STA 567: Final Project

Erik M. Painter

2024-12-05

Introduction

In today's current political landscape, Pennsylvania is seen as one of the most important states in determining who will win the U.S presidential election. Pennsylvania is known for its status as a battleground, or 'swing' state, as it historically can be won by either the Democratic or Republican candidate. Since 1900, Pennsylvania has voted Democrat ~45% of the time and Republican ~53% of the time. Pennsylvania's importance for presidential candidates is evident from the fact that the candidate who has won the Commonwealth has gone on to win the presidential election 78% of time. Pennsylvania has shifted towards Democratic candidates in recent years, with Democrats winning the state 71% of the time since 2000. Donald Trump is the only Republican presidential candidate to win Pennsylvania since 2000, with him carrying the state in both 2016 and 2024.

In the three presidential elections before 1992, Pennsylvania had voted for the Republican candidate which included Ronald Reagan winning the state in both 1980 and 1984, and George H.W. Bush winning in 1988. In 1992, Bill Clinton faced off against the Republican incumbent George H.W. Bush (and Independent Ross Perot), where he won the states 23 electoral votes by a 9 point margin (45% vs. 36%).

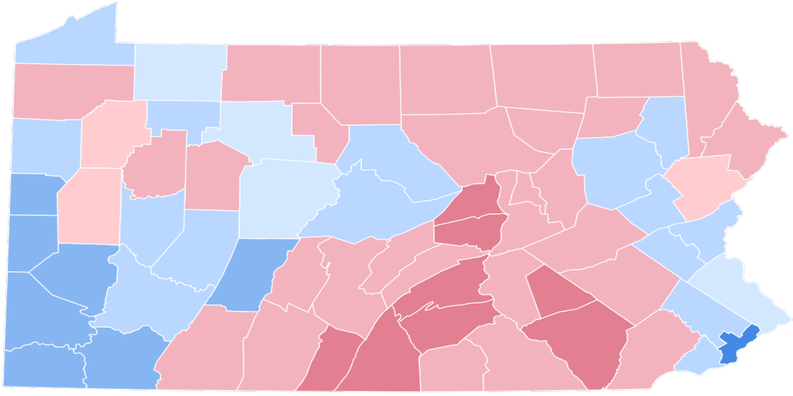


Figure 1: Pennsylvania 1992 - Presidential Election Results by County

The above figure shows the results of the 1992 presidential election broken down by Pennsylvania's 67 counties. Bill Clinton won big in urban counties, which is historically on par with Democrat's election performances, but he was also able to make up ground in some rural counties which solidified his electoral win of Pennsylvania.

The focus of this analysis is to develop a valid and viable multiple linear regression model to predict the percentage of votes Bill Clinton (Democrat) received in Pennsylvania's 67 counties during the 1992 presidential election. Using county-level demographic variables as median age, per capita income, poverty rate, and other relevant variables, this study aims to identify which factors were most influential in determining Clinton's vote share across Pennsylvania's counties.

Table 1: Sample of Pennsylvania Dataset

pvote	ma	ms	pci	pp	pv	pf	pd	pnh	ci
32.46	33.5	106575	17959	7.4	9.95	51.11	156.3	11.88	170
52.75	36.7	165128	23541	12.8	16.08	53.02	1826.5	5.09	408
45.87	36.6	99587	15964	14.0	15.03	51.82	113.1	5.15	113
54.50	37.0	75921	16872	13.7	16.33	52.27	432.1	6.16	195
31.04	35.7	87327	14071	14.5	11.25	51.08	47.8	4.16	150
35.03	35.4	139730	20786	11.2	12.80	51.65	399.2	7.01	353

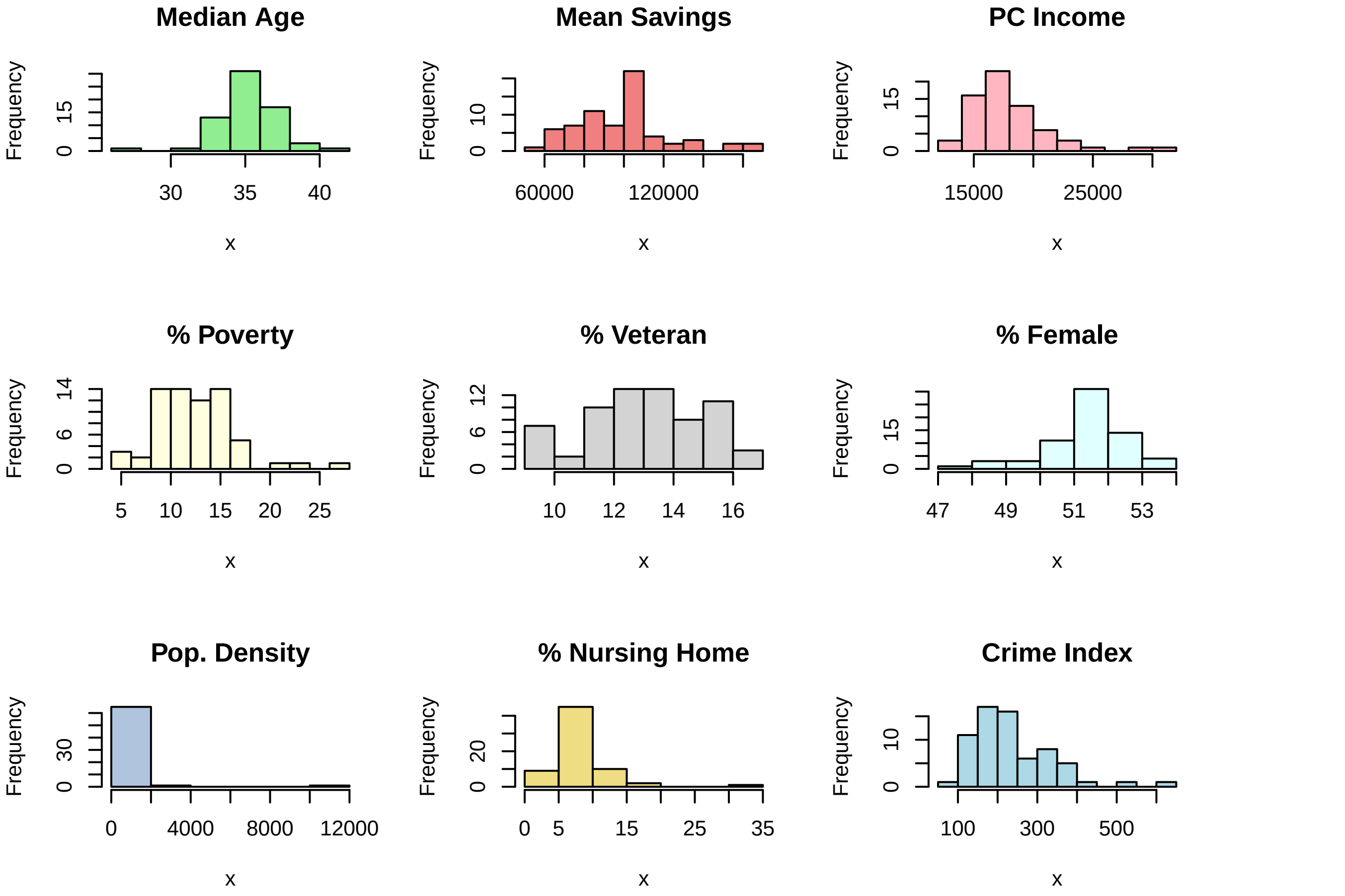
Understanding these relationships can provide insights into the factors that influenced the voting patterns in 1992 presidential election, where Clinton successfully flipped Pennsylvania after several cycles of Republican victories.

Exploratory Data Analysis

Data Introduction

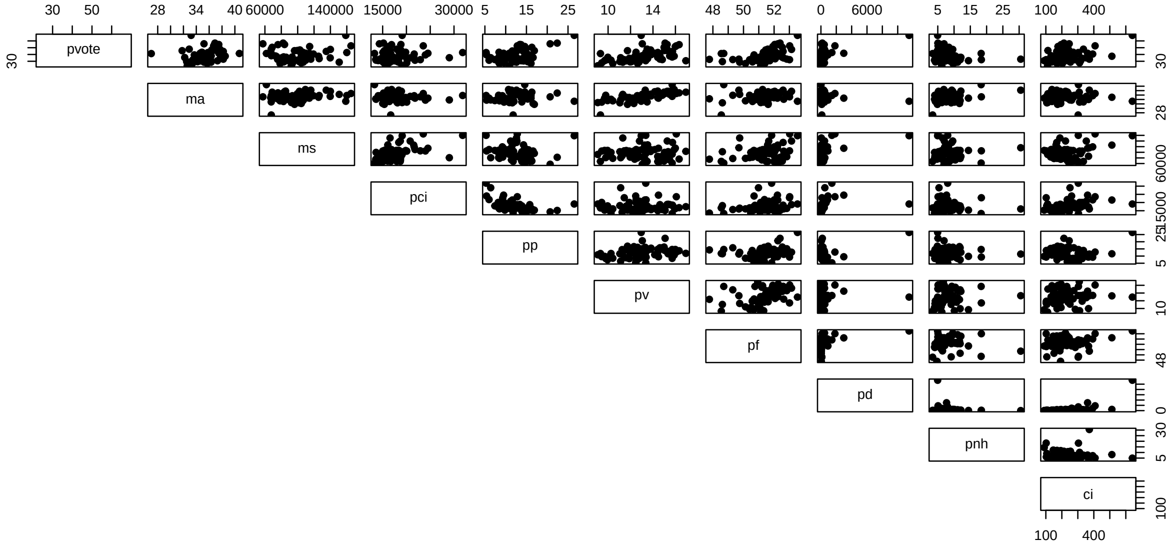
The data used in this analysis came from the U.S. Census Bureau. Each observation includes the percentage of the vote that Clinton received in each of Pennsylvania's 67 counties, along with 9 demographic variables. The variables include: Median Age, Mean Savings, Per Capita Income, Percent Poverty, Percent Veteran, Percent Female, Population Density, Percent in Nursing Homes, and Crime Index.

