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)!)

Answer:
I have tried to answer the questions along with Rcodes as below:

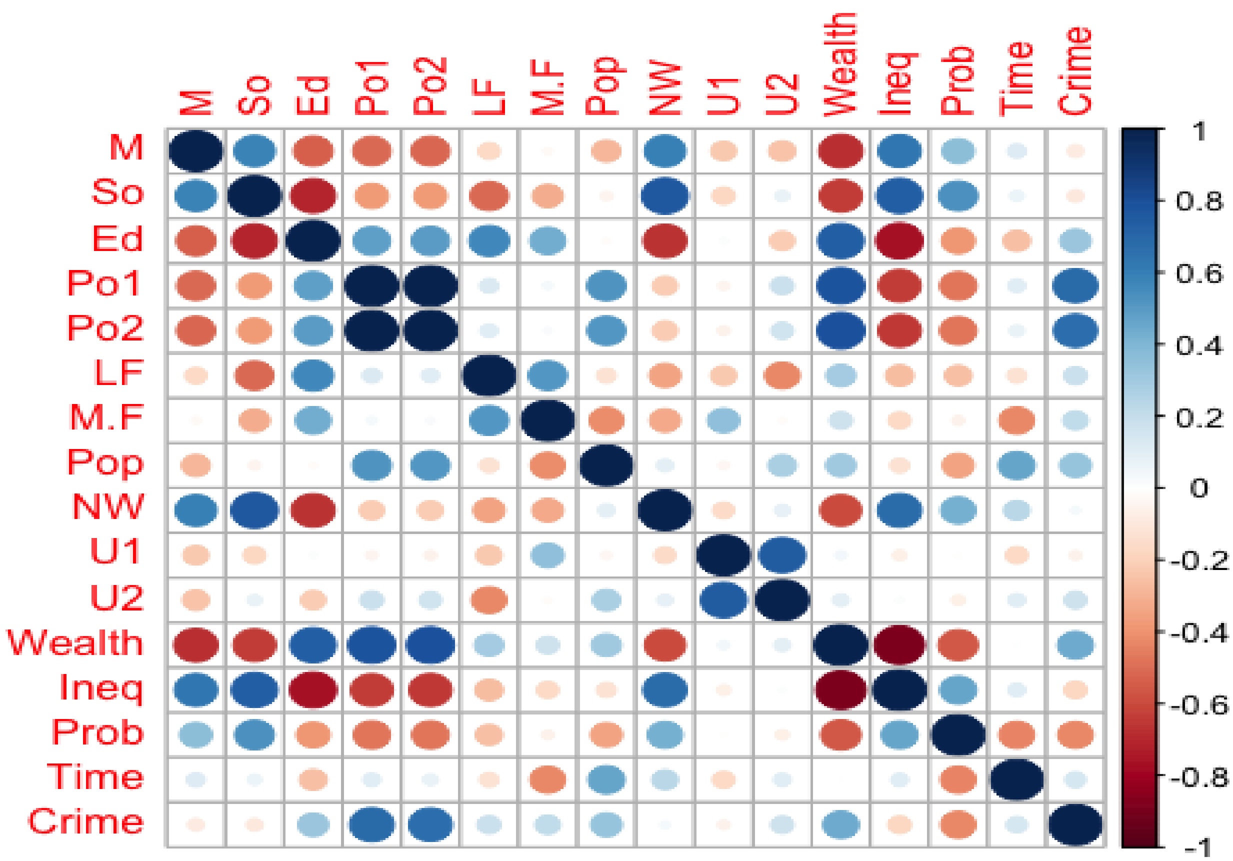
```
#read the dataset
rm(list = ls())
getwd()
setwd("/Users/admin/Desktop/ISYE6501")
uscrime <- read.csv("uscrime.csv", stringsAsFactors = FALSE, header = TRUE)
```

```
#let's see the correlation between the predictors first here
> corr <- cor(uscrime)
> round(corr, 2)
```

