Assignment 6

Question 9.1

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 9.1

```
library(psych)
library(ggbiplot)
library(ggplot2)
```

Model 1

```
#Getting the training and test data
crime_data<-read.table('uscrime.txt', sep = "", header = TRUE )
test<-read.table('test.txt', sep="", header = TRUE)

#Splitting the data to predictors and response
X<-crime_data[,1:15]
y<-crime_data[16]

#Applying Principle component analysis
pca<-prcomp(X, scale. = TRUE, center = TRUE)

#center and scale in the above formula refers to respective mean and standard
#deviation of the variables that are used for normalization prior to
#implementing PCA

#Understanding the results
print(pca)
```

```
## Standard deviations (1, .., p=15):
##  [1] 2.45335539 1.67387187 1.41596057 1.07805742 0.97892746 0.74377006
##  [7] 0.56729065 0.55443780 0.48492813 0.44708045 0.41914843 0.35803646
## [13] 0.26332811 0.24180109 0.06792764
```

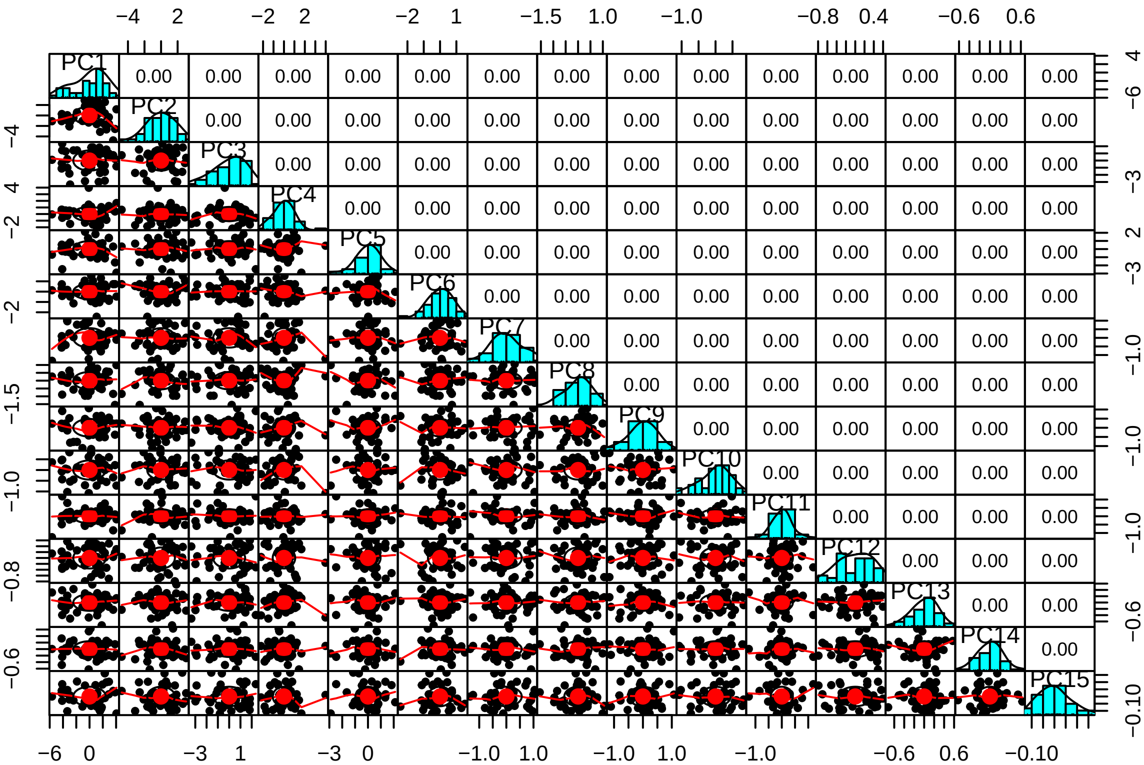
```
##
## Rotation (n x k) = (15 x 15):
##                PC1         PC2           PC3         PC4         PC5
## M      -0.30371194  0.06280357  0.1724199946 -0.02035537 -0.35832737
## So     -0.33088129 -0.15837219  0.0155433104  0.29247181 -0.12061130
## Ed      0.33962148  0.21461152  0.0677396249  0.07974375 -0.02442839
## Po1     0.30863412 -0.26981761  0.0506458161  0.33325059 -0.23527680
## Po2     0.31099285 -0.26396300  0.0530651173  0.35192809 -0.20473383
## LF      0.17617757  0.31943042  0.2715301768 -0.14326529 -0.39407588
## M.F     0.11638221  0.39434428 -0.2031621598  0.01048029 -0.57877443
## Pop     0.11307836 -0.46723456  0.0770210971 -0.03210513 -0.08317034
## NW     -0.29358647 -0.22801119  0.0788156621  0.23925971 -0.36079387
## U1      0.04050137  0.00807439 -0.6590290980 -0.18279096 -0.13136873
## U2      0.01812228 -0.27971336 -0.5785006293 -0.06889312 -0.13499487
## Wealth  0.37970331 -0.07718862  0.0100647664  0.11781752  0.01167683
## Ineq   -0.36579778 -0.02752240 -0.0002944563 -0.08066612 -0.21672823
## Prob   -0.25888661  0.15831708 -0.1176726436  0.49303389  0.16562829
## Time   -0.02062867 -0.38014836  0.2235664632 -0.54059002 -0.14764767
##                 PC6         PC7         PC8         PC9        PC10        PC11
## M      -0.449132706 -0.15707378 -0.55367691  0.15474793 -0.01443093  0.39446657
## So     -0.100500743  0.19649727  0.22734157 -0.65599872  0.06141452  0.23397868
## Ed     -0.008571367 -0.23943629 -0.14644678 -0.44326978  0.51887452 -0.11821954
## Po1    -0.095776709  0.08011735  0.04613156  0.19425472 -0.14320978 -0.13042001
## Po2    -0.119524780  0.09518288  0.03168720  0.19512072 -0.05929780 -0.13885912
## LF      0.504234275 -0.15931612  0.25513777  0.14393498  0.03077073  0.38532827
## M.F    -0.074501901  0.15548197 -0.05507254 -0.24378252 -0.35323357 -0.28029732
## Pop     0.547098563  0.09046187 -0.59078221 -0.20244830 -0.03970718  0.05849643
## NW      0.051219538 -0.31154195  0.20432828  0.18984178  0.49201966 -0.20695666
## U1      0.017385981 -0.17354115 -0.20206312  0.02069349  0.22765278 -0.17857891
## U2      0.048155286 -0.07526787  0.24369650  0.05576010 -0.04750100  0.47021842
## Wealth -0.154683104 -0.14859424  0.08630649 -0.23196695 -0.11219383  0.31955631
## Ineq    0.272027031  0.37483032  0.07184018 -0.02494384 -0.01390576 -0.18278697
## Prob    0.283535996 -0.56159383 -0.08598908 -0.05306898 -0.42530006 -0.08978385
## Time   -0.148203050 -0.44199877  0.19507812 -0.23551363 -0.29264326 -0.26363121
##               PC12        PC13        PC14          PC15
## M       0.16580189 -0.05142365  0.04901705  0.0051398012
## So     -0.05753357 -0.29368483 -0.29364512  0.0084369230
## Ed      0.47786536  0.19441949  0.03964277 -0.0280052040
## Po1     0.22611207 -0.18592255 -0.09490151 -0.6894155129
## Po2     0.19088461 -0.13454940 -0.08259642  0.7200270100
## LF      0.02705134 -0.27742957 -0.15385625  0.0336823193
## M.F    -0.23925913  0.31624667 -0.04125321  0.0097922075
## Pop    -0.18350385  0.12651689 -0.05326383  0.0001496323
## NW     -0.36671707  0.22901695  0.13227774 -0.0370783671
## U1     -0.09314897 -0.59039450 -0.02335942  0.0111359325
## U2      0.28440496  0.43292853 -0.03985736  0.0073618948
## Wealth -0.32172821 -0.14077972  0.70031840 -0.0025685109
## Ineq    0.43762828 -0.12181090  0.59279037  0.0177570357
## Prob    0.15567100 -0.03547596  0.04761011  0.0293376260
## Time    0.13536989 -0.05738113 -0.04488401  0.0376754405
```

```
#Each PC is a normalized linear combinations of original variables
#Rotation or loading are the coefficients of the liner combinations of the
#continuous variables. These value lie between 1 and -1 and show the degree
```