Exploratory data analysis is useful in exploring our data to gain a deeper understanding of the relationships among the variables. By examining each variable on its own we can obtain an idea about the variables structure such as its distribution, existence of outliers, and possible transformations to perform on these variables.



It is clear that some of the distributions are skewed and/or the range of variables are very large. It is also evident that outliers exist. An important step before finding a model is to clean the data such that the data conforms to the assumptions needed for constructing a valid multiple regression model.

In addition, we also want to understand the relationships between variables. A pairwise plot is useful for identifying the relationship between each predictor and our response variable, as well as identifying possible cases of multi-collinearity between predictors (will address later).



As we can see there are outliers (as stated above) and the strengths of linearity between the response and the predictors vary. I believed that this was mostly due to the existence of outliers. Therefore, I removed 4 observations that I subjectively determined to be outliers. After removing these observations, the correlation between the predictors seemed to improve with each predictor and the response. I only removed 4 observations as my data set is not that large and I did not want to remove to many observations in the initial cleaning stage.

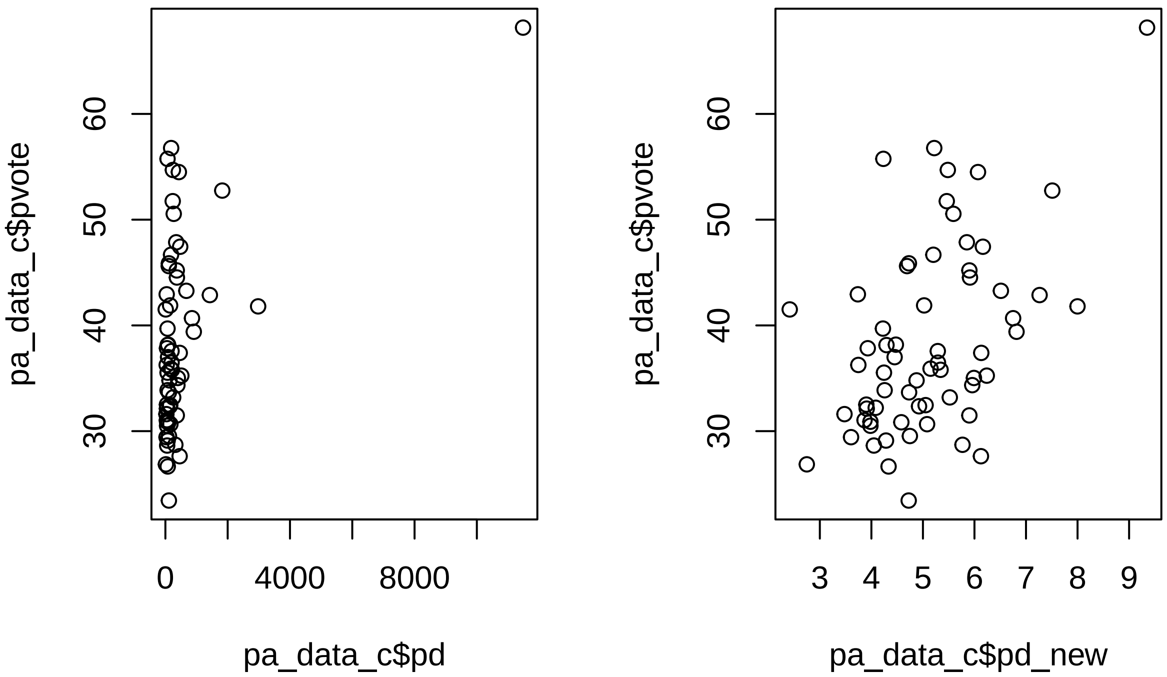
I did not remove some of the most prominent outliers, as I wanted to utilize Box-Cox transformations to see if I could retain these observations before dropping them. **Not all outliers are bad**. For example, the most distinguished outlier was Philadelphia County, which is the most populous county in Pennsylvania. I ran the subsequent analysis by dropping Philadelphia County and obtained a model with higher predictive power, however I felt that by losing this county we were losing very important information in predicting the percentage of the vote for Bill Clinton. Therefore, I applied Box-Cox transformations in order to keep observations such a Philadelphia County.

Initial Cleaning & Transformations

Based on the prior analysis, I decided to drop the following observations: 14,12,31,57. The counties associated with these observations were: Centre, Cameron, Huntingdon, and Sullivan. These counties are mostly rural and have very low populations as they are home to state parks, except for Centre County which is home to Penn State University. I felt that dropping these counties was appropriate as these counties do not reflect the overall state's voting patterns in a meaningful way. After dropping these observations, the data is cleaned up to an extent and where the relationships between the predictors and response are more clear. I then went forward with applying Box-Cox transformations to each predictor to see if transforming them (according to the optimal λ) would improve their correlation with *% vote*.

After applying the transformations, many correlations did not improve. However, one predictor did benefit from the transformation, *pop. density*. I chose to include this transformed variable in my model due to the fact that after applying the transformation, this variable had a higher correlation with percentage of the vote compared to the non-transformed (original) variable.

To provide an example, I found the optimal λ for the *pop. density* transformation to be 0. I then applied a *log* transformation and below is the result of the transformation. It is important to point out that after the initial cleaning of my data (where I removed 4 observations) the matrix plots still showed some evidence of outliers/influential points, especially with *pop. density*. However, after the transformation, this observation was no longer an outlier. My data set is not very large and so I did not want to remove too many data points. To reiterate, some of these observations may actually provide some valuable information.



Lastly, to get an idea of the correlations, below is a correlation matrix. We can use this matrix to get an idea of the strength of the linear relationships between our predictors and response, as well as the expected sign estimated coefficients in our analysis. If we get a sign in our analysis that does not match coincide with our expectations, this may be an instance of multicollinearity issues. This matrix also provides an idea of the extent of multicollinearity between our predictors.

```
##              pvote          ma          ms         pci          pp         pv
## pvote   1.00000000  0.34082174  0.11989884  0.07053706  0.50589847  0.6105327
## ma      0.34082174  1.00000000  0.03576841 -0.12619269  0.12128800  0.7098378
## ms      0.11989884  0.03576841  1.00000000  0.45037796 -0.13788877  0.0112238
## pci     0.07053706 -0.12619269  0.45037796  1.00000000 -0.55544449 -0.0114976
## pp      0.50589847  0.12128800 -0.13788877 -0.55544449  1.00000000  0.3117376
## pv      0.61053266  0.70983783  0.01122380 -0.01149760  0.31173762  1.0000000
## pf      0.54444099  0.16288678  0.28444692  0.19542315  0.29527485  0.3801675
## pd_new  0.49445106 -0.07275610  0.54033725  0.68655040 -0.13581151  0.1673025
## pnh    -0.22271221  0.19732074 -0.10228412 -0.06739343 -0.06831902  0.0656144
## ci      0.41097035 -0.07049616  0.31852304  0.39427181  0.09574020  0.1070063
##                 pf     pd_new         pnh          ci
## pvote   0.54444099  0.4944511 -0.22271221  0.41097035
## ma      0.16288678 -0.0727561  0.19732074 -0.07049616
## ms      0.28444692  0.5403373 -0.10228412  0.31852304
## pci     0.19542315  0.6865504 -0.06739343  0.39427181
## pp      0.29527485 -0.1358115 -0.06831902  0.09574020
## pv      0.38016750  0.1673025  0.06561440  0.10700633
## pf      1.00000000  0.5363219 -0.05781115  0.28770670
## pd_new  0.53632194  1.0000000 -0.23136469  0.61112663
## pnh    -0.05781115 -0.2313647  1.00000000 -0.17603046
## ci      0.28770670  0.6111266 -0.17603046  1.00000000
```