	M	So	Ed	Po1	Po2	LF	M.F	Pop	NW	U1	U2	Wealth	Ineq	Prob	Time	Crime
M	1	0.58	-0.53	-0.51	-0.51	-0.16	-0.03	-0.28	0.59	-0.22	-0.24	-0.67	0.64	0.36	0.11	-0.09
So	0.58	1	-0.7	-0.37	-0.38	-0.51	-0.31	-0.05	0.77	-0.17	0.07	-0.64	0.74	0.53	0.07	-0.09
Ed	-0.53	-0.7	1	0.48	0.5	0.56	0.44	-0.02	-0.66	0.02	-0.22	0.74	-0.77	-0.39	-0.25	0.32
Po1	-0.51	-0.37	0.48	1	0.99	0.12	0.03	0.53	-0.21	-0.04	0.19	0.79	-0.63	-0.47	0.1	0.69
Po2	-0.51	-0.38	0.5	0.99	1	0.11	0.02	0.51	-0.22	-0.05	0.17	0.79	-0.65	-0.47	0.08	0.67
LF	-0.16	-0.51	0.56	0.12	0.11	1	0.51	-0.12	-0.34	-0.23	-0.42	0.29	-0.27	-0.25	-0.12	0.19
M.F	-0.03	-0.31	0.44	0.03	0.02	0.51	1	-0.41	-0.33	0.35	-0.02	0.18	-0.17	-0.05	-0.43	0.21
Pop	-0.28	-0.05	-0.02	0.53	0.51	-0.12	-0.41	1	0.1	-0.04	0.27	0.31	-0.13	-0.35	0.46	0.34
NW	0.59	0.77	-0.66	-0.21	-0.22	-0.34	-0.33	0.1	1	-0.16	0.08	-0.59	0.68	0.43	0.23	0.03
U1	-0.22	-0.17	0.02	-0.04	-0.05	-0.23	0.35	-0.04	-0.16	1	0.75	0.04	-0.06	-0.01	-0.17	-0.05
U2	-0.24	0.07	-0.22	0.19	0.17	-0.42	-0.02	0.27	0.08	0.75	1	0.09	0.02	-0.06	0.1	0.18
Wealth	-0.67	-0.64	0.74	0.79	0.79	0.29	0.18	0.31	-0.59	0.04	0.09	1	-0.88	-0.56	0	0.44
Ineq	0.64	0.74	-0.77	-0.63	-0.65	-0.27	-0.17	-0.13	0.68	-0.06	0.02	-0.88	1	0.47	0.1	-0.18
Prob	0.36	0.53	-0.39	-0.47	-0.47	-0.25	-0.05	-0.35	0.43	-0.01	-0.06	-0.56	0.47	1	-0.44	-0.43
Time	0.11	0.07	-0.25	0.1	0.08	-0.12	-0.43	0.46	0.23	-0.17	0.1	0	0.1	-0.44	1	0.15
Crime	-0.09	-0.09	0.32	0.69	0.67	0.19	0.21	0.34	0.03	-0.05	0.18	0.44	-0.18	-0.43	0.15	1

#it might be much easier to view the correlation in the plot.

```
library(corrplot)
corrplot(cor(uscrime))
```



These correlation plots show that there are some correlations in between the predictors, and I believe converting them to PCA will remove the correlation.

