**HMW 6**

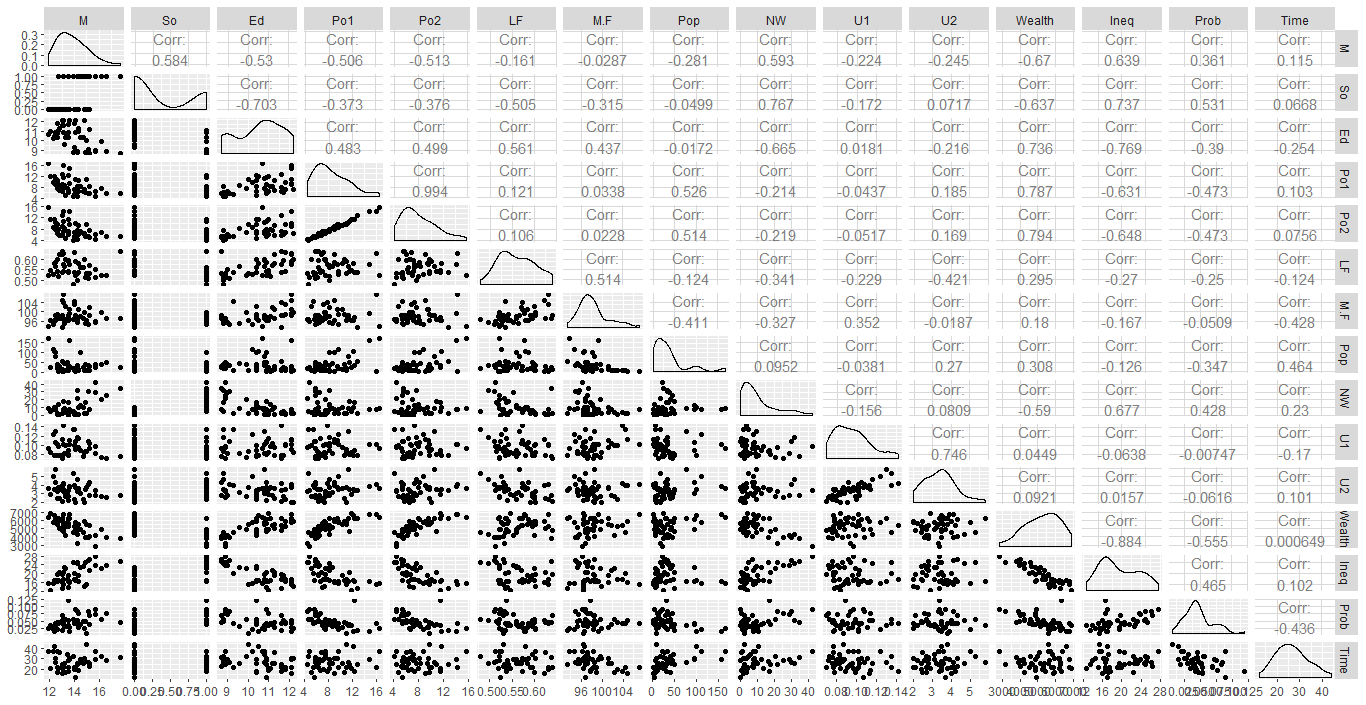
**Shafagh Yazdani**

**Q 9.1**

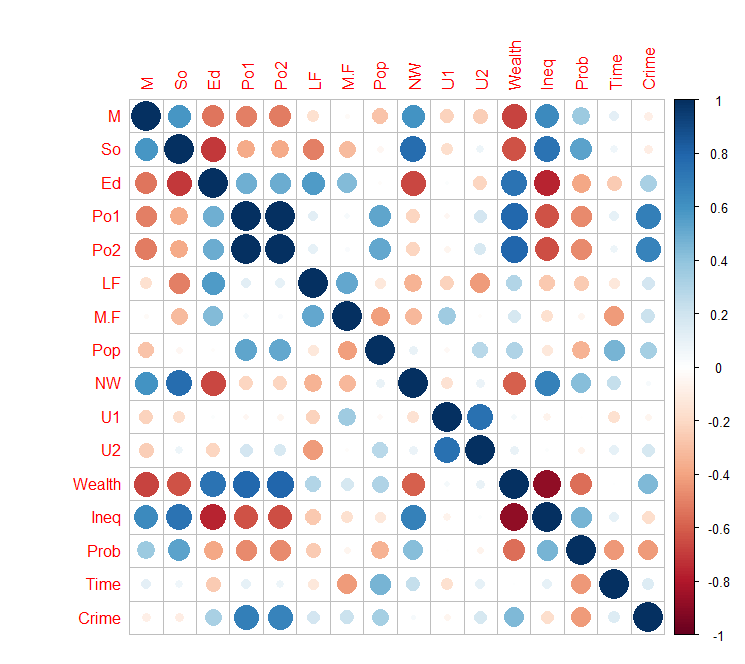
We saw in the dataset that so many of the predictors have correlation with each other. Which this cause the multicollinearity which can affect our coefficient value in our model.