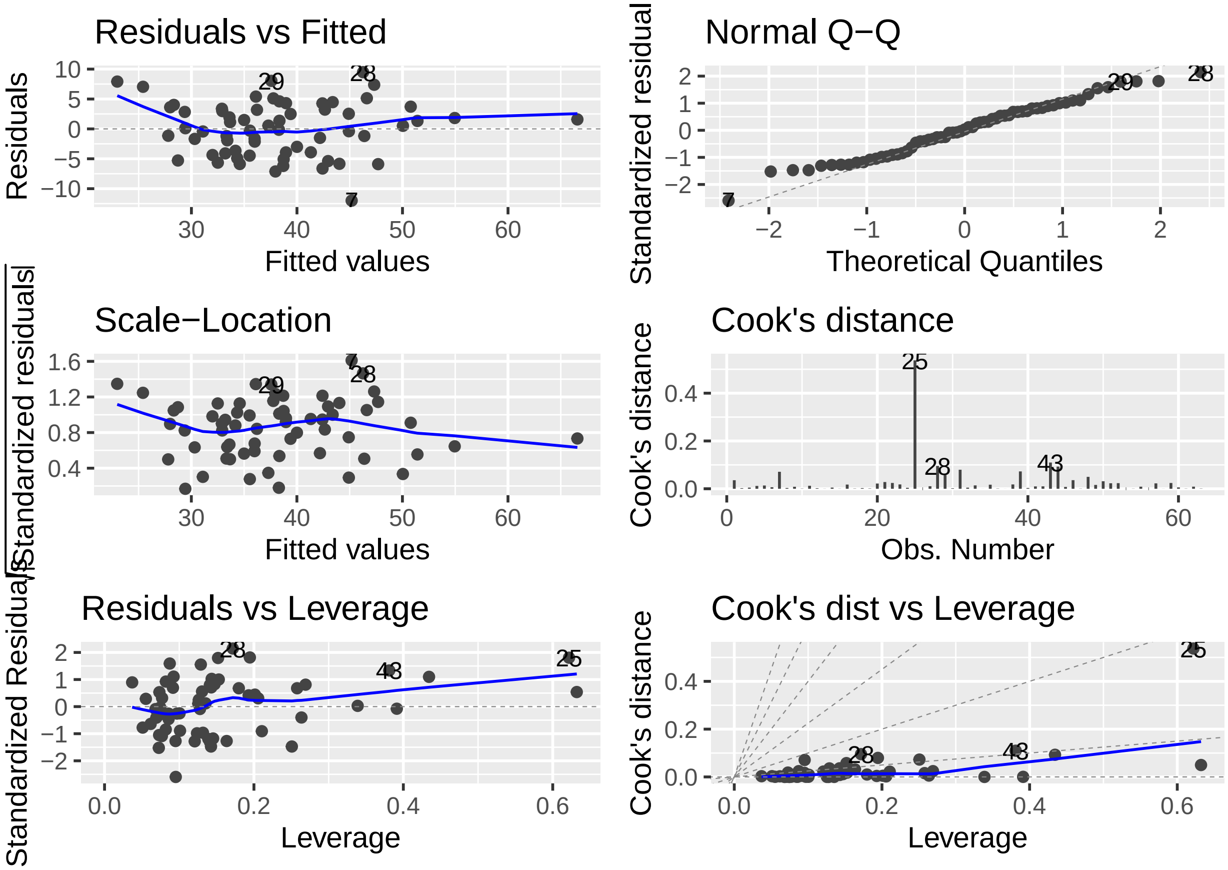
Fitting Initial Model

Now that I have cleaned my data and transformed some of my predictors to a **subjectively** satisfactory level, I will fit the the full linear model to gain a deeper understanding of my data and identify any additional cleaning or transformations that need to be done.

```
##
## Call:
## lm(formula = pvote ~ ., data = pa_data_c)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.9937  -3.9229   0.1126   3.2960   9.4837
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.742e+01  4.482e+01  -1.058 0.294836
## ma           9.213e-01  5.747e-01   1.603 0.114860
## ms          -5.801e-05  3.223e-05  -1.800 0.077594 .
## pci          6.685e-05  3.372e-04   0.198 0.843618
## pp           1.023e+00  2.495e-01   4.100 0.000143 ***
## pv           1.191e+00  5.556e-01   2.143 0.036748 *
## pf           2.425e-01  8.949e-01   0.271 0.787478
## pd_new       3.766e+00  1.023e+00   3.681 0.000545 ***
## pnh         -3.826e-01  2.215e-01  -1.728 0.089900 .
## ci           3.945e-03  8.339e-03   0.473 0.638096
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.858 on 53 degrees of freedom
## Multiple R-squared:  0.7439, Adjusted R-squared:  0.7005
## F-statistic: 17.11 on 9 and 53 DF,  p-value: 8.498e-13
```

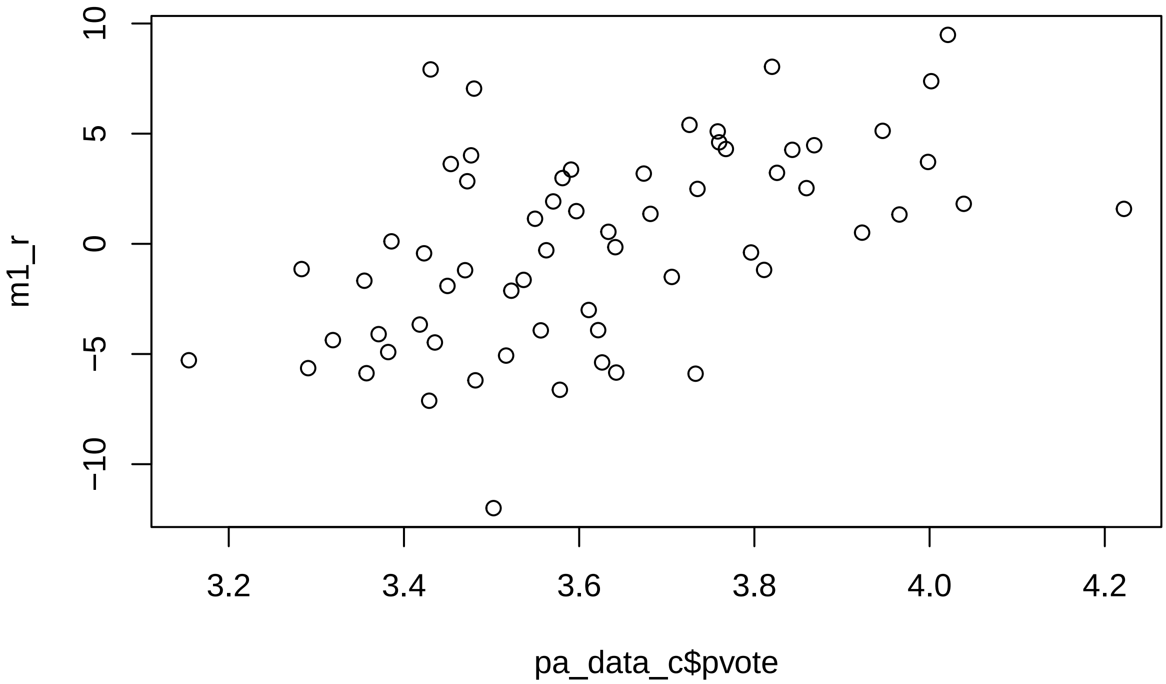
Based on these results, we can see the predictors that are significant, somewhat significant, and not significant. This is in line with our expectations from EDA and matrix plots.

For each step of the model development/selection process, it is important to check the assumptions using the residual and diagnostic plots. We can see the diagnostic plots for our initial full linear model.



The largest violation to any assumption is heteroscedasticity. This implies that the relationship between our response and our predictors is not inherently linear. Therefore, I applied a log transformation on *% vote*, and reran the analysis. After rerunning the analysis, the assumptions were much better met. I checked the residuals vs. each predictor and looked for any sort of non-random pattern for identifying if I should apply another transformation to any of my predictors. There were no instances of non-random pattern, so I decided to go forward with searching for the “best” model.

Before searching for the “best” model, I felt it was important to point out that the relationship between the residuals for our initial model and the *% vote* was linear. This suggests that we are missing a predictor in our data collection that is positively linearly related to our response. This predictor could be education level, race, economic factors, regional cultural factors, etc. We obviously do not have the ability to go back and collect this data, however it should be pointed out that there exists other predictors that have predictive power or a relationship with determining the *% vote*.



Model Seleciton

The selection of the appropriate, or "best" model, is crucial for ensuring reliable predictions and meaningful insights into the relationship between the percentage of the vote for Clinton and our predictors. The following approach to selecting a model follows a systematic and dynamic process that utilizes multiple selection criteria and validation techniques.

