ISYE 6501 Intro Analytics Modeling – HW6

Question 9.1 Using the same crime data set `uscrime.txt` as in Question 8.2, apply Principal Component Analysis and then create a regression model using the first few principal components. Specify your new model in terms of the original variables (not the principal components), and compare its quality to that of your solution to Question 8.2. You can use the R function `prcomp` for PCA. (**Note** that to first scale the data, you can include `scale. = TRUE` to scale as part of the PCA function. Don't forget that, to make a prediction for the new city, you'll need to unscale the coefficients (i.e., do the scaling calculation in reverse)!)

From the lecture, we get to know PCA can deal with 2 data issues, 1. high dimensional predictors and 2. High correlation predictors. Before using PCA, let's take a look at the data to see if our data also have these two problems. The data set only have 47 data points, 15 predictors cannot be called high dimensional but still very likely to cause overfit for such a small dataset. From the correlogram, we can see they are some highly correlated variables e.g. Po1 and Po2, U1 and U2. *(Fig1)*

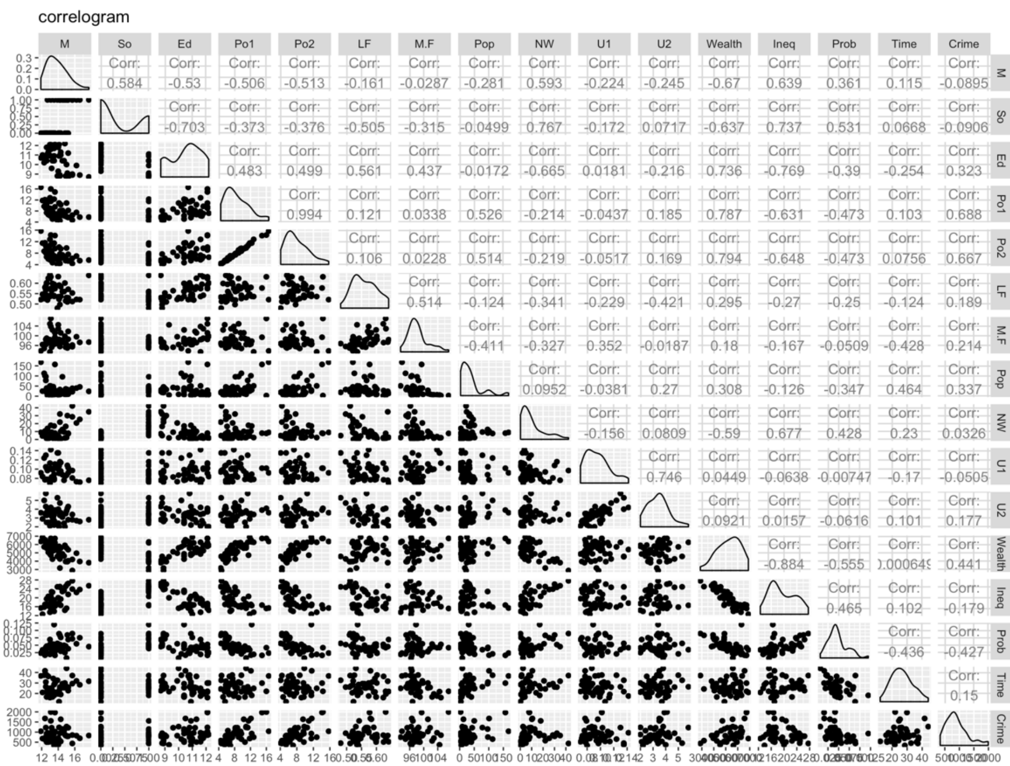

Fig1

Then I created a regular regression model as I did in the HW5 but use the scaled independent variables, I got Sum of squared error: 554,100 MSE: 17,873 for the train set and MSE: 94,588 for the test set. Which is overfitted.

```
Call:  glm(formula = Crime ~ ., family = gaussian, data = train)
Coefficients:
(Intercept)            M           So           Ed          Po1          Po2           LF          M.F
     888.79       112.21       -60.76       178.09      1411.54     -1177.96       -55.73        23.85
        Pop           NW           U1           U2       Wealth         Ineq         Prob         Time
     -19.80       101.48      -122.34       123.04        43.28       227.22      -213.81      -182.86
Residual Deviance: 554100
AIC: 425.5
```