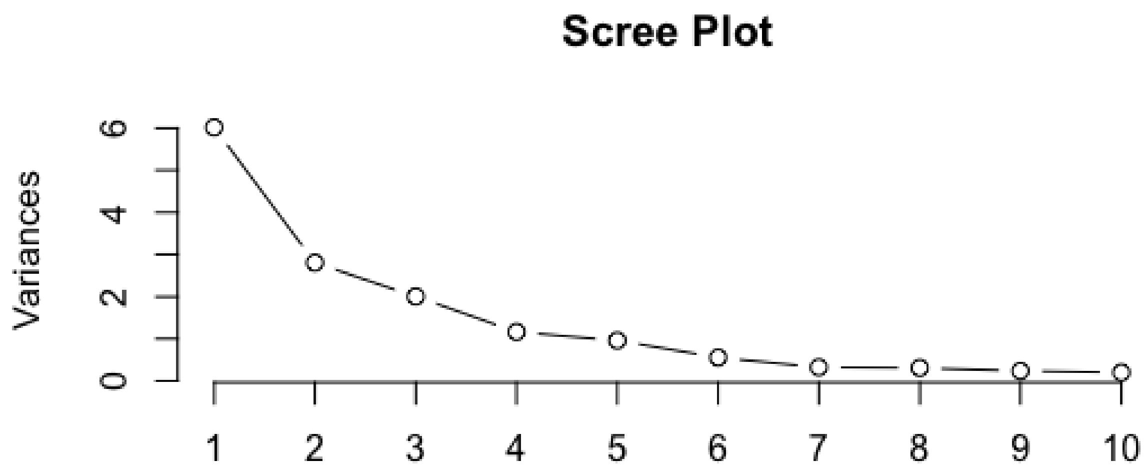
#I used prcorp function in R to convert the predictors from uscrime data set to Principle components. This function will scale= "TRUE" will scale the data and covert data in the same range.

```
uscrime.pca <- prcomp(uscrime[,1:15],scale.=TRUE,center=TRUE)
```

Summary of uscrime.pca shows that most of the variance is present in PC1 which slowly goes down with PC2, PC3 and so on.

Calculate the variances and proportion of variances from the PC object

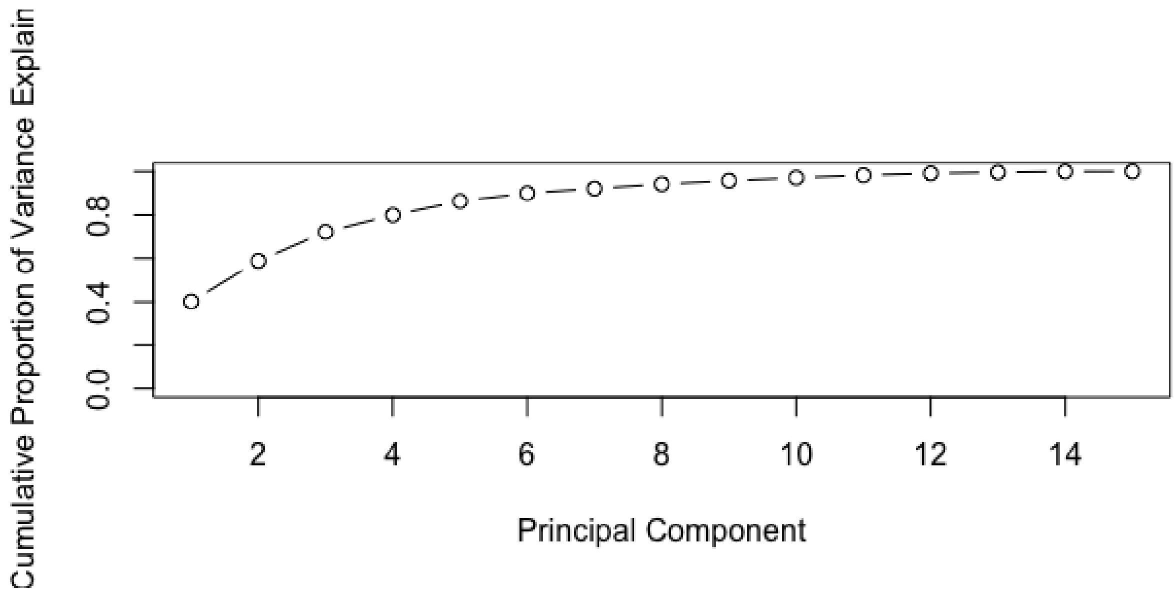
```
# Plot the Screeplot of variances from PCA
screeplot(uscrime.pca, main = "Scree Plot", type = "line")
```



looking at a scree plot is one of the valid method to select the number of PCs we are going to use. Here it looks like first 4,5 or 6 PC selection would be good.