1. Selection Criteria

We will be using two primary selection criteria for model evaluation:

- R^2_{adj}: Finding models with the largest R^2_{adj}
- BIC: Finding models with the minimum BIC

2. Selection Methods

This analysis employs multiple selection techniques:

- All Possible Subsets: Using regsubsets function, it provides the best set of variables for each model size (using RSS)
- Forward Selection: Beginning with no predictors and sequentially adding significant variables to the equation

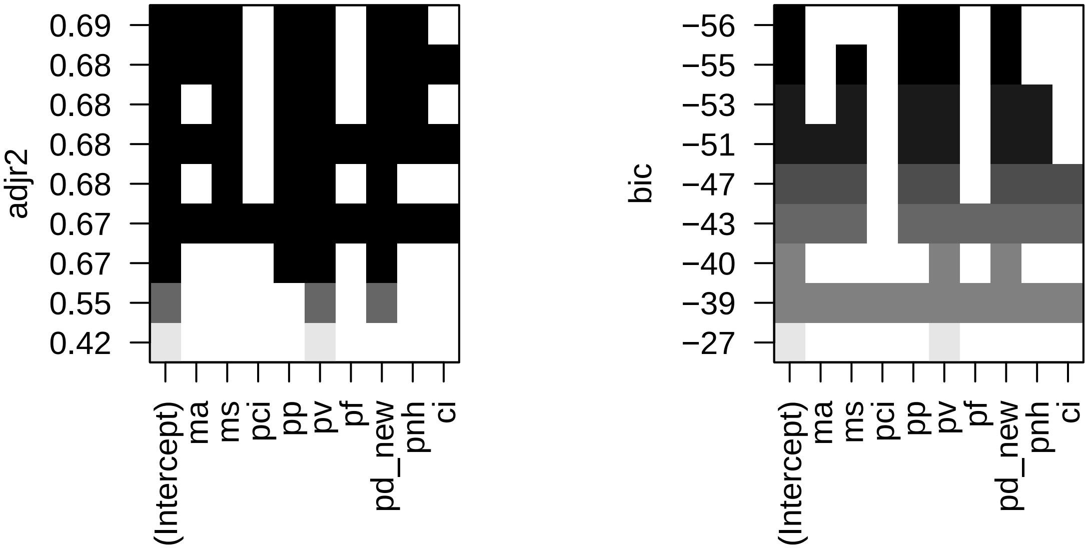
- Backward Selection: Beginning with the full fitted model and drops variables sequentially based on dropping the variables with the smallest contribution to the reduction to the error sums of squares (insignificant t-Tests)
- Step-wise Selection: Combining both forward and backward approaches to optimize variable selection

This process consisted of 4 phases:

- Searching all possible subsets for "best" model using R_a^2 and BIC
- Using Forward, Backward, and Step-wise selection using BIC criteria to find the "best" model
- K-fold Cross Validation
- Selecting "best" model and checking residual and diagnostic plots

I completed these 4 phases in 2 iterations (searched for best models, compared them, and then checked assumptions) to find a model that was the "best", where all of the assumptions were met. I will the highlight the process for the first iteration only, as I followed the same steps for the second iteration. Each iteration included a data set that was a bit more cleaned up. All of the code however can be found in the appendix (pg. 18).

Below are the results obtained from running all possible subsets:



Next, these are the results obtained from using step-wise selection:

```
##                Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept) 2.24980613 0.125326121 17.951614 1.393093e-25
## pp          0.02099977 0.004423472  4.747349 1.355203e-05
## pv          0.05356030 0.009161998  5.845920 2.326780e-07
## pd_new      0.08051716 0.013481963  5.972214 1.438061e-07
```

Now we have two models to compare. As stated, I will be using K-fold cross validation and residual analysis to find the "best" model.

$$Y_1 = pp + pv + log(pd) \text{ (BIC)}$$
$$Y_2 = ma + ms + pp + pv + pf + log(pd) + pnh \ (R_a^2)$$

3.) K-Fold Cross Validation After obtaining models of interest it can be difficult to determine the "best" model. We can use cross validation procedure to asses the predictability of these models.

- Divide entire data set in k folds ($k = 5$)
- Use remaining k - 1 folds to estimate model parameters
- Assess model predictability by calculating Mean Square Prediction Error (MSPE) for the testing data
- Repeat this for each fold, and we find the average MSPE across all folds
- The "best" model is the one with the smallest average MSPE

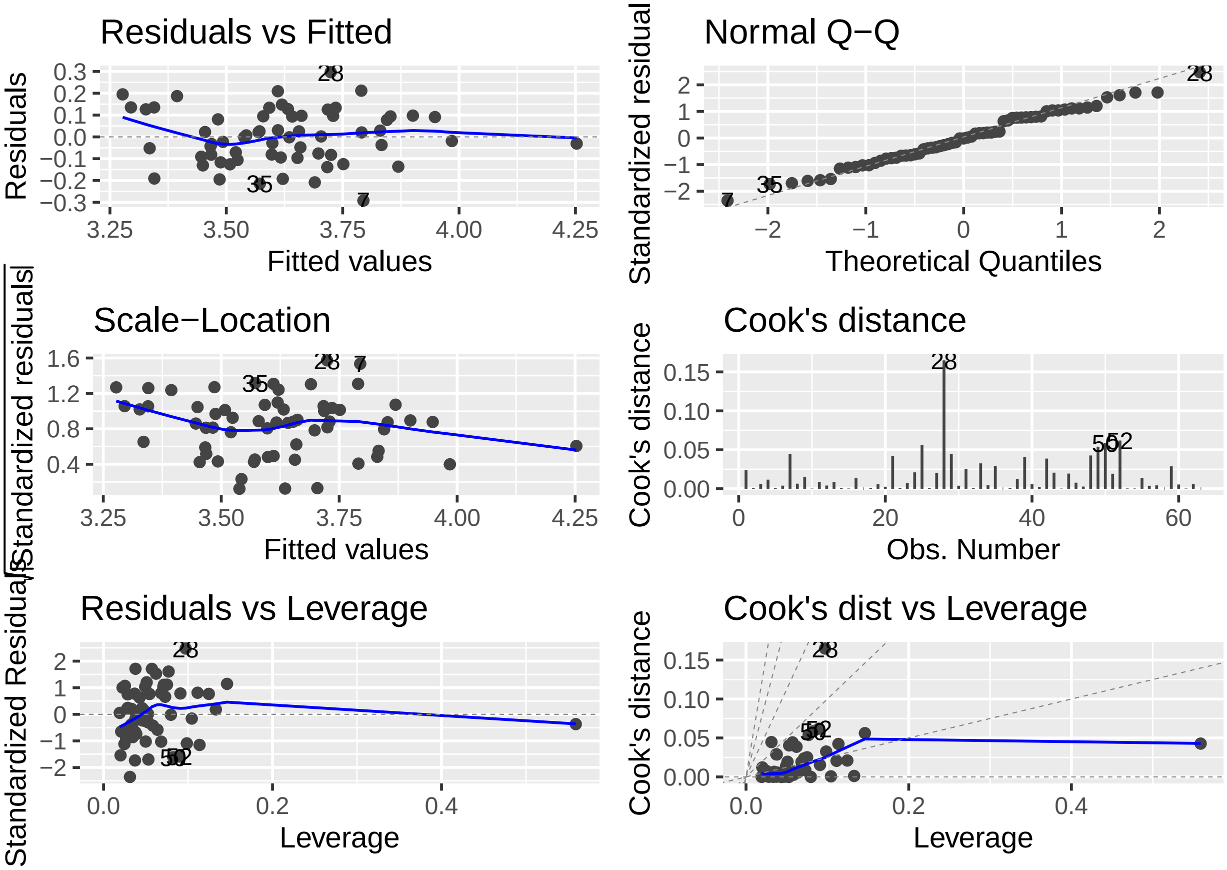
```
fm1cv <- CVlm(pa_data_c, form.lm = formula(fm1), m = 5, seed = 7, plotit = F) # 0.017 (MSPE)
fm2cv <- CVlm(pa_data_c, form.lm = formula(fm2), m = 5, seed = 7, plotit = F) # 0.019 (MSPE)
```

I proceeded to perform K-Fold Cross Validation on the two models above and both had very similar Mean Square Prediction Errors. Then, I fitted both models and checked their respective summaries and residual plots and found model Y_1 (BIC criteria) to be the "best" model.

```
##
## Call:
## lm(formula = pvote ~ pp + pv + pd_new, data = pa_data_c)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.292246 -0.086993 -0.001865  0.095344  0.296613
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.249806   0.125326  17.952  < 2e-16 ***
## pp          0.021000   0.004423   4.747 1.36e-05 ***
## pv          0.053560   0.009162   5.846 2.33e-07 ***
## pd_new      0.080517   0.013482   5.972 1.44e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1261 on 59 degrees of freedom
## Multiple R-squared:  0.6856, Adjusted R-squared:  0.6696
## F-statistic: 42.88 on 3 and 59 DF,  p-value: 7.792e-15
```

Model Y_1 as all of the predictors were statistically significant at the 5% level. The R_a^2 was not that of models were very similar, but based on the residual analysis and all predictors being significant, I chose this to be my "best" model.

I chose this model as all of the predictors in the model were significant unlike in model Y_2. Also, the residual plots showed that the assumptions were more closely met. However, we can see that observation 28 is a clear influential point, so I removed it for the next iteration.



I dropped these observations and went through the second iteration (following the same process) of trying to find the "best" model.

Variable/Model Selection Process

I will not be showing my results and code for each iteration, however, I below is a table outlining the results that I obtained after each iteration of my variable/model selection process. It shows the predictors in each of the "best" models, R_a^2, BIC, residual analysis, and actions taken before beginning the next iteration.

Table 2: Model/Variable Selection Process

Response	Predictors	Adj. R^2	BIC	Residuals	Action
Y	All	0.70	NA	Heteroscedasticity	log(Y)
log(Y)	pp,pv,pd_new	0.67	-56	Outliers	Dropped: 28
log(Y)	pp,pv,pd_new,pnh	0.69	-59	NA	NA

Final Model

After 2 iterations of searching for the best model, I obtained a model where all predictors were significant and the assumptions for the model were met to a satisfactory level.