Then I used *prcomp* function from R to make PCA for independent variables. The first principle component can explain 40% of the data variances. And with the top 6 components, 90% of data variances can be explained *(Fig2).* I also plotted the % of contributions of independent variables to PC1 and PC2 *(Fig3).* I plotted all data points using PC1 and PC2 as x and y axes, colored by qualities of representation. The closer to the center, the smaller the qualities of representation are *(Fig4).* From fig5, we can see the variables contribution at PC1 and PC2.

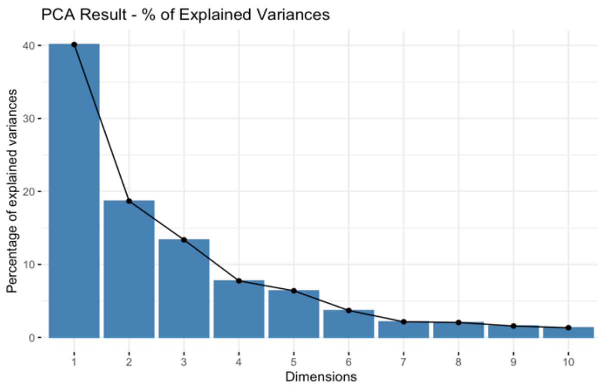

Fig2

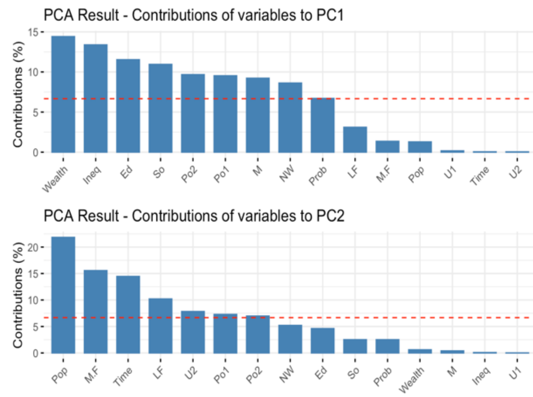

Fig3

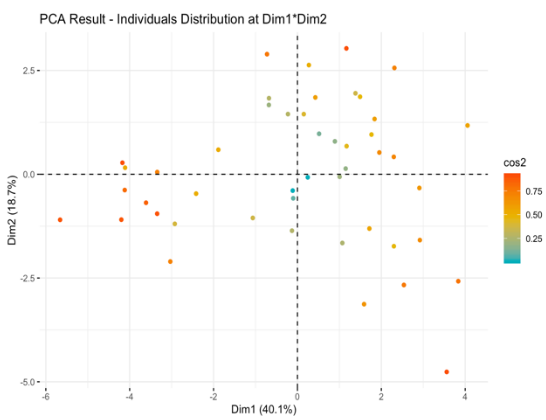

Fig4

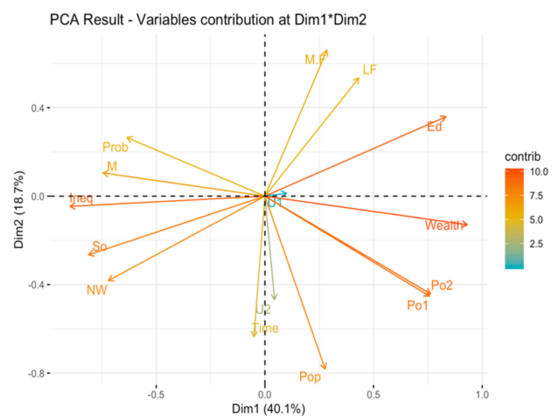

Fig5

To figure out what will be the best number of components to keep, I created 15 linear regression models by keeping a different number of components and plotted the MSE_Test and MSE_Train. Orange and green reference lines are the test and train MSE of the regular linear regression model from the first step. When K smaller than 5, the PCA_GLMs are under fitted with MSE smaller than the MSE _test of regular GLM. But with k from 5 to 14, all models are performed better than regular GLM. Here I choose k=7, which has the closest MSE_train and MSE_test and can explain 92% variances. *(Fig6)*