One way to reduce this collinearity in our data set is that to use PCA to overcome the overfitting issue.

We see that many of our predictors are correlated to each other. We can run this for all of our data set to see which ones are correlated. To illustrate this phenomenon, we begin by generating a covariance plot for the 15-predictor Crime dataset.



Clearly, the variables Po1 and Po2 have extremely high cross-correlation (0.994). Additionally, NW and SO have a strong correlation (0.767), as do U1 and U2 (0.746). And in terms of negative correlation (just as problematic), Wealth and Ineq are strongly correlated (-0.884). Finally, SO and Ed share a fairly strong negative cross-correlation (-0.703). To remedy this, we use PCA to transform the dataset into a new, 15-ary orthogonal set of bases such that the new, transformed predictors have no cross-correlation.

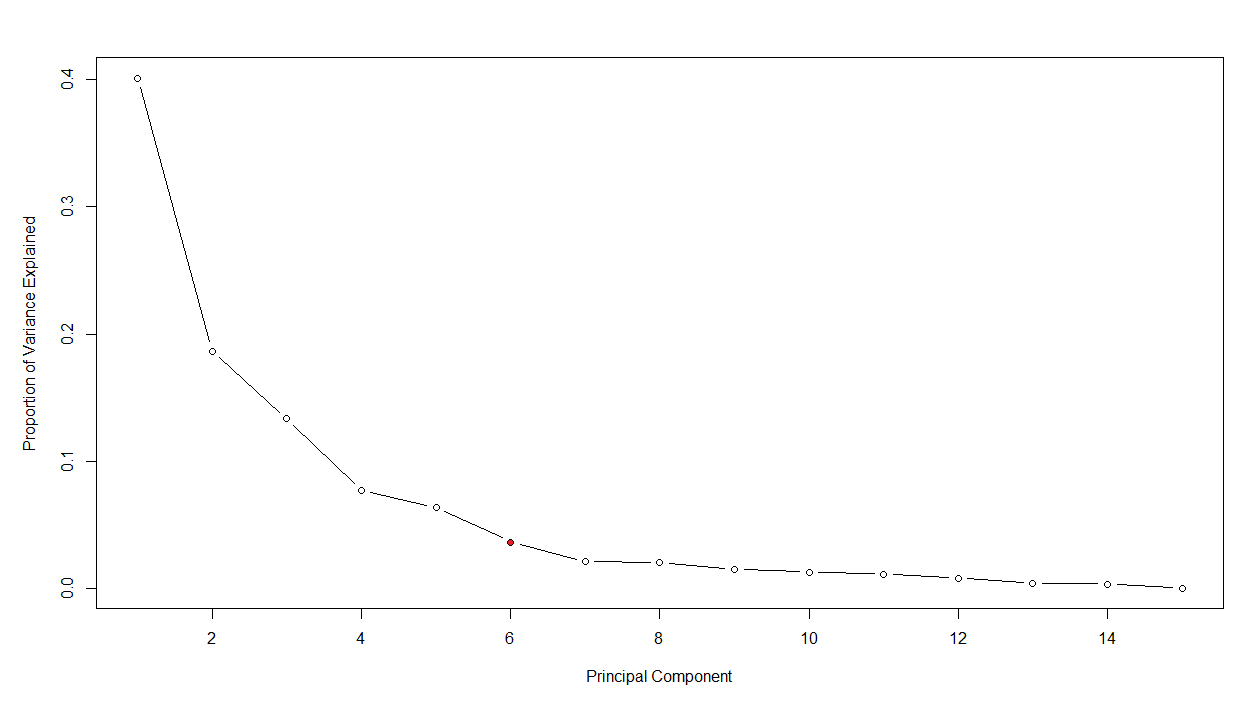
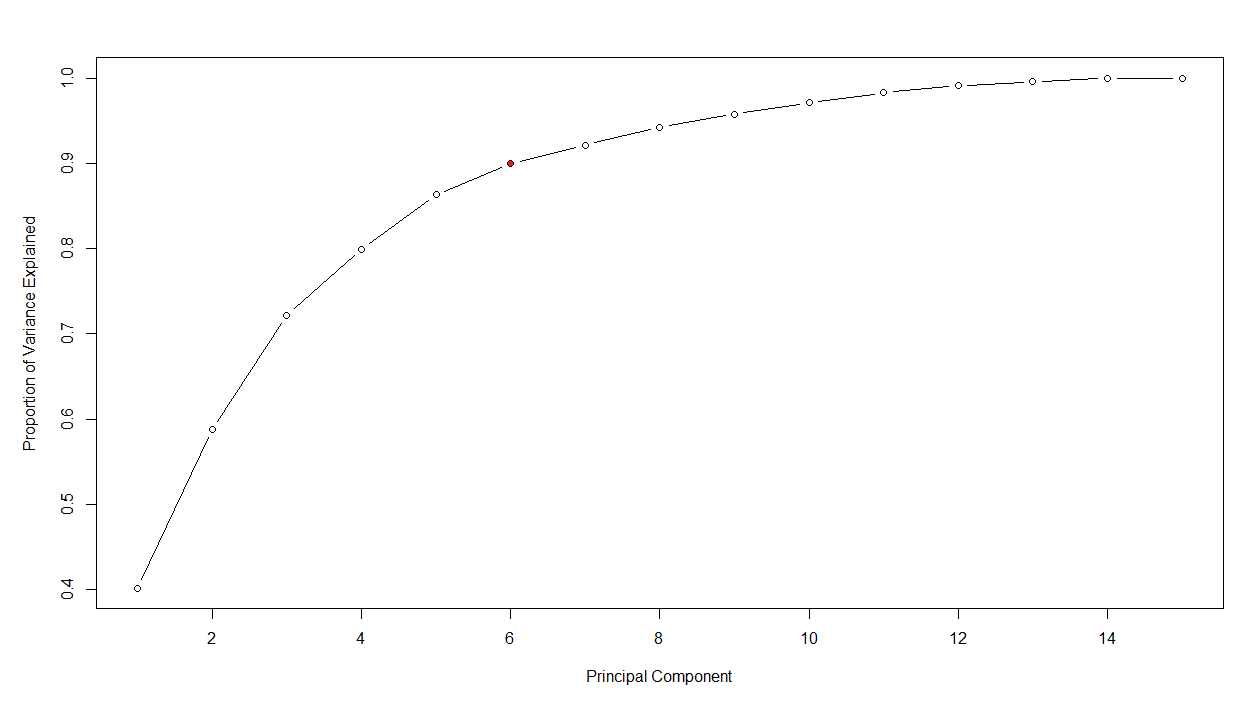


We run the PCA on the scaled data set.

|  |
| --- |