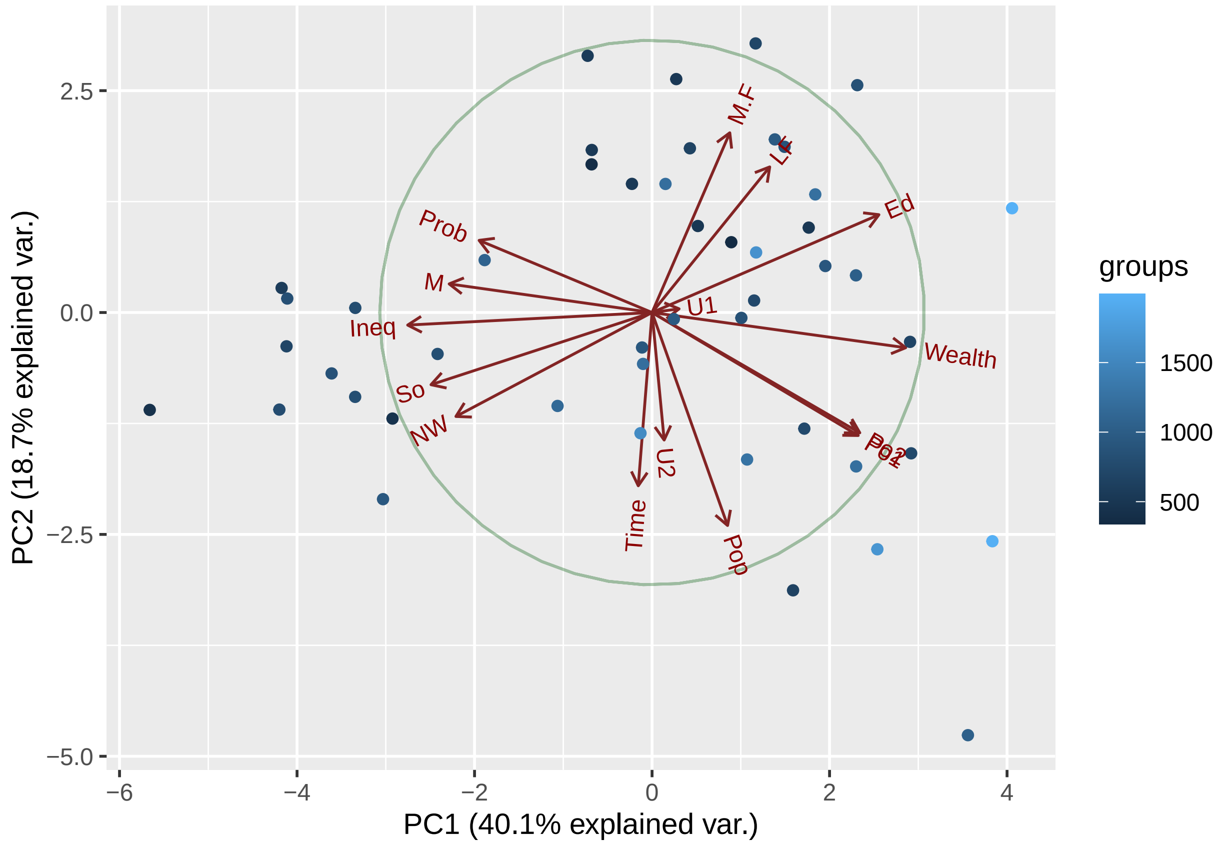
```r
#of correlation with the principle component

summary(pca)
```

