Train2.Std$Population_of_2_or_More_Races_prop<-NULL
Train2.Std$Population_over_65_prop<-NULL
Train2.Std$Population_under_5_prop<-NULL

Test2.Std<-Test.Std
Test2.Std$Households_W_Income_between_60k_And_125k_Prop<-NULL
Test2.Std$Population_Non_Latino_prop<-NULL
Test2.Std$Population_of_2_or_More_Races_prop<-NULL
Test2.Std$Population_over_65_prop<-NULL
Test2.Std$Population_under_5_prop<-NULL

## Check for Collinearity
library(car)

## Loading required package: carData

vif(mod1)
## There is high collinearity among many of the predictors predictors

## See what AIC step wise selection will remove
library(MASS)
stepAIC(mod1, direction="backward")

## StepAIC backward selection suggested variables
mod2<-lm(formula = povPropCol ~ Households + Persons_per_Household +
    Total_Population + One_Person_House_Prop + Families_With_Children_Prop +
    Families_WithOut_Children_Prop + Foreign_Born_Prop + Households_W_Income_
less_than_60k_Prop +
    Housing_Vacancy_Prop + Multifamiliy_Units_Prop + Overcrowded_housing_Unit
```

```
s_Prop +
    PeopleOver25_with_College_Associates_Prop + PeopleOver25_with_highschool_
only_Prop +
    PeopleOver25_with_lessThan12thGradeEdu_Prop + PeopleOver25_with_someColle
geNoDegree_Prop +
    Population_18_24_prop + Population_25_34_prop + Population_35_44_prop +
    Population_45_54_prop + Population_5_17_prop + Population_55_64_prop +
    Population_Native_American_prop + Population_Non_Latino_White_prop +
    SingleFamily_Units_Prop + Renters_Spending_MoreThan30Perc_of_Income_on_ho
using_Prop +
    SingleMom_w_Children_inPoverty_Prop + Unemployed_in_Civilian_Labor_Force_
prop +
    Births_African_American_prop + Births_Asian_per_Pacific_Islander_prop +
    Births_Latino_prop + Births_Non_Latino_White_prop, data = Train.Std)
stepAIC(mod1, direction="both")
## StepAIC both direction selection variables (same as backward)

## Save variables not thrown out as Train3.Std
Train3.Std = subset(Train2.Std, select = c(Households,                    Per
sons_per_Household, Total_Population,                                  One_Pe
rson_House_Prop, Families_With_Children_Prop,                              Fa
milies_WithOut_Children_Prop, Foreign_Born_Prop,
Households_W_Income_less_than_60k_Prop, Housing_Vacancy_Prop,
Multifamiliy_Units_Prop, Overcrowded_housing_Units_Prop,         PeopleOver25
_with_College_Associates_Prop,        PeopleOver25_with_highschool_only_Prop,
PeopleOver25_with_lessThan12thGradeEdu_Prop,         PeopleOver25_with_someCo
llegeNoDegree_Prop, Population_18_24_prop,                   Population_25_34
_prop, Population_35_44_prop,                                  Population_45_
54_prop, Population_5_17_prop,                                    Population_
55_64_prop, Population_Native_American_prop,                         Populati
on_Non_Latino_White_prop, SingleFamily_Units_Prop,                         Re
nters_Spending_MoreThan30Perc_of_Income_on_housing_Prop, SingleMom_w_Children
_inPoverty_Prop, Unemployed_in_Civilian_Labor_Force_prop,        Births_Afric
an_American_prop, Births_Asian_per_Pacific_Islander_prop, Births_Latino_prop,
Births_Non_Latino_White_prop, povPropCol))