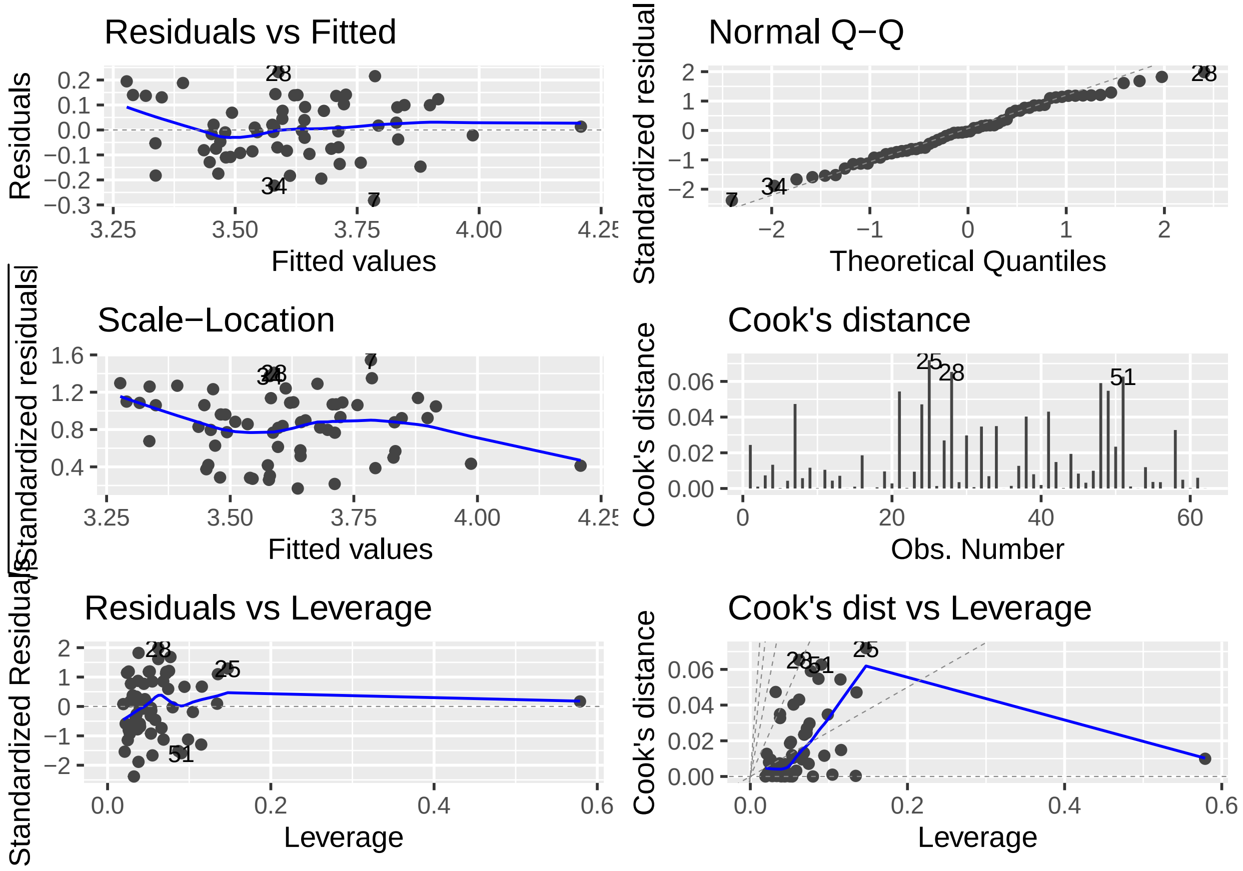
```
##
## Call:
## lm(formula = pvote ~ pp + pv + pd_new, data = pa_data_c2)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28232 -0.08283 -0.00439  0.09724  0.23142
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.252321   0.119666  18.822  < 2e-16 ***
## pp          0.017883   0.004391   4.072 0.000143 ***
## pv          0.055476   0.008779   6.319 4.04e-08 ***
## pd_new      0.081715   0.012881   6.344 3.67e-08 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.1204 on 58 degrees of freedom
## Multiple R-squared:  0.7021, Adjusted R-squared:  0.6867
## F-statistic: 45.57 on 3 and 58 DF,  p-value: 2.899e-15
```

The interpretation of each coefficient is as follows:

- $\hat{\beta}_0$: Expected value of *log(pvote)* when all other predictors are zero. This does not have any practical or intuitive interpretations.
- $\hat{\beta}_1$: A one unit increase in the poverty rate is associated with an expected 1.7% increase in the % of the vote received by Bill Clinton
- $\hat{\beta}_2$: A one unit increase in the percentage of veterans is associated with an expected 5.5% increase in the percentage of the vote received by Bill Clinton
- $\hat{\beta}_3$: Each percentage increase in the population density for a county is associated with an expected 0.8% increase in the percentage of the vote received by Bill Clinton

To validate our findings, we performed a thorough residual analysis to check the assumptions of our model. The residual plots indicate that our model assumptions are satisfied, suggesting that the model is well-fitted. As a result, we have derived a simple and interpretable model that effectively highlights the relationships between the percentage of votes received by Bill Clinton and various demographic variables. This confirmation strengthens the reliability and validatity of our final model.



Multicollinearity

Multicollinearity occurs when predictor variables in a regression model are highly correlated, creating challenges in understanding individual variable effects. In exploratory data analysis, multicollinearity becomes evident through high correlation matrix values and strongly linear relationships between predictors in the matrix plots, which can lead to unstable and unreliable coefficient estimates.

To diagnose multicollinearity, I used techniques such as Variance Inflation Factor (VIF) and condition index analysis. VIF quantifies how much the variance of a coefficient is inflated due to correlations, with values exceeding 10 signaling severe (potential) multicollinearity. Condition index analysis explores eigenvalues and variance proportions, identifying specific sets of collinear variables through detailed matrix examinations, with condition indices above 30 indicating significant interdependence among predictors. Loading's exist for each predictor at each of the condition indices, and predictors which have non-negligible loading's are usually the predictors that are collinear with one another. These loading's show the predictors that are within each collinear set.

Multicollinearity can mask the true relationship between individual predictors and the response variable, making interpretation of regression models challenging. By systematically identifying and addressing these intricate statistical relationships, we can develop more robust and interpretable models that accurately represent the underlying data and relationship.

Below is a table of the multicollinearity diagnostics for the final model:

```
## Tolerance and Variance Inflation Factor
## ---------------------------------------
##   Variables Tolerance      VIF
## 1        pp 0.8648074 1.156327
## 2        pv 0.8519691 1.173752
## 3    pd_new 0.9392835 1.064641
##
##
## Eigenvalue and Condition Index
## ------------------------------
##    Eigenvalue Condition Index   intercept          pp           pv      pd_new
## 1 3.885533272         1.00000 0.001069751 0.004580565 0.0011376536 0.003246653
## 2 0.078773811         7.02319 0.002758276 0.557187179 0.0002623268 0.248427444
## 3 0.025712742        12.29281 0.128597669 0.434533623 0.1891374479 0.723685397
## 4 0.009980176        19.73132 0.867574304 0.003698633 0.8094625718 0.024640506
```

As we can see no VIF > 10, so there are no instances of severe multicollinearity between any of the predictors in my final model.

Looking at the condition indices, we have two that are greater than 15. Although 30 is a threshold for severe multicollinearity and corrective measures, a condition index > 15 indicates that there is potential multicollinearity and we should investigate. Based on the results above, it appears that there are no extreme instances of multicollinearity as the Condition Index of 19.73 does not have two non-negligible loading's that highlight a collinear set of predictors.

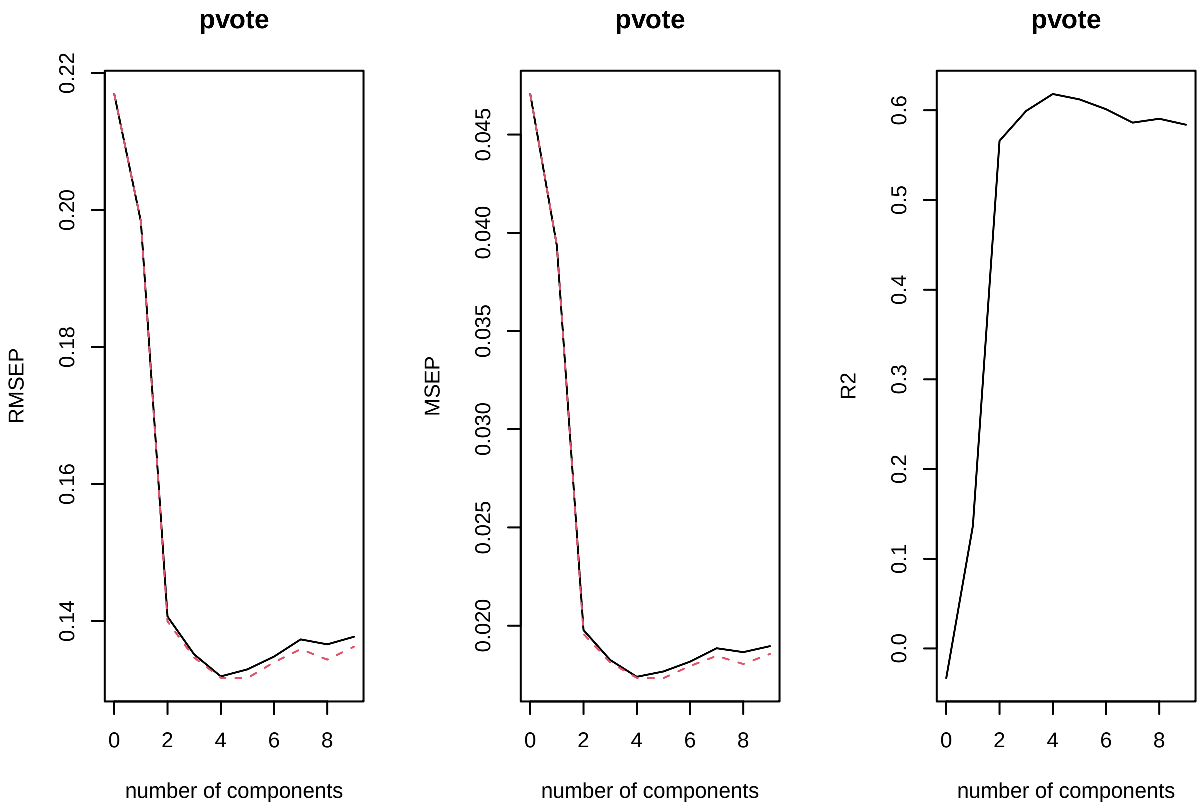
I did not undertake any corrective actions due to the findings from my multicollinearity diagnostics.