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

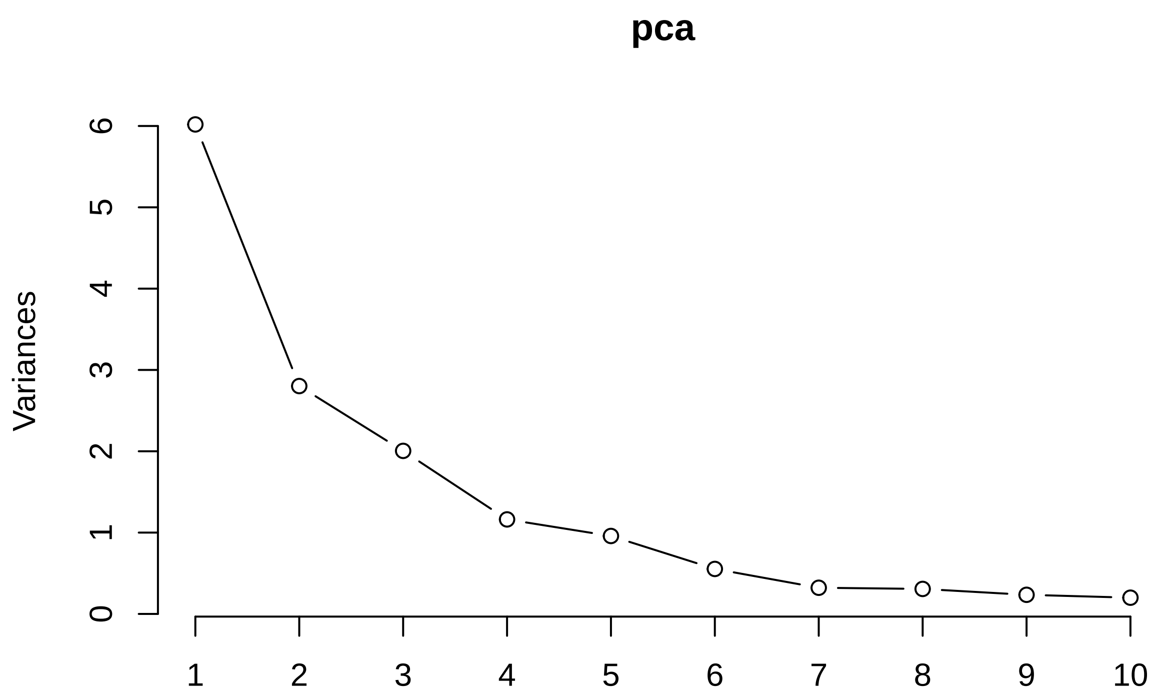
```r
#The proportion of variance in the summary shows the how well the PC can explain
#the variability in the data. Here we can see PC1 alone explains 40% of the
#variability in the data followed by PC2 19% and PC3 13%. If we need to cover
#upto 90% of the variability in the data we need to consider PC upto 7

#Plotting the Principle components

pairs.panels(pca$x, gap=0)
```



```
#This plot shows the orthogonality of the principle components as you can see
#there is no correlation between the PC's and thus we can say we have removed
#the multi-collinearity issue which need to handeld before doing regression.

#Visualizing principle components using Bi-Plots
ggbiplot(pca,
         obs.scale=1,
         var.axes=TRUE,
         var.scale = 1,
         groups = crime_data$Crime,
         circle = TRUE,
         )
```




```
#Understanding this plot. Closer the vectors more the correlation between them.
#The plot shows vectors such as wealth ,Ed, PO2, Po1 have positive correlation
#with PC1 as they are on the right side of the 0 mark. In other words the vectors
#on the right side of the 0 mark on the PC1 axis have positive contribution on PC1

#Selecting number of PC using the variance plot
screeplot(pca, type = "l")
```




```r
#Now building a regression model using the principle components.
#We will use the first 4 PC using the plot above as they encapsulate 80 of the
#variability in the data

#Creating a new data with the PC
new_data<-as.data.frame(cbind(pca$x[,1:4], crime_data$Crime))

#Doing a Linear regression on our new model have 4 principle components
new_model<-lm(V5~.,data = new_data)
summary(new_model)
```

```
##
## Call:
## lm(formula = V5 ~ ., data = new_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -557.76 -210.91  -29.08  197.26  810.35
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      49.07  18.443  < 2e-16 ***
## PC1            65.22      20.22   3.225  0.00244 **
## PC2           -70.08      29.63  -2.365  0.02273 *
## PC3            25.19      35.03   0.719  0.47602
## PC4            69.45      46.01   1.509  0.13872
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 336.4 on 42 degrees of freedom
## Multiple R-squared:  0.3091, Adjusted R-squared:  0.2433
## F-statistic: 4.698 on 4 and 42 DF,  p-value: 0.003178
```

```
#the new model shows a low R2 and adjusted R2 value as compared to the linear
#regression model done in Assignment 1

#Finding the model coefficients in terms of the original variables
coeff <- pca$rotation[,1:4]%*%new_model$coefficients[-1]

#Converting standardized coefficient and intercept back to original variables
