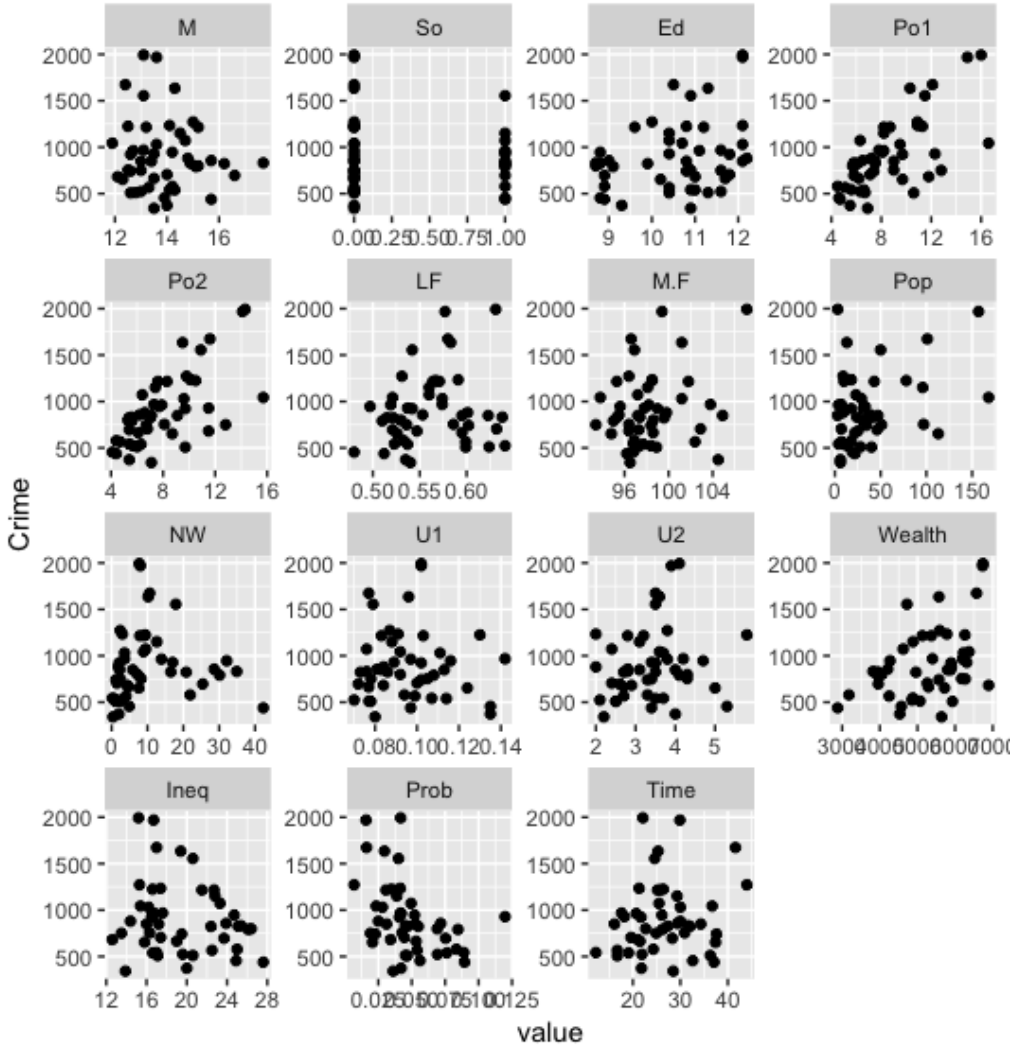
9.1

Please review the included 9.1.R file along with this explanation.