Test3.Std = subset(Test2.Std, select = c(Households,                    Perso
ns_per_Household, Total_Population,                                  One_Pers
on_House_Prop, Families_With_Children_Prop,                              Fami
lies_WithOut_Children_Prop, Foreign_Born_Prop,
Households_W_Income_less_than_60k_Prop, Housing_Vacancy_Prop,
Multifamiliy_Units_Prop, Overcrowded_housing_Units_Prop,         PeopleOver25
_with_College_Associates_Prop,        PeopleOver25_with_highschool_only_Prop,
PeopleOver25_with_lessThan12thGradeEdu_Prop,         PeopleOver25_with_someCo
llegeNoDegree_Prop, Population_18_24_prop,                   Population_25_34
_prop, Population_35_44_prop,                                  Population_45_
54_prop, Population_5_17_prop,                                    Population_
55_64_prop, Population_Native_American_prop,                         Populati
on_Non_Latino_White_prop, SingleFamily_Units_Prop,                         Re
nters_Spending_MoreThan30Perc_of_Income_on_housing_Prop, SingleMom_w_Children
_inPoverty_Prop, Unemployed_in_Civilian_Labor_Force_prop,        Births_Afric
an_American_prop, Births_Asian_per_Pacific_Islander_prop, Births_Latino_prop,
Births_Non_Latino_White_prop, povPropCol))
summary(mod2)
vif(mod2)

## Correlation Plot
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
correlTrain<-Train
numcolnames<-seq(1, 52)
colnames(correlTrain)=numcolnames

correlat<-cor(correlTrain)

corrplot::corrplot(correlat, type = "upper", order = "hclust",
         tl.col = "black", tl.cex = 0.8, cl.cex=0.8)


## Remove High VIF one at a time by highest vif number
mod2 <- update(mod2, . ~ . - PeopleOver25_with_College_Associates_Prop)
VifNums<-vif(mod2)
VifNums
VifNums<-sort(VifNums, decreasing = TRUE)
VifNums[1]

mod2 <- update(mod2, . ~ . - Total_Population)
VifNums<-vif(mod2)
VifNums
VifNums<-sort(VifNums, decreasing = TRUE)
VifNums[1]

mod2<- update(mod2, . ~ . - Multifamiliy_Units_Prop)
VifNums<-vif(mod2)
VifNums
VifNums<-sort(VifNums, decreasing = TRUE)
VifNums[1]

mod2<- update(mod2, . ~ . - Population_Non_Latino_White_prop)
VifNums<-vif(mod2)
VifNums
VifNums<-sort(VifNums, decreasing = TRUE)
VifNums[1]

mod2<- update(mod2, . ~ . - Persons_per_Household)
VifNums<-vif(mod2)
VifNums
VifNums<-sort(VifNums, decreasing = TRUE)
```

```r
VifNums[1]

mod2<- update(mod2, . ~ . - Households_W_Income_less_than_60k_Prop)
VifNums<-vif(mod2)
VifNums
VifNums<-sort(VifNums, decreasing = TRUE)
VifNums[1]

mod2<- update(mod2, . ~ . - One_Person_House_Prop)
VifNums<-vif(mod2)
VifNums
VifNums<-sort(VifNums, decreasing = TRUE)
VifNums[1]

mod2<- update(mod2, . ~ . - PeopleOver25_with_lessThan12thGradeEdu_Prop)
VifNums<-vif(mod2)
VifNums
VifNums<-sort(VifNums, decreasing = TRUE)
VifNums[1]

mod2<- update(mod2, . ~ . - Population_5_17_prop)
VifNums<-vif(mod2)
VifNums
VifNums<-sort(VifNums, decreasing = TRUE)
VifNums[1]

## All VIFs now under 10

summary(mod2)

## Rerun step AIC
stepAIC(mod2, direction = "backward")

## new model of suggested variables
mod3<-lm(formula = povPropCol ~ Families_With_Children_Prop + Families_WithOu
t_Children_Prop +
    Housing_Vacancy_Prop + Overcrowded_housing_Units_Prop + PeopleOver25_with
_someCollegeNoDegree_Prop +
    Population_18_24_prop + SingleFamily_Units_Prop + Renters_Spending_MoreTh
an30Perc_of_Income_on_housing_Prop +
    SingleMom_w_Children_inPoverty_Prop + Births_Non_Latino_White_prop,
    data = Train.Std)
summary(mod3)

## Cross Validation
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2
```

```r
cv_10fold = trainControl(method = "cv", number = 10)
f1 = formula(mod1, data=Train)
f2 = formula(mod3, data=Train)
modela = train(f1, data = Train, trControl = cv_10fold,
               method = "lm")
modelb = train(f2, data = Train, trControl = cv_10fold,
               method = "lm")
print(modela) # full, 10-fold
print(modelb) # reduced, 10-fold