Principal Component Regression

Principal Component Regression (PCR) offers a powerful statistical technique for validating regression models by transforming potentially correlated predictors into a set of linearly independent (uncorrelated) principal components. By creating orthogonal predictors that capture the maximum variance in the original data, PCA provides a robust method to compare and validate traditional regression models. This approach allows us to assess model stability, understand complex predictor relationships, and potentially improve predictive power by reducing multicollinearity. I will be comparing my final model with PCR and identyfying if we can get better results.

```
## Data:    X dimension: 62 9
##  Y dimension: 62 1
## Fit method: svdpc
## Number of components considered: 9
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV          0.2169   0.1983   0.1406   0.1351   0.1319   0.1329   0.1348
## adjCV       0.2169   0.1984   0.1399   0.1346   0.1317   0.1316   0.1340
##        7 comps  8 comps  9 comps
## CV      0.1373   0.1366   0.1377
## adjCV   0.1359   0.1343   0.1362
##
## TRAINING: % variance explained
##        1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X        32.18    55.54    69.81    78.92    86.52    93.31    96.44    98.28
## pvote    24.00    61.89    64.46    66.20    68.50    68.83    70.74    72.73
##        9 comps
## X       100.00
## pvote    72.74
```

From the results above, we can see the percentage of the variation explained by each additional principal component. We can see that most of the variability explained is mostly in the first 5-6 components. We now want to figure out how many principal components to keep in our model. We can use validation plots to find the optimal number of PC's to keep. We look for the number of PC's to have the lowest RMSEP, MSEP, and maximum R^2.

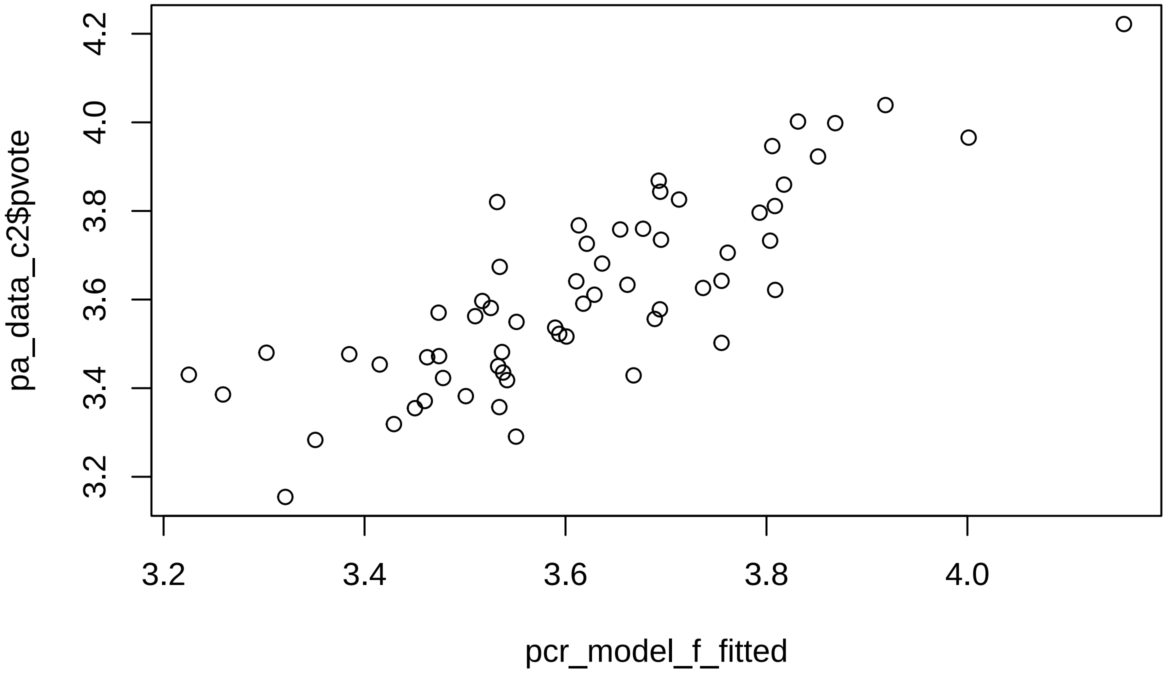


We utilize the above validation plots to find how many principal components to keep. We looks for the number of components that result in the lowest RMSEP, MSEP, and the largest R^2. Based on the results, above it seems appropriate to keep 5 principal components.

Here are the coefficients for the PCR model transformed back into the original units:

```
## , , 5 comps
##
##                    pvote
## (Intercept) -0.074790269
## ma           0.042806679
## ms          -0.011218955
## pci          0.002236822
## pp           0.043427143
## pv           0.070652282
## pf           0.038170911
## pd_new       0.035597913
## pnh         -0.035003036
## ci           0.053489428
```

Below shows the results from the PCA regression model:

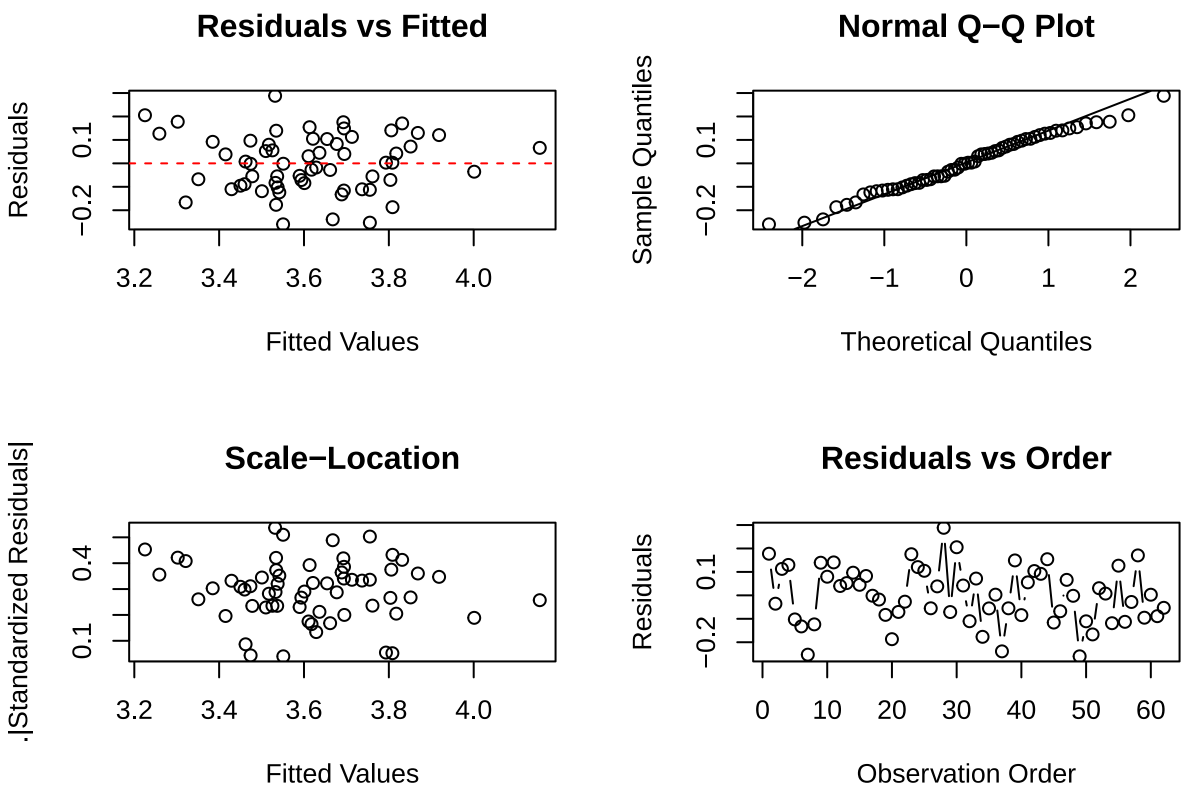


```
## [1] 0.6850373
```

The PCA regression analysis yielded promising results. When comparing the fitted values to the actual values, the relationship appears to follow a linear trend, indicating a strong predictive power of the model. This is further supported by an R^2 value of 0.69, which means that 69% of the variance in the actual values is explained by the model. Additionally, the principal component analysis revealed that 5 components were sufficient to explain 86% of the variance in the data. This suggests that a substantial amount of the original data's information is captured by these components, allowing for a more efficient and insightful analysis. The combination of a high R^2 and a significant proportion of explained variance demonstrates the effectiveness of using PCA in this regression model.