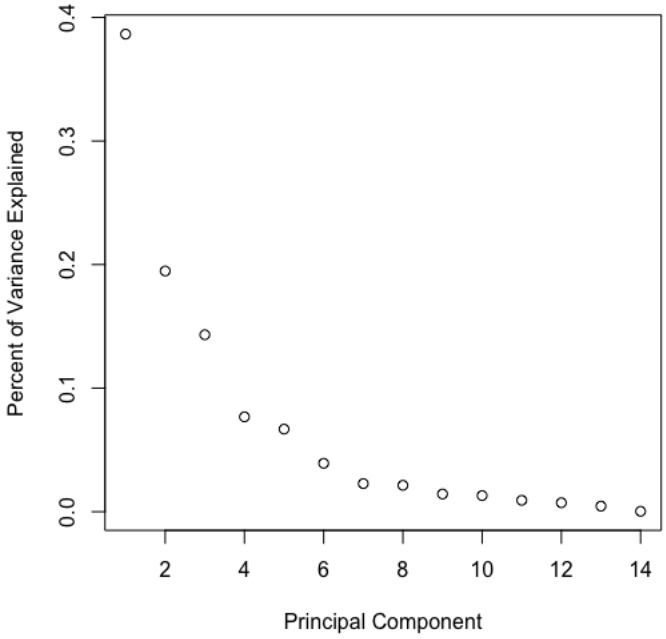
First, I loaded and inspected the data. With 16 variables and only 47 observations, it's likely that a model using all of the variables will heavily over fit.

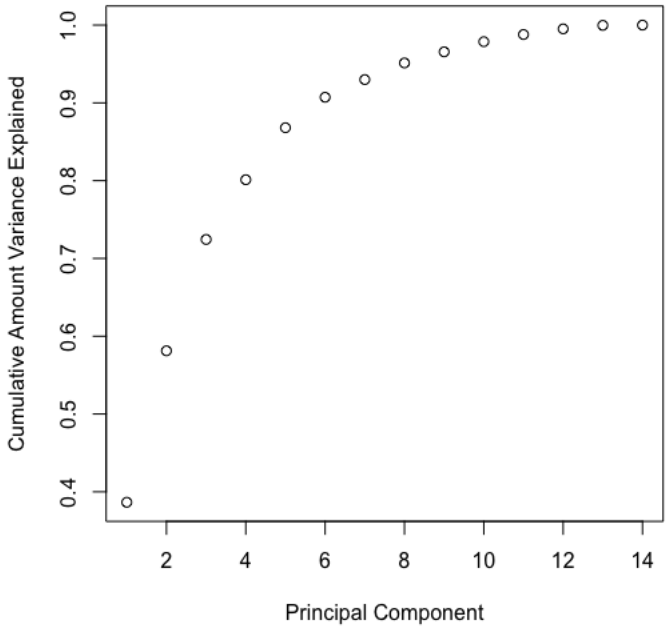
Because this is the same data set we used in week three for outlier detection, I again considered removing the two data points that are potential outliers. As a reminder, there are two potential outliers in the data set. The grubbs test finds two outliers at a p-value of .1, but none at .05. Based on this finding I would likely not remove them, but it could possibly help with model accuracy. Ultimately, I decided not to remove the outliers to so that I could accurately compare the PCA Model with my basic linear regression from week 5.

After deciding not to remove any potential outliers, I plotted each of the variables against crime rates in each city to review how each variable fit the model before applying PCA. From the chart below it is clear that So (Indicator for a Southern State) is a categorical variable. Because PCA does not work well on categorical data and works best on continuous numerical data, I removed So from the dataset.