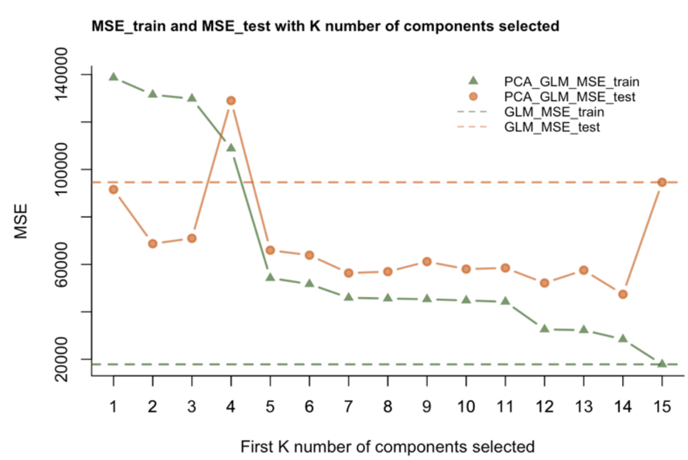

Fig6

Through some calculate (see code for details), I get the unscaled intercept and coefficients. The predicted Crime for the new city is **1120.944**.

```
Coefficients of PCs from Linear Regression
  (Intercept)       PC1        PC2        PC3        PC4        PC5        PC6        PC7
   909.0727  76.05102  -52.42113   27.81874  108.05693 -211.03281  -64.07115  144.85501

Calculate Coefficients of Variables from Eigenvectors and PC Coefficients
  (Intercept)           M          So          Ed         Po1         Po2          LF         M.F
     909.0727   57.849489   75.529943   -3.899501  142.428069  141.644398   16.504733  133.095923
          Pop          NW          U1          U2      Wealth        Ineq        Prob        Time
    27.368430   45.400514  -33.957421    7.003625   31.856040   47.502467 -112.454687  -57.207877

Unscaled Coefficients by Variables mean and SD
  (Intercept)           M          So          Ed         Po1         Po2          LF         M.F
    -5104.849    46.03053    157.6907   -3.485744    47.92496    50.65727    408.4136    45.16723
          Pop          NW          U1          U2      Wealth        Ineq        Prob        Time
    0.7188751    4.415155   -1883.512    8.292779  0.03301454    11.90656   -4945.895   -8.072347
```