## All variables in mod3 are significant
## Saving remaining varibles as Train4.Std and Test4.Std
Train4.Std<-subset(Train.Std, select = c(Families_With_Children_Prop,
Families_WithOut_Children_Prop, Housing_Vacancy_Prop,             Overcrowded
_housing_Units_Prop, PeopleOver25_with_someCollegeNoDegree_Prop, Population_1
8_24_prop, SingleFamily_Units_Prop,                         Renters_Spending_
MoreThan30Perc_of_Income_on_housing_Prop, SingleMom_w_Children_inPoverty_Prop
, Births_Non_Latino_White_prop, povPropCol))

Test4.Std<-subset(Test.Std, select = c(Families_With_Children_Prop,
Families_WithOut_Children_Prop, Housing_Vacancy_Prop,             Overcrowded
_housing_Units_Prop, PeopleOver25_with_someCollegeNoDegree_Prop, Population_1
8_24_prop, SingleFamily_Units_Prop,                         Renters_Spending_
MoreThan30Perc_of_Income_on_housing_Prop, SingleMom_w_Children_inPoverty_Prop
, Births_Non_Latino_White_prop, povPropCol))

## Checking structure
residualPlots(mod3)

crPlots(mod3)

ceresPlots(mod3)

## Plots look good except there might be some outliers/influence points

## More Diagnostics
plot(mod3)

influenceIndexPlot(mod3)

## both 59 and 227 appear to be influential
##Note the row names are not the same as the row, so row 59 is not named 59 b
ecuase of sampling for train, test
## Row Name 59 and Row Name 227 are the points in question
Train4.Std["59",]
Train["59",]
Train4.Std["227",]
Train["227",]

## Removing Influential Outliers 227 and 59 to see what happens
```

```
rownames(Train4.Std)[c(180, 93)]->remove


Train5.Std<-Train4.Std[!rownames(Train4.Std) %in% remove, ]

## Now Refit and check diagnostics
mod4<-lm(formula = povPropCol ~ .,  data = Train5.Std)
vif(mod4)
plot(mod4)

influencePlot(mod4)

influenceIndexPlot(mod4)

## Still not perfect but now none of the observations are outside cooks D ban
ds in plots and not influential

## Housing Vacany no longer significant after influential points are removed

Train5.Std$Housing_Vacancy_Prop<-NULL
Test5.Std<-Test4.Std
Test5.Std$Housing_Vacancy_Prop<-NULL
mod4<-lm(formula = povPropCol ~ .,  data = Train5.Std)
summary(mod4)
vif(mod4)

residualPlots(mod4)

crPlots(mod4)

ceresPlots(mod4)

qqnorm(mod4$residuals)

plot(mod4)

durbinWatsonTest(mod4)
```

Prediction for OLS

```
set.seed(123)
yTest<-Test5.Std$povPropCol
lmod.pred<-predict(mod4, Test5.Std)
lmod.MSE=mean((lmod.pred-yTest)^2)

library(tidyverse)

## ── Attaching packages ───────────────────── tidyverse 1.2.1 ──

## ✓ tibble  1.4.2     ✓ purrr   0.2.5
## ✓ tidyr   0.8.2     ✓ dplyr   0.7.8
## ✔ readr   1.1.1     ✔ stringr 1.3.1
## ✔ tibble  1.4.2     ✔ forcats 0.3.0

## ── Conflicts ──────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ✖ purrr::lift()   masks caret::lift()
## ✖ dplyr::recode() masks car::recode()
## ✖ dplyr::select() masks MASS::select()
## ✖ purrr::some()   masks car::some()

library(caret)
predictions <- mod4 %>% predict(Test5.Std)
lmod.results=data.frame(
  R2 = caret::R2(predictions, Test5.Std$povPropCol),
  RMSE = RMSE(predictions, Test5.Std$povPropCol),
  MSE = (RMSE(predictions, Test5.Std$povPropCol)^2),
  MAE = MAE(predictions, Test5.Std$povPropCol)
)
colnames(lmod.results)<-c("R2","RMSE", "MSE", "MAE")
```

Ridge Regression

```
set.seed(123)
library(glmnet)

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following object is masked from 'package:tidyr':
##
##     expand