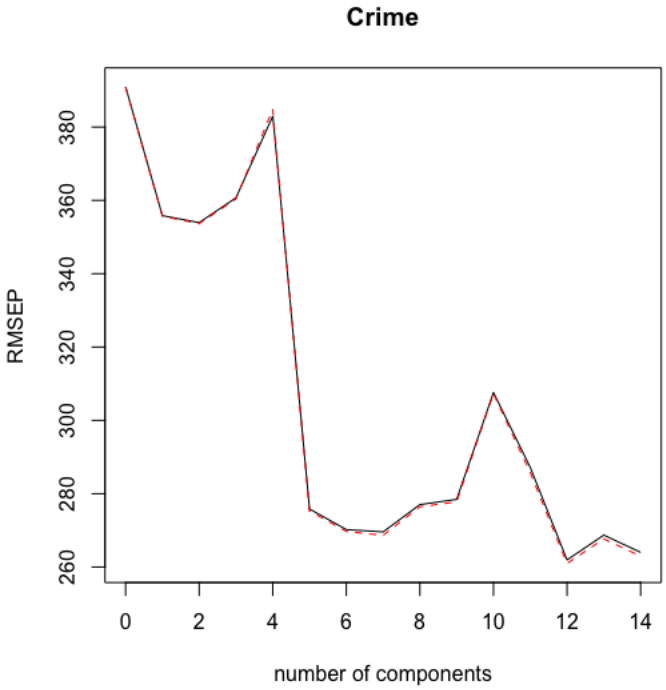


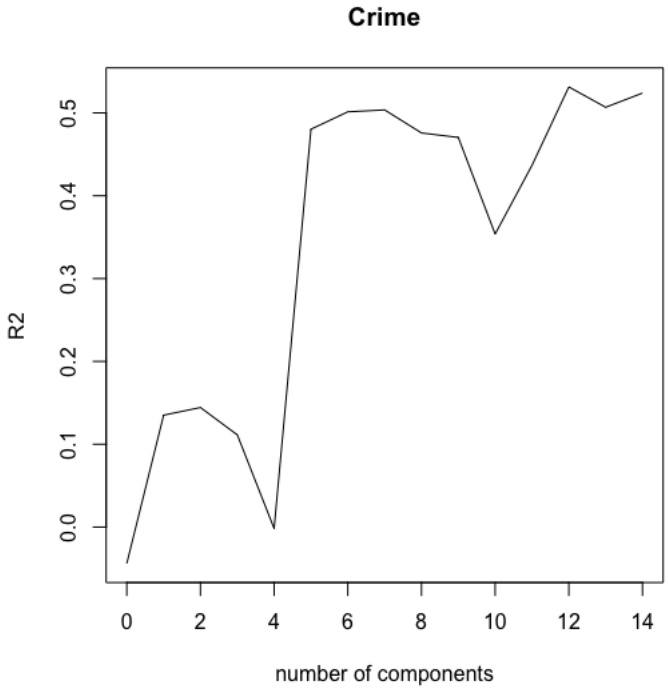
After removing So from the data, I used the pcr function from the pls package to create a principal component regression model with all 14 principal components, using leave-one-out cross-validation on scaled data. I then plotted the amount of variance explained by each principal component and the cumulative sum of the variance explained by each grouping of principal components as shown below.





Based on these plots it looks like the model will be best with around 5 or 6 principal components. It's possible to get a better idea of the ideal number of principal components by plotting root mean squared error and R2 on the cross-validation models with each set of principal components.





There is a large improvement in R2 and RMSE when moving from 4 to 5 components, and a further slight improvement when moving from 5 to 6. Adding more than 6 components actually worsens these metrics until 12 components are reached, but the improvement with 12 components is likely due to overfitting and defeats much of the purpose of a PCA approach.

I ultimately decided to use six principal components due to the slight improvement over using 5, although I think you could also make a valid argument to only use 5, as that is where the major improvement in model quality occurs. I created the model using 6 principal components, and then moved on to predict the crime rate in the hypothetical test city.

I created a test city dataframe, excluding the So data point as it was not used in creation of the model. I used the pls predict function on the test city data frame with the 6 component model, with a resulting crime rate of 1302. This is very close to the prediction of my best model from last week (1334) so it looks like the 6 component PCA model is calculating a similar result as the simple linear model from last week.

In addition to the calculating the crime rate using pls built in predict function, I decided to manually calculate the crime rate by converting the principal components back to original coefficients and plugging the values from the test city into a linear equation with those coefficients. The PCR function includes scaled coefficients for each of the original variables for one of its outputs, so I first extracted the scaled coefficients from the 6 component model, then divided them by the scaling factor for each coefficient to remove the scaling.

The equation on the converted coefficients is below:

-5717.97 + 95.40*M + 13.47*ED + 124.09*Po1 + 119.59*Po2 + 43.19*LF + 107.62*M.F 28.79*Pop + 103.81*NW + 2.76*U1 +29.06*U2 + 38.51*Wealth + 8.49*Ineq - 43.15*Prob + 35.34*Time

If you plug the test city values into the equation above, it also returns a crime rate of 1302.

Last week, my best model had an r2 of about .7, significantly higher than the .5 of the six-component PCA model from this week. However, it's not fair to say that that model was better based on R2 alone. Last week's model was likely heavily overfit based on the high number of variables and low number of data points. It isn't possible to tell which model is truly better without testing them both on more data, but I would guess that the .5 R2 from this model is more likely to be how the model would perform in the real world.