| Importance of components:  PC1 PC2 PC3 PC4  Standard deviation 2.4534 1.6739 1.4160 1.07806  Proportion of Variance 0.4013 0.1868 0.1337 0.07748  Cumulative Proportion 0.4013 0.5880 0.7217 0.79920  PC5 PC6 PC7  Standard deviation 0.97893 0.74377 0.56729  Proportion of Variance 0.06389 0.03688 0.02145  Cumulative Proportion 0.86308 0.89996 0.92142  PC8 PC9 PC10  Standard deviation 0.55444 0.48493 0.44708  Proportion of Variance 0.02049 0.01568 0.01333  Cumulative Proportion 0.94191 0.95759 0.97091  PC11 PC12 PC13  Standard deviation 0.41915 0.35804 0.26333  Proportion of Variance 0.01171 0.00855 0.00462  Cumulative Proportion 0.98263 0.99117 0.99579  PC14 PC15  Standard deviation 0.2418 0.06793  Proportion of Variance 0.0039 0.00031  Cumulative Proportion 0.9997 1.00000 |
|  |
| |  | | --- | | > | |

Studying the summarized results above, one can see that PC1 alone accounts for over 40% of the transformed dataset’s variance, with each additional Principal Component contributing progressively less to the total proportion of variance.

We can plot variance of each of the PCA to visualize and to decide which PCA we should use.