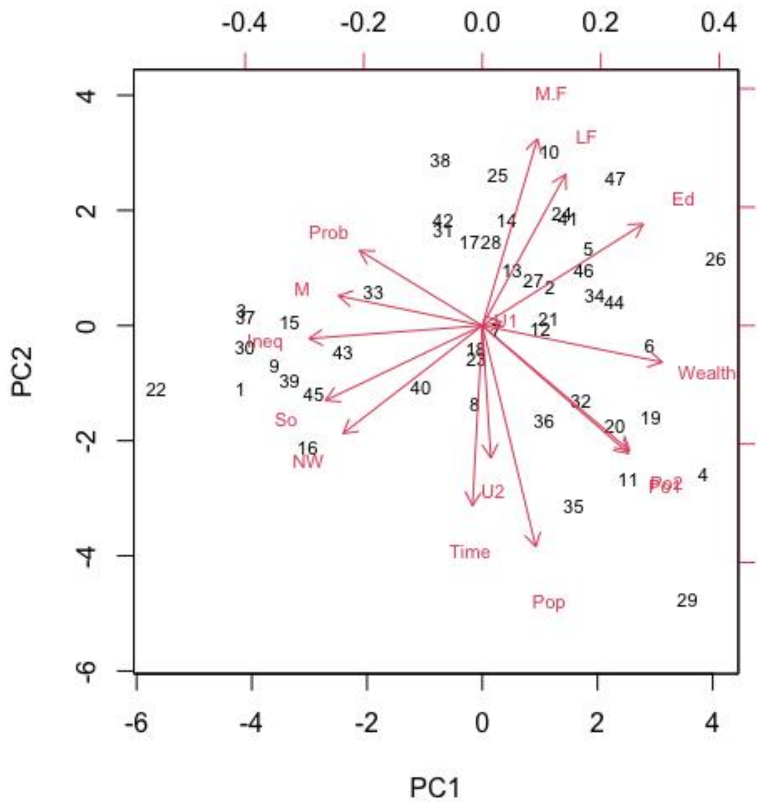
##Plot the cumsum proportion of variances from PCA

```
cumsum(propVAR)
plot(cumsum(propVAR), xlab = "Principal Component", ylab = "Cumulative Proportion of Variance Explained",ylim = c(0,1), type = "b")
```



This graph also shows similar idea. Here it looks like first 4,5 or 6 PC selection would be good. I am planning to use top 5 PCs to develop a linear regression model.

> biplot(uscrime.pca,scale=0, cex=.5)



Biplots shows PC1 and PC2 and how datapoints lie in the component space. Graph is kind of messy so I am not going to interpret it here.

Now let's use 5 PCs to create a new data frame and use it to prepare a linear model

FivePCs <- uscrime.pca$x[,1:5]

PC5 <- cbind(FivePCs, uscrime[,16]) #Create new data matrix with first 5 PCs and crime rate

PC5df <- as.data.frame(PC5)# make it data frame
model1 <- lm(V6~., data = PC5df) #Create regression model on new data matrix

summary(model1)
all:
lm(formula = V6 ~ ., data = PC5df)

Residuals:

```
   Min     1Q Median     3Q    Max 
-420.8 -185.0   12.2  146.2  447.9 
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    905.1       35.6   25.43  < 2e-16 ***
PC1             65.2       14.7    4.45  6.5e-05 ***
PC2            -70.1       21.5   -3.26   0.0022 **
PC3             25.2       25.4    0.99   0.3272
PC4             69.4       33.4    2.08   0.0437 *
PC5           -229.0       36.8   -6.23  2.0e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 244 on 41 degrees of freedom
Multiple R-squared:  0.645,	Adjusted R-squared:  0.602
F-statistic: 14.9 on 5 and 41 DF,  p-value: 2.45e-08
```