Importantly, this model did **not** produce results significantly better than my final fitted model. This shows that my final fitted model is already capturing the relationships in the data effectively. In addition, my final fitted model remains simpler and more interpretable than the PCA regression model without sacrificing much (if any) predictive power.

Lastly, to validate our results for the PCA regression model, we need to check the assumptions. They appear to be met.



Conclusion

In this analysis, we sought to understand the determinants of the percentage of votes received by Bill Clinton in a given election, focusing on various demographic variables. Through cleaning, transforming, variable/model selection, cross validation, and residual analysis we identified key factors influencing voting patterns in Pennslyvania for the 1992 presidential election and obtained a final model with an $R_a^2 = 67\%$, indicating that our model explains a substantial portion of the variance in the data.

The final model's coefficients provided insightful interpretations, highlighting the significant impact of specific demographic factors on the vote percentage. This analysis underscores the importance of these variables in understanding electoral outcomes and offers a robust framework for predicting future voting behavior.

Overall, our findings contribute to a deeper understanding of the electoral dynamics at play and demonstrate the effectiveness of regression analysis in deriving meaningful and actionable insights from complex datasets. This model not only aids in interpreting past elections but also serves as a valuable tool for strategizing future campaigns.

Appendix

This is all of the code for this project:

```r
setwd("/Users/epainter/Desktop/STA 567/Final Project")
```

```r
# Libraries
library(dplyr)
library(magrittr)
library(ggfortify)
library(corrplot)
library(car)
library(tidyverse)
library(MASS) ## may need this for transformations later

# 1.) Loading Data
data <- read.csv("clinton92.csv", header=TRUE)

# 2.) Setting up data
pa_data <- data %>%
  filter(grepl(", PA", Name)) #%>%
  #dplyr::select(-Name)

colnames(pa_data) <- c("name", "pvote", "ma", "ms",
                       "pci", "pp", "pv",
                       "pf","pd", "pnh", "ci")

rownames(pa_data) <- NULL

# 3.) EDA
# Summary statistics
summary(pa_data)

par(mfrow=c(3,3))
# Create histograms for each variable
hist(pa_data$pvote, main = "% Vote", xlab = "y", col = 'black')
hist(pa_data$ma, main = "Median Age", xlab = "x", col = "lightgreen")
hist(pa_data$ms, main = "Mean Savings", xlab = "x", col = "lightcoral")
hist(pa_data$pci, main = "PC Income", xlab = "x", col = "lightpink")
hist(pa_data$pp, main = "% Poverty", xlab = "x", col = "lightyellow")
hist(pa_data$pv, main = "% Veteran", xlab = "x", col = "lightgray")
hist(pa_data$pf, main = "% Female", xlab = "x", col = "lightcyan")
hist(pa_data$pd, main = "Pop. Density", xlab = "x", col = "lightsteelblue")
hist(pa_data$pnh, main = "% Nursing Home", xlab = "x", col = "lightgoldenrod")
hist(pa_data$ci, main = "Crime Index", xlab = "x", col = "lightblue")

# Matrix plot
pairs(pa_data[,!(names(pa_data) %in% 'name')], pch=19, bg="dimgray", cex.labels=1, lower.panel = NULL)

# Correlation Matrix
cor(pa_data[,!(names(pa_data) %in% 'name')])

par(mfrow=c(1,1))
# Identifying outliers
plot(pa_data$ci, pa_data$pvote)
identify(pa_data$ci, pa_data$pvote)

##################################### INITIAL CLEANING ####################################
pa_data_c <- pa_data[-c(14,12,31,57),] # Drop 4 observations
```

```r
rownames(pa_data_c) <- 1:nrow(pa_data_c)
pairs(pa_data_c[,!(names(pa_data_c) %in% 'name')], pch=19, bg="dimgray", cex.labels=1, lower.panel = NU

cor(pa_data_c[,!(names(pa_data_c) %in% 'name')])

# TRANSFORMATIONS
# Want to use boxcox transformations to find transformations that increase correlation with response
# If lambda = 0, then use log transformation
library(MASS)

# Median Age
lm1 = lm(ma ~ pvote ,data = pa_data_c)
mtr = boxCox(lm1, lambda = seq(-2, 2, 1/10), plotit = TRUE)

lam = mtr$x[which(mtr$y==max(mtr$y))]
lam

pa_data_c$ma_new <- (pa_data_c$ma^lam-1)/lam

plot(pa_data_c$ma, pa_data_c$pvote)
plot(pa_data_c$ma_new, pa_data_c$pvote)

cor(pa_data_c$ma, pa_data_c$pvote)
cor(pa_data_c$ma_new, pa_data_c$pvote) # no change

# Mean Savings
lm2 = lm(ms ~ pvote ,data = pa_data_c)
mtr2 = boxCox(lm2, lambda = seq(-2, 2, 1/10), plotit = TRUE)

lam2 = mtr2$x[which(mtr2$y==max(mtr2$y))]
lam2

pa_data_c$ms_new <- (pa_data_c$ms^lam2-1)/lam2

par(mfrow=c(1,2))
plot(pa_data_c$ms, pa_data_c$pvote)
plot(pa_data_c$ms_new, pa_data_c$pvote)

cor(pa_data_c$ms, pa_data_c$pvote)
cor(pa_data_c$ms_new, pa_data_c$pvote) # 8% worse

# Pop. Density
lm3 = lm(pd ~ pvote ,data = pa_data_c)
mtr3 = boxCox(lm3, lambda = seq(-2, 2, 1/10), plotit = TRUE)

lam3 = mtr3$x[which(mtr3$y==max(mtr3$y))]
lam3

pa_data_c$pd_new <- log(pa_data_c$pd)

par(mfrow=c(1,2))
plot(pa_data_c$pd, pa_data_c$pvote)
plot(pa_data_c$pd_new, pa_data_c$pvote)
```

```r
cor(pa_data_c$pd, pa_data_c$pvote)
cor(pa_data_c$pd_new, pa_data_c$pvote) # 1% better (keep)

# PC Income
lm4 = lm(pci ~ pvote ,data = pa_data_c)
mtr4 = boxCox(lm4, lambda = seq(-2, 2, 1/10), plotit = TRUE)

lam4 = mtr4$x[which(mtr4$y==max(mtr4$y))]
lam4

pa_data_c$pci_new <-  (pa_data_c$pci^lam4-1)/lam4

plot(pa_data_c$pci, pa_data_c$pvote)
plot(pa_data_c$pci_new, pa_data_c$pvote)

cor(pa_data_c$pci, pa_data_c$pvote)
cor(pa_data_c$pci_new, pa_data_c$pvote) # no change

# Female
lm5 = lm(pf ~ pvote ,data = pa_data_c)
mtr5 = boxCox(lm5, lambda = seq(-2, 2, 1/10), plotit = TRUE)

lam5 = mtr5$x[which(mtr5$y==max(mtr5$y))]
lam5

pa_data_c$pf_new <-  (pa_data_c$pf^lam5-1)/lam5

plot(pa_data_c$pf, pa_data_c$pvote)
plot(pa_data_c$pf_new, pa_data_c$pvote)

cor(pa_data_c$pf, pa_data_c$pvote)
cor(pa_data_c$pf_new, pa_data_c$pvote) # no change

# Nursing Home
lm6 = lm(pnh ~ pvote ,data = pa_data_c)
mtr6 = boxCox(lm6, lambda = seq(-2, 2, 1/10), plotit = TRUE)

lam6 = mtr6$x[which(mtr6$y==max(mtr6$y))]
lam6

pa_data_c$pnh_new <-  (pa_data_c$pnh^lam6-1)/lam6

plot(pa_data_c$pnh, pa_data_c$pvote)
plot(pa_data_c$pnh_new, pa_data_c$pvote)

cor(pa_data_c$pnh, pa_data_c$pvote)
cor(pa_data_c$pnh_new, pa_data_c$pvote) # no change

# Crime Index
lm7 = lm(ci ~ pvote ,data = pa_data_c)
mtr7 = boxCox(lm7, lambda = seq(-2, 2, 1/10), plotit = TRUE)

lam7 = mtr7$x[which(mtr7$y==max(mtr7$y))]
```

```r
lam7

pa_data_c$ci_new <-  log(pa_data_c$ci)

plot(pa_data_c$ci, pa_data_c$pvote)
plot(pa_data_c$ci_new, pa_data_c$pvote)

cor(pa_data_c$ci, pa_data_c$pvote)
cor(pa_data_c$ci_new, pa_data_c$pvote) # worse

# Dropping columns
pa_data_c <- pa_data_c %>%
  dplyr::select(c(pvote, ma, ms, pci, pp, pv, pf, pd_new, pnh, ci))

# Correlation matrix
cor(pa_data_c)

#################################### INITIAL MODEL ####################################
m1 = lm(pvote ~ ., data = pa_data_c)
summary(m1)
autoplot(m1, which = 1:6, ncol = 2, label.size = 3)

# Residuals vs each predictor
m1_r <- residuals(m1)