Code

```
# ISYE 6501 Intro Analytics Modeling - HW6
# IP uscrime.txt

library(GGally)
library(factoextra)
library(gridExtra)
library(plyr)


# Loading and examining data
df<-read.delim("uscrime.txt", header = TRUE, sep = "\t")
ggpairs(df, title="correlogram")  #pair-wise correlation

##scaling 0-100##
fn <- function(x) scale(x,  scale = TRUE)
df_scaled<-as.data.frame(lapply(df[,-16], fn))
df_scaled$Crime<-df$Crime

#spiliting test and train
set.seed(666)
g <- sample(1:2,size=nrow(df_scaled),replace=TRUE,prob=c(0.7,0.3))
train <- df_scaled[g==1,]
test <- df_scaled[g==2,]

# Fit glm model: gaussian model
glm_model1<-glm(Crime~.,family = gaussian,train)
MSE_train1<-mean(glm_model1$residuals^2) #MSE Train
confint(glm_model1) # 95% CI for the coefficients

p1_test<-predict(glm_model1,test,type="response")
p1_residials<-p1_test-test$Crime
```

```r
MSE_test1<-mean(p1_residials^2) #MSE Train

#PCA
pca<-prcomp(df_scaled[,-16])
summary(pca)

#plot dimensions explained variances
fviz_eig(pca,title="PCA Result - % of Explained Variances")

# Contributions of variables to PC1
g1<-fviz_contrib(pca, choice = "var", axes = 1,title="PCA Result - Contributi
ons of variables to PC1")
g2<-fviz_contrib(pca, choice = "var", axes = 2,title="PCA Result - Contributi
ons of variables to PC2")
grid.arrange(g1, g2, nrow = 2)

#Individuals Distribution at Dim1*Dim2
F1<-fviz_pca_ind(pca,
             col.ind = "cos2", # Color by the quality of representation
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             title="PCA Result - Individuals Distribution at Dim1*Dim2",label
=FALSE
)

#Variables contribution at Dim1*Dim2
F2<-fviz_pca_var(pca,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             title="PCA Result - Variables contribution at Dim1*Dim2",
             repel = TRUE     # Avoid text overlapping
)
grid.arrange(F1, F2, nrow = 1)


summary_table <- data.frame()
for (k in c(1:15)){
  # Fit glm model: gaussian model_pca result
  summary_table[k,'k']<-k
  PCA_df <- as.data.frame(cbind(pca$x[,1:k],'Crime'=df$Crime))
  PCA_train <- PCA_df[g==1,]
  PCA_test <- PCA_df[g==2,]

  glm_model2<-glm(Crime~.,family = gaussian, PCA_train)
  assign(paste0("glm_model_k", k), glm(Crime~.,family = gaussian, PCA_train))
  summary_table[k,'MSE_train']<-mean(glm_model2$residuals^2) #MSE Train

  p2_test<-predict(glm_model2,PCA_test,type="response")
  p2_residials<-p2_test-PCA_test$Crime
  summary_table[k,'MSE_test']<-mean(p2_residials^2) #MSE Train
```

```
}

#plot error
p2<-plot(summary_table$MSE_train~summary_table$k, type="b" , bty="l", xlab="F
irst K number of components selected" , ylab="MSE" , col=rgb(0.2,0.4,0.1,0.7)
 , lwd=2 , pch=17)+
   lines(summary_table$MSE_test~summary_table$k, col=rgb(0.8,0.4,0.1,0.7) , l
wd=2 , pch=19 , type="b" )+
   abline(h=c(MSE_train1,MSE_test1), col=c(rgb(0.2,0.4,0.1,0.7),rgb(0.8,0.4,0
.1,0.7)), lty=c(2,2), lwd=c(2,2))+
   axis(side=1, at=seq(1, 15, by=1), labels =c(1:15))+
  title("MSE_train and MSE_test with K number of components selected",adj =0,
cex.main=0.9)+
  legend("topright",
         legend = c("PCA_GLM_MSE_train", "PCA_GLM_MSE_test", "GLM_MSE_train",
"GLM_MSE_test"),
         col = c(rgb(0.2,0.4,0.1,0.7),
                 rgb(0.8,0.4,0.1,0.7),
                 rgb(0.2,0.4,0.1,0.7),
                 rgb(0.8,0.4,0.1,0.7)),
         pch = c(17,19,NA,NA),
         lty = c(NA,NA,2,2),
         bty = "n",
         pt.cex = 1,
         cex = 0.8,
         text.col = "black",
         horiz = F
        )

##select first 7 PC calculate variables' coef
pca_glm_coef<-glm_model_k7$coefficients[2:8] # coefficients from pca glm
Intercept<-glm_model_k7$coefficients[1]      # intercept from pca glm

eigenvectors<-pca$rotation[,1:7]             # eigenvectors of pc1 to pc7
coeff<-colSums(t(eigenvectors)*pca_glm_coef) # coeff for scaled x

sd_df<-apply(df[,-16], 2, sd)                # df variables sd
mean_df<-apply(df[,-16], 2, mean)            # df variables mean

coeff_unscale<-t(coeff)/sd_df                # unscale coefficients
Intercept_unscale<-Intercept-colSums(t(t(coeff)*mean_df)/sd_df)   # unscale i
ntercept

#new data
new<-c(14.0, 0, 10.0, 12.0, 15.5, 0.640, 94.0, 150, 1.1, 0.120, 3.6, 3200, 20
.1, 0.04, 39.0)
new_predict<-sum(t(new)*coeff_unscale)+Intercept_unscale
```