s<-sapply(crime_data[,1:15], sd) #SD of each variable in the original dataset
m<-sapply(crime_data[,1:15], mean)#Mean of each variable in the original dataset
intercept<-new_model$coefficients[1]

coeff_new<-coeff/s
intercept_new<-intercept-sum(coeff*m/s)
print(coeff_new)
```

```
##                [,1]
## M        -16.9307630
## So        21.3436771
## Ed        12.8297238
## Po1       21.3521593
## Po2       23.0883154
## LF      -346.5657125
## M.F       -8.2930969
## Pop        1.0462155
## NW         1.5009941
## U1     -1509.9345216
## U2         1.6883674
## Wealth     0.0400119
## Ineq      -6.9020218
## Prob     144.9492678
## Time      -0.9330765
```

```
print(intercept_new)
```

```
## (Intercept)
##    1666.485
```

```
res<-as.matrix(X)%*%coeff_new+intercept_new #Prediction on the training data
pre<-as.matrix(test)%*%coeff_new+intercept_new #Prediction on the test data from
                                                #assignment 5
print(pre)
```

```
##          [,1]
## [1,] 1112.678
```

```r
#Calculating R2
r2<-1-sum((res-crime_data$Crime)^2)/sum((crime_data$Crime-mean(crime_data$Crime))^2)
print(r2)
```

```
## [1] 0.3091121
```

```r
#Comparison with the previous assignment

#Linear regression model
model1<-lm(Crime~.,data = crime_data)

summary(model1) # Model summary
```

```
##
## Call:
## lm(formula = Crime ~ ., data = crime_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -395.74  -98.09   -6.69  112.99  512.67
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.984e+03  1.628e+03  -3.675 0.000893 ***
## M            8.783e+01  4.171e+01   2.106 0.043443 *
## So          -3.803e+00  1.488e+02  -0.026 0.979765
## Ed           1.883e+02  6.209e+01   3.033 0.004861 **
## Po1          1.928e+02  1.061e+02   1.817 0.078892 .
## Po2         -1.094e+02  1.175e+02  -0.931 0.358830
## LF          -6.638e+02  1.470e+03  -0.452 0.654654
## M.F          1.741e+01  2.035e+01   0.855 0.398995
## Pop         -7.330e-01  1.290e+00  -0.568 0.573845
## NW           4.204e+00  6.481e+00   0.649 0.521279
## U1          -5.827e+03  4.210e+03  -1.384 0.176238
## U2           1.678e+02  8.234e+01   2.038 0.050161 .
## Wealth       9.617e-02  1.037e-01   0.928 0.360754
## Ineq         7.067e+01  2.272e+01   3.111 0.003983 **
## Prob        -4.855e+03  2.272e+03  -2.137 0.040627 *
## Time        -3.479e+00  7.165e+00  -0.486 0.630708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 209.1 on 31 degrees of freedom
## Multiple R-squared:  0.8031, Adjusted R-squared:  0.7078
## F-statistic: 8.429 on 15 and 31 DF,  p-value: 3.539e-07
```

```r
pred<-predict(model1, test) #Prediction on test data
```