The model has very less difference in-between Multiple Squared value and Adjusted R squared value.
Overall summary shows that PC3 is not significant but other principle components are significant)
****

#Let's try to develop a model eliminating PC3 from a linear regression.

```
> model2 <- lm (V6 ~ PC1 + PC2 +PC4 + PC5, data = PC5df)
> summary(model2)

Call:
lm(formula = V6 ~ PC1 + PC2 + PC4 + PC5, data = PC5df)

Residuals:
   Min     1Q Median     3Q    Max
-401.9 -181.5  -33.9  124.5  465.8

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    905.1       35.6   25.43  < 2e-16 ***
PC1             65.2       14.7    4.45  6.3e-05 ***
PC2            -70.1       21.5   -3.26   0.0022 **
PC4             69.4       33.4    2.08   0.0435 *
PC5           -229.0       36.7   -6.23  1.8e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 244 on 42 degrees of freedom
Multiple R-squared:  0.637,    Adjusted R-squared:  0.602
F-statistic: 18.4 on 4 and 42 DF,  p-value: 8.38e-09
```

model does not much show much difference in the multiple R squared between model2(0.637) and model1(0.645) and the Adjusted R-squared values(0.603) are same in the model2 and model1

```
> anova(model1, model2)
Analysis of Variance Table

Model 1: V6 ~ PC1 + PC2 + PC3 + PC4 + PC5
Model 2: V6 ~ PC1 + PC2 + PC4 + PC5
  Res.Df     RSS Df Sum of Sq    F Pr(>F)
1     41 2441394
2     42 2499935 -1    -58541 0.98   0.33
```

0.33 p-value proves that model 1 and model 2 are not significantly different.

Get coefficients in terms of original data from PCA coefficients
PCA Coefficients for this linear regression model

```
> Bs <- model1$coefficients[2:6]
> Bs
  PC1    PC2    PC3    PC4    PC5
 65.2  -70.1   25.2   69.4 -229.0
> B0 <- model1$coefficients[1]
> B0
(Intercept)
        905
```

Transform the PC coefficients into coefficients for the original variables

```
> alphas <- uscrime.pca$rotation[,1:5] %*% Bs
> t(alphas)
        M   So   Ed Po1 Po2   LF M.F  Pop   NW   U1   U2 Wealth Ineq  Prob
[1,] 60.8 37.8 19.9 117 111 76.3 108 58.9 98.1 2.87 32.3   35.9 22.1 -34.6
     Time
[1,] 27.2
```

these coefficients above are using scaled data, we need to convert them back to the original data. When scaling, this function subtracts the mean and divides by the standard deviation, for each variable. we can modify the constant term a0 by alpha*mean/sd
Here are the coefficients for unscaled data:

```
> OAlpha <- alphas/sapply(uscrime[,1:15],sd) # OAlpha is original alpha
> t(OAlpha)
        M So   Ed  Po1  Po2   LF  M.F  Pop   NW  U1   U2 Wealth Ineq  Prob
```

```
[1,] 48.4 79 17.8 39.5 39.9 1887 36.7 1.55 9.54 159 38.3 0.0372 5.54 -1524
    Time
[1,] 3.84
OB0 <- B0 - sum(alphas*sapply(uscrime[,1:15],mean)/sapply(uscrime[,1:15],sd))
> OB0
(Intercept)
     -5934
```