## Loading required package: foreach

##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when

## Loaded glmnet 2.0-16

grid=10^seq(10,-2,length=100)
xTrain.Full=model.matrix(povPropCol~.,Train.Std)[,-1]
yTrain.Full=Train.Std$povPropCol
xTest.Full=model.matrix(povPropCol~.,Test.Std)[,-1]
yTest.Full=Test.Std$povPropCol
## Cross Validate to find the best lambda tuning parameter
```

```r
cv.out.ridge<-cv.glmnet(xTrain.Full, yTrain.Full, alpha=0)
plot(cv.out.ridge)

bestLam.ridge<-cv.out.ridge$lambda.min
bestLam.ridge
## Note that Lambda=0 corresponds to ordinary least squares regression so bes
t lambda=0.0877 isn't that much different
ridge.mod=glmnet(xTrain.Full, yTrain.Full, alpha=0, lambda=grid, thresh=1e-12
)
ridge.pred=predict(ridge.mod, s=bestLam.ridge, newx = xTest.Full)
ridge.MSE=mean((ridge.pred-yTest.Full)^2)
ridge.results=data.frame(
  R2 = caret::R2(ridge.pred, yTest.Full),
  RMSE = RMSE(ridge.pred, yTest.Full),
  MSE = (RMSE(ridge.pred, yTest.Full)^2),
  MAE = MAE(ridge.pred, yTest.Full)
)
colnames(ridge.results)<-c("R2","RMSE", "MSE", "MAE")
## Coefficent Estimates
out.ridge=glmnet(xTrain.Full,yTrain.Full,alpha = 0)
ridge.coef=predict(out.ridge, type="coefficients", s=bestLam.ridge)[1:51,]
```

Lasso Regression

```r
set.seed(123)
## Find best Lambda using cross validation
cv.out.lasso<-cv.glmnet(xTrain.Full, yTrain.Full, alpha=1)
plot(cv.out.lasso)

bestLam.lasso=cv.out.lasso$lambda.min
bestLam.lasso
lasso.mod=glmnet(xTrain.Full, yTrain.Full, alpha=1, lambda=grid, thresh=1e-12
)
lasso.pred=predict(lasso.mod, s=bestLam.lasso, newx=xTest.Full)
lasso.MSE=mean((lasso.pred-yTest.Full)^2)
lasso.results=data.frame(
  R2 = caret::R2(lasso.pred, yTest.Full),
  RMSE = RMSE(lasso.pred, yTest.Full),
  MSE = (RMSE(lasso.pred, yTest.Full)^2),
  MAE = MAE(lasso.pred, yTest.Full)
)
colnames(lasso.results)<-c("R2","RMSE", "MSE", "MAE")
out.lasso=glmnet(xTrain.Full, yTrain.Full, alpha=1, lambda=grid)
lasso.coef<-predict(out.lasso, type="coefficients", s=bestLam.lasso)[1:51,]
## Coefficients shrunk to zero:
shrunkCoefs<-lasso.coef[lasso.coef==0]
shrunkCoefs
length(shrunkCoefs)
```

Partial Least Squares

```
set.seed(123)
library(pls)

##
## Attaching package: 'pls'

## The following object is masked from 'package:caret':
##
##     R2

## The following object is masked from 'package:corrplot':
##
##     corrplot

## The following object is masked from 'package:stats':
##
##     loadings

pls.mod=plsr(povPropCol~., data=Train.Std, scale=TRUE, validation="CV")
summary(pls.mod)
validationplot(pls.mod, val.type = "MSEP", legendpos = "topright", main="Mean Square Error Prediction vs. Number of Components")

pls.cv = RMSEP(pls.mod)
pls.best.dims = which.min(pls.cv$val[estimate = "adjCV", , ]) - 1
pls.best.dims
abline(v=pls.best.dims)

pls.pred=predict(pls.mod, xTest.Full, ncomp = pls.best.dims)

pls.results=data.frame(
  R2 = caret::R2(pls.pred, yTest.Full),
  RMSE = RMSE(pls.pred, yTest.Full),
  MSE = (RMSE(pls.pred, yTest.Full)^2),
  MAE = MAE(pls.pred, yTest.Full)

)
colnames(pls.results)<-c("R2","RMSE", "MSE", "MAE")
```

Combine and Compare Results

```
predResults<-rbind(lmod.results, ridge.results, lasso.results, pls.results)
rownames(predResults)<-c("OLS", "Ridge", "Lasso", "PLS")
predResults
```