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)!)

The following contains my code, approach, and explanations to this problem:

```
data <- read.table("C:\\Users\\User\\OneDrive\\Desktop\\Data 9.1\\uscrime.txt
", stringsAsFactors = FALSE, header = TRUE)

#Use PCA and store it
data_PCA <- prcomp(data[,1:15], scale. = TRUE)
data_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  0.394
46657
## So     -0.100500743  0.19649727  0.22734157 -0.65599872  0.06141452  0.233
97868
## Ed     -0.008571367 -0.23943629 -0.14644678 -0.44326978  0.51887452 -0.118
21954
## Po1    -0.095776709  0.08011735  0.04613156  0.19425472 -0.14320978 -0.130
```

```
42001
## Po2    -0.119524780  0.09518288  0.03168720  0.19512072 -0.05929780 -0.138
85912
## LF      0.504234275 -0.15931612  0.25513777  0.14393498  0.03077073  0.385
32827
## M.F    -0.074501901  0.15548197 -0.05507254 -0.24378252 -0.35323357 -0.280
29732
## Pop     0.547098563  0.09046187 -0.59078221 -0.20244830 -0.03970718  0.058
49643
## NW      0.051219538 -0.31154195  0.20432828  0.18984178  0.49201966 -0.206
95666
## U1      0.017385981 -0.17354115 -0.20206312  0.02069349  0.22765278 -0.178
57891
## U2      0.048155286 -0.07526787  0.24369650  0.05576010 -0.04750100  0.470
21842
## Wealth -0.154683104 -0.14859424  0.08630649 -0.23196695 -0.11219383  0.319
55631
## Ineq    0.272027031  0.37483032  0.07184018 -0.02494384 -0.01390576 -0.182
78697
## Prob    0.283535996 -0.56159383 -0.08598908 -0.05306898 -0.42530006 -0.089
78385
## Time   -0.148203050 -0.44199877  0.19507812 -0.23551363 -0.29264326 -0.263
63121
##              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
```

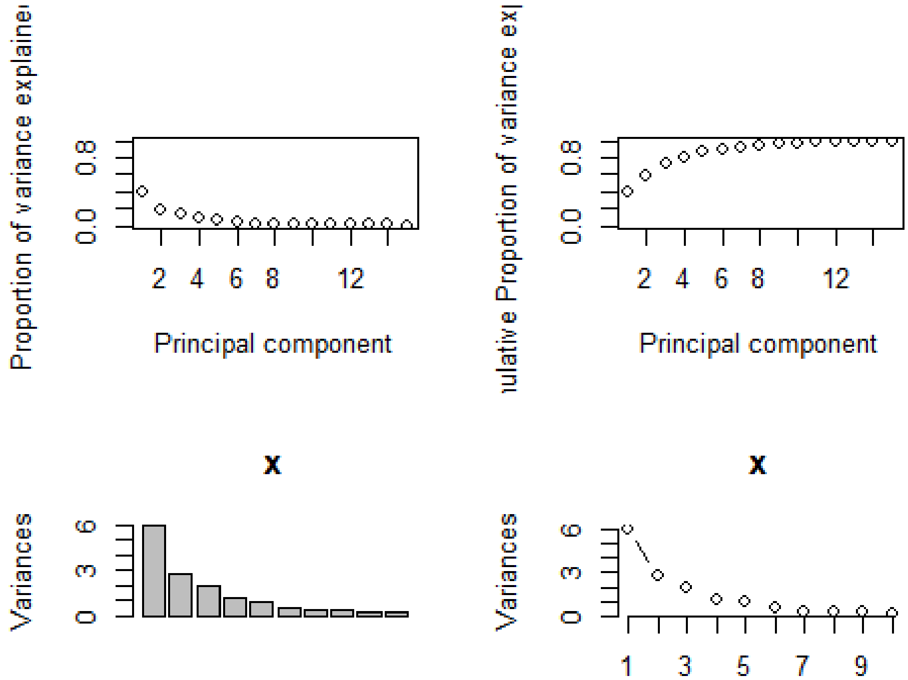
```
summary(data_PCA)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC
7
## Standard deviation     2.4534 1.6739 1.4160 1.07806 0.97893 0.74377 0.5672
9
## Proportion of Variance 0.4013 0.1868 0.1337 0.07748 0.06389 0.03688 0.0214
5
## Cumulative Proportion  0.4013 0.5880 0.7217 0.79920 0.86308 0.89996 0.9214
```

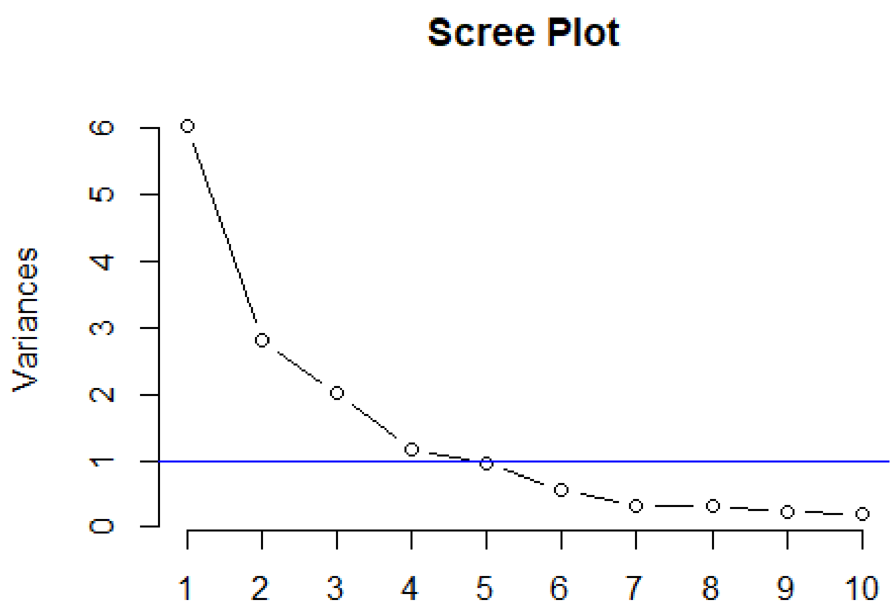
```
2
##                            PC8     PC9    PC10    PC11    PC12    PC13   P
C14
## Standard deviation     0.55444 0.48493 0.44708 0.41915 0.35804 0.26333 0.2
418
## Proportion of Variance 0.02049 0.01568 0.01333 0.01171 0.00855 0.00462 0.0
039
## Cumulative Proportion  0.94191 0.95759 0.97091 0.98263 0.99117 0.99579 0.9
997
##                           PC15
## Standard deviation     0.06793
## Proportion of Variance 0.00031
## Cumulative Proportion  1.00000
```

The summary of this data orders the principal components by their overall proportion of total variance. Components one to three appear to have the greatest proportion of overall variance. However, I want to more concretely identify the number of components to use. For this, I will generate a few plots, including a scree plot, and examine it.

```
#Used code from PCA lesson on https