# So model using 5 top Principal components for dataset is:
Crime = -5934 + 48.4*M + 79*So + 17.8Ed + 39.5*Po1 + 39.9*Po2 + 1887*LF + 36.7 MF + 1.55 Pop + 9.54*NW + 159*U1 + 38.3*U2 + 0.0372*Wealth + 5.54* Ineq – 1524*Prob + 3.84*Time

```
#test data
testdat <- data.frame(M = 14.0,So = 0,Ed = 10.0,Po1 = 12.0,Po2 = 15.5,LF = 0.640,M.F = 94.0,Pop = 150, NW = 1.1,U1 = 0.120,U2 = 3.6,Wealth = 3200,Ineq = 20.1,Prob = 0.04,Time = 39.0)
#predict test crime
predictcrime = as.matrix(testdat[,1:15]) %*% OAlpha + OB0
> predictcrime
    [,1]
[1,] 1389
```

hence the predicted crime rate now is 1389, which is slightly different than the prediction of model using significant predictors from original dataset, which had predict the crime value of 1304(previous homework)

```
#Now let's cross-validate the model
library("DAAG")


model_cv <- cv.lm(PC5df,model1,m=5)

summary(model_cv)
Analysis of Variance Table

Response: V6
          Df  Sum Sq Mean Sq F value  Pr(>F)
PC1        1 1177568 1177568   19.78 6.5e-05 ***
PC2        1  633037  633037   10.63  0.0022 **
PC3        1   58541   58541    0.98  0.3272
PC4        1  257832  257832    4.33  0.0437 *
PC5        1 2312556 2312556   38.84 2.0e-07 ***
Residuals 41 2441394   59546
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
ms
72199
```

```
> SSres_cv <- attr(model_cv,"ms")*nrow(uscrime)
> totalSSEc <- sum((uscrime$Crime - mean(uscrime$Crime))^2)
> RScv1<- 1 - SSres_cv/totalSSEc
> RScv1
[1] 0.507
> RScv1 - (1 - RScv1)*5/(nrow(uscrime)-5-1)###adjusted R square
[1] 0.447
```

The R-squared value for the cross validated model with first five PCs is 0.507 and the Adjusted R-squared value is 0.447.

```
### lets trey to calculate R2 with the number of top principle components
R2 <- numeric(15) # create a vector to store the R-squared values
> for (i in 1:15) {
+   pclist <- uscrime.pca$x[,1:i]  # use the first i prinicipal components
+   pcc <- cbind(uscrime[,16],pclist)  # create data set
+   model <- lm(V1~.,data = as.data.frame(pcc)) # fit model
+   R2[i] <- 1 - sum(model$residuals^2)/sum((uscrime$Crime - mean(uscrime$Crime))^2) #
calculate R-squared
+ }

> R2
 [1] 0.171 0.263 0.272 0.309 0.645 0.659 0.688 0.690 0.692 0.696 0.697 0.769
[13] 0.772 0.791 0.803
```

```
### lets trey to calculate cross validated R2 with the number of top principle components

library(DAAG)
r2cv <- numeric(15)
for (i in 1:15) {
  pclist <- uscrime.pca$x[,1:i]  # use the first i prinicipal components
  pcc <- cbind(uscrime[,16],pclist)  # create data set
  model <- lm(V1~.,data = as.data.frame(pcc)) # fit model
  c <- cv.lm(as.data.frame(pcc),model,m=5) # cross-validate
  r2cv[i] <- 1 - attr(c,"ms")*nrow(uscrime)/sum((uscrime$Crime - mean(uscrime$Crime))^2) #
calculate R-squared
}
```

```
> r2cv
 [1] 0.0711 0.1228 0.0963 0.0392 0.5068 0.5218 0.5306 0.4706 0.4299 0.4085
[11] 0.2768 0.3808 0.3461 0.4172 0.4198
```

Model and PC numbers in model	R -squared in training data	Cross Validated R squared
1	0.171	0.0711
2	0.263	0.1228
3	0.272	0.0963
4	0.309	0.0392
5	0.645	0.5068
6	0.659	0.5218
7	0.688	0.5306
8	0.69	0.4706
9	0.692	0.4299
10	0.696	0.4085
11	0.697	0.2768
12	0.769	0.3808
13	0.772	0.3461
14	0.791	0.4172
15	0.803	0.4198

Analyzing the above table we can see the R-squared value and cross validated value of the model. The model still looks overfitted. If planning to build a model using top Principal components. If using principle components 5 top PCs would give better option but the model is still overfitted.