plot(pa_data_c$ms, m1_r) # Looks good for all predictors (no transformation needed at this stage)
plot(pa_data_c$pvote, m1_r)

# Making a log transformation to Y, there is a curved shape in residuals vs fitted
pa_data_c$pvote <- log(pa_data_c$pvote)

m2 = lm(pvote ~ ., data = pa_data_c)
summary(m2)
autoplot(m2, which = 1:6, ncol = 2, label.size = 3)

m2_r <- residuals(m2)
plot(pa_data_c$ci, m2_r) # Looks good for all predictors (no transformation needed at this stage)
plot(pa_data_c$pvote, m2_r)

# Based on this analysis, we are missing a postively related variable to % vote
# Think it could be education level, race, etc.
###### SEARCING FOR BEST MODEL (1) ######
### Will be using Adjusted R^2 and BIC for model selection criterion
library(leaps)

# All Possible Subsets
regfit.full <- regsubsets(pvote ~., data = pa_data_c, nvmax=9)
reg.summary <- summary(regfit.full)

names(reg.summary)

reg.summary$adjr2
reg.summary$bic
```

```r
par(mfrow=c(1,2))
# Plotting Adj. R^2 and BIC
# R-squared
plot(reg.summary$adjr2 ,xlab="Number of Variables ", ylab="Adjusted RSq",type="l") # 6 variables
# BIC
plot(reg.summary$bic ,xlab="Number of Variables ",ylab="BIC",type='l') # 4 variables

plot(regfit.full, scale='adjr2') # 69%
plot(regfit.full, scale='bic') # -56

# Forward, Backward, and Step-wise Selection
n1 = dim(pa_data_c)[1]

# Forward
fit0 <- lm(pvote ~ 1, data = pa_data_c)
fitall <- lm(pvote ~ ., data = pa_data_c)

# Forward
m1f <- step(fit0, fitall, direction = "forward", k = log(n1)) # use k = log(n) for BIC selection

# Backward
m1b <- step(fitall, direction = "backward", k = log(n1)) #

# Both
m3s <- step(fitall, direction = "both", k = log(n1)) #

# Results
summary(m1f)$coefficients
summary(m1b)$coefficients
summary(m3s)$coefficients

# K - Cross Validation
library(DAAG)
fm1 <- "pvote ~ pp + pv + pd_new" # Step-wise selection (BIC)
fm2 <- "pvote ~ ma + ms + pp + pv + pf + pd_new + pnh" # Adj. R^2

fm1cv <- CVlm(pa_data_c, form.lm = formula(fm1), m = 5, seed = 7, plotit = F) # 0.017
fm2cv <- CVlm(pa_data_c, form.lm = formula(fm2), m = 5, seed = 7, plotit = F) # 0.019

fm1.fit <- lm(pvote ~ pp + pv + pd_new, data = pa_data_c)
summary(fm1.fit) # SELECTED MODEL (66%)

fm2.fit <- lm(pvote ~ ma + ms + pp + pv + pf + pd_new + pnh, data = pa_data_c)
summary(fm2.fit)

autoplot(fm1.fit, which = 1:6, ncol = 2, label.size = 3)

#################################### SECOND MODEL ####################################
pa_data_c2 <- pa_data_c[-c(28),] # 10 observations total dropped
rownames(pa_data_c2) <- 1:nrow(pa_data_c2)

###### SEARCING FOR BEST MODEL (2) ######
### Will be using Adjusted R^2 and BIC for model selection criterion
```

```r
# All Possible Subsets
regfit.full2 <- regsubsets(pvote ~., data = pa_data_c2, nvmax=9)
reg.summary2 <- summary(regfit.full2)

names(reg.summary2)

reg.summary2$adjr2
reg.summary2$bic

par(mfrow=c(1,2))
# Plotting Adj. R^2 and BIC
# R-squared
plot(reg.summary2$adjr2 ,xlab="Number of Variables ", ylab="Adjusted RSq",type="l") # 5 variables
# BIC
plot(reg.summary2$bic ,xlab="Number of Variables ",ylab="BIC",type='l') # 4 variables

plot(regfit.full2, scale='adjr2') # 0.70
plot(regfit.full2, scale='bic') # -59

# Forward, Backward, and Step-wise Selection
n2 = dim(pa_data_c2)[1]

# Forward
fit0 <- lm(pvote ~ 1, data = pa_data_c2)
fitall <- lm(pvote ~ ., data = pa_data_c2)

# Forward
m1f2 <- step(fit0, fitall, direction = "forward", k = log(n2)) # use k = log(n) for BIC selection

# Backward
m1b2 <- step(fitall, direction = "backward", k = log(n2)) #

# Both
m3s2 <- step(fitall, direction = "both", k = log(n2)) #

# Results
summary(m1f2)$coefficients
summary(m1b2)$coefficients
summary(m3s2)$coefficients

# K - Cross Validation
library(DAAG)
fm1.2 <- "pvote ~ pp + pv + pd_new" # Step-wise selection (BIC)
fm2.2 <- "pvote ~ ma + ms + pp + pv + pd_new + pnh" # Adj. R^2

fm1cv2 <- CVlm(pa_data_c2, form.lm = formula(fm1.2), m = 5, seed = 7, plotit = F) # 0.017
fm2cv2 <- CVlm(pa_data_c2, form.lm = formula(fm2.2), m = 5, seed = 7, plotit = F) # 0.018

fm1.fit2 <- lm(pvote ~ pp + pv + pd_new, data = pa_data_c2)
summary(fm1.fit2) # SELECTED MODEL (68%)

fm2.fit2 <- lm(pvote ~ ma + ms + pp + pv + pd_new + pnh, data = pa_data_c2)
summary(fm2.fit2) (#69%)
```

```
autoplot(fm1.fit2, which = 1:6, ncol = 2, label.size = 3)

m3_r <- residuals(fm1.fit2)
plot(pa_data_c2$pvote, m3_r) # missing linear related model in data collection

############################### MULTICOLLINEARITY DIAGNOSTICS ##############################
# Will just do this for the 3rd iteration of the model
library(olsrr)
cor(pa_data_c2$pp, pa_data_c2$pvote) # 0.41
cor(pa_data_c2$pv, pa_data_c2$pvote) # 0.67
cor(pa_data_c2$pd_new, pa_data_c2$pvote) # 0.51

ols_coll_diag(fm1.fit2)
# No evidence of SEVERE multicollinearity (NO VIF > 10 & all tolerances > 0.1)
# looks like we have 2 sets of multicollinear data: (pp, pd_new), (pp, pv, pd_new)

################### Principal Component Analysis ###########################
# PCA serves as a way to deal with multicollinearity
# Creates components that are orthogonal, which means no multicollinearity between predictors
library(pls)

pcr_model <- pcr(pvote ~ ., data = pa_data_c2,
                 scale = T, center=TRUE, validation = "CV")

summary(pcr_model) # Variance explained by each pc

library(factoextra)
pcr_model2 <- prcomp(pa_data_c4, scale = TRUE)
print(pcr_model2)
summary(pcr_model2)
eig.val <- get_eigenvalue(pcr_model2)
eig.val
fviz_eig(pcr_model2, col.var="blue")

# Determining number of PC's to keep in the model (Threshold at 90% of variation explained)
par(mfrow=c(1,3))
validationplot(pcr_model)
validationplot(pcr_model, val.type="MSEP")
validationplot(pcr_model, val.type = "R2")

#### Looks like we want to keep 7 principal components
pcr_model_f <- pcr(pvote ~ ., data = pa_data_c2,
                   scale = TRUE, center = TRUE, ncomp = 5)

# Getting the model coefficients (have already been transformed back)
pca_coef <- coef(pcr_model_f, ncomp = 5, intercept = TRUE)
pca_coef

# Now we need to get results
# Fitted values
pcr_model_f_fitted <- pcr_model_f$fitted.values[,,5]
pcr_model_f_fitted
```

```r
# Residuals
pcr_model_f_residuals <- pcr_model_f$residuals[,,5]
pcr_model_f_residuals

# Plotting fitted vs. actual
par(mfrow=c(1,1))
plot(pcr_model_f_fitted, pa_data_c2$pvote) # Looks good

# R^2 for this model
cor(pcr_model_f_fitted, pa_data_c2$pvote)^2 # No better when using pca (68%)

summary(pcr_model_f)

# Create diagnostic plots
par(mfrow=c(2,2))
# Residuals vs Fitted
plot(pcr_model_f_fitted, pcr_model_f_residuals,
     xlab="Fitted Values", ylab="Residuals",
     main="Residuals vs Fitted")
abline(h=0, col="red", lty=2) # Looks good

# Normal Q-Q plot
qqnorm(pcr_model_f_residuals)
qqline(pcr_model_f_residuals) # Looks good

# Scale-Location plot
plot(pcr_model_f_fitted, sqrt(abs(pcr_model_f_residuals)),
     xlab="Fitted Values", ylab="Standardized Residuals",
     main="Scale-Location")

# Residuals vs Order (independence)
plot(pcr_model_f_residuals, type="b",
     xlab="Observation Order", ylab="Residuals",
     main="Residuals vs Order")
```