These two plots clearly illustrate that 90% or more of the transformed dataset’s variance can be explained by using just the first 5 or 6 Principal Components. Therefore, we keep PCA1 – PCA6.

Therefore, we pick 15 factor and transformed to 6 PCA. we build a linear regression model with the first 6 principal components.

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 905.09 35.35 25.604 < 2e-16 \*\*\*

PC1 65.22 14.56 4.478 6.14e-05 \*\*\*

PC2 -70.08 21.35 -3.283 0.00214 \*\*

PC3 25.19 25.23 0.998 0.32409

PC4 69.45 33.14 2.095 0.04252 \*

PC5 -229.04 36.50 -6.275 1.94e-07 \*\*\*

PC6 -60.21 48.04 -1.253 0.21734

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 242.3 on 40 degrees of freedom

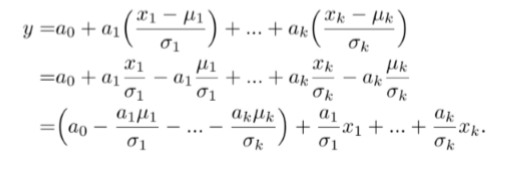
Multiple R-squared: 0.6586, Adjusted R-squared: 0.6074

F-statistic: 12.86 on 6 and 40 DF, p-value: 4.869e-08

we must transform the PCI model coefficients into new values expressed in terms of the original (non-orthogonal) bases.

In order to implement this inverse transformation process, we must access the eigenvectors used to translate into the PCI domain, as well as the scale factors (mean and standard deviation, on a predictor-by-predictor basis)

In the PCI transformation, the original dataset was scaled (such that it becomes zero-mean with unit standard deviation). Thus, to convert a linear model from the PCI domain back to the original domain, the data must be “unscaled.”



Specifically, the first line of the equation represents the model transformed back from the PCI domain, using scaled data from the original dataset (subtract the mean, divide by sigma). But to invert the scaling, we need to re-express this in terms of a new constant (the intercept) and new coefficients that multiply the original x-values (the unscaled data). By rearranging the terms in the second and third lines, we achieve this goal: the final term in parenthesis is the new unscaled intercept, and the aj/j terms are the new unscaled x-coefficients.

***Transform to original coordinates***

PC1 65.21593

PC2 -70.08312

PC3 25.19408

PC4 69.44603

PC5 -229.04282

PC6 -60.21329

***Our new coefficient***

***a\_vals\_6 = PCA$rotation[,1:6] %\*% b\_vals\_6/ sigma***

M 6.989232e+01

So 9.165344e+01

Ed 1.829254e+01

Po1 4.142536e+01

Po2 4.243282e+01

LF 1.135641e+03

M.F 3.821603e+01

Pop 6.812930e-01

NW 9.237458e+00

U1 1.009450e+02

U2 3.486602e+01

Wealth 4.689284e-02

Ineq 1.434742e+00

Prob -2.274397e+03

Time 5.097975e+00

***Our new intercept***

***a0\_15 <- b0\_15 - sum(a\_vals\_6\*mu)***

(Intercept)

-5923.647

Finally, the prediction of crime rate for the “New City” data point with the PCI-15 model is:

**1248.427**

**Here are Summary of regression model prediction with PCA and Cross validation and comparison with hmw 5 output**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Principal Components* | *# of Components w/ p<0.05* | *R2* | *4-fold*  *R2* | *Adjusted R2* | *4-fold Adjusted R2* | *Predicted (New Value)* |
| {1, … , 6} | 6 | **0.659** | **0.501** | **0.607** | **0.427** | **1248** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Principal Components* | *# of Components w/ p<0.05* | *R2* | *Adjusted R2* | *Predicted (New Value)* |
| {1, … , 15} | 15 | **0.803** | **0.708** | **155** |

|  |  |  |  |
| --- | --- | --- | --- |
| *Hmw 5* | *R2* | *Adjusted R2* | *Predicted (New Value)* |
| On 15 variables | **0.8031** | **0.7078** | **1304** |
| *Hmw 5* | *R2* | *Adjusted R2* | *4-fold Adjusted R2* | *Predicted (New Value)* |
| 6 selected Varirable | **0.766** | **0.731** | **0.731** | **155** |

As we can see the PCA 4 fold cross validation adjusted R is lowest. Therefor, we better chosen the regression model with 6 variable selection using 4 fold cross validation.