Conclusions: Model 1

In the above exercise we applied PCA to our data and then used the first four principle components were used to run a linear regression model. Clearly, the PCA model performed worse then the ordinary linear regression model which can be seen by the low R2 values. Note, that for the above model the principle components only contained 80% of the variance. Thus, to test further we can add more principle components and see the results.

Model 2

Models with 7 principle components covering 92% variance

```
#Creating a new data with the PC
new_data_7p<-as.data.frame(cbind(pca$x[,1:7], crime_data$Crime))

#Doing a Linear regression on our new model have 7 principle components
new_model_7p<-lm(V8~.,data = new_data_7p)
summary(new_model_7p)
```

```
##
## Call:
## lm(formula = V8 ~ ., data = new_data_7p)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -475.41 -141.65   34.73  137.25  412.32
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      34.21  26.454  < 2e-16 ***
## PC1            65.22      14.10   4.626 4.04e-05 ***
## PC2           -70.08      20.66  -3.392   0.0016 **
## PC3            25.19      24.42   1.032   0.3086
## PC4            69.45      32.08   2.165   0.0366 *
## PC5          -229.04      35.33  -6.483 1.11e-07 ***
## PC6           -60.21      46.50  -1.295   0.2029
## PC7           117.26      60.96   1.923   0.0617 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 234.6 on 39 degrees of freedom
## Multiple R-squared:  0.6882, Adjusted R-squared:  0.6322
## F-statistic:  12.3 on 7 and 39 DF,  p-value: 3.513e-08
```

```
#the new model shows a low R2 and adjusted R2 value as compared to the linear
#regression model done in Assignment 1

#Finding the model coefficients in terms of the original variables
coeff_7p <- pca$rotation[,1:7]%*%new_model_7p$coefficients[-1]

#Converting standardized coefficient and intercept back to original variables
s<-sapply(crime_data[,1:15], sd) #SD of each variable in the original dataset
m<-sapply(crime_data[,1:15], mean)#Mean of each variable in the original dataset
intercept_7p<-new_model_7p$coefficients[1]
```

```r
coeff_new_7p<-coeff_7p/s
intercept_new<-intercept_7p-sum(coeff_7p*m/s)
print(coeff_new)
```

```
##                [,1]
## M        -16.9307630
## So        21.3436771
## Ed        12.8297238
## Po1       21.3521593
## Po2       23.0883154
## LF      -346.5657125
## M.F       -8.2930969
## Pop        1.0462155
## NW         1.5009941
## U1     -1509.9345216
## U2         1.6883674
## Wealth     0.0400119
## Ineq      -6.9020218
## Prob     144.9492678
## Time      -0.9330765
```

```r
print(intercept_new)
```

```
## (Intercept)
##   -5498.458
```

```r
res_7p<-as.matrix(X)%*%coeff_new_7p+intercept_new #Prediction on the training data
pre_7p<-as.matrix(test)%*%coeff_new_7p+intercept_new #Prediction on the test data from
                                                  #assignment 5
print(pre_7p)
```

```
##          [,1]
## [1,] 1230.418
```

```r
#Calculating R2
r2<-1-sum((res_7p-crime_data$Crime)^2)/sum((crime_data$Crime-mean(crime_data$Crime))^2)
print(r2)
```

```
## [1] 0.6881819
```

Conclusion: Model 2

We can see that even after adding 7 principle components the the model quality did not improve. We can continue adding more PC but it defies one of the main purpose of reducing dimensions. Thus in this case PCA was not very helpful and we are better off using the ordinary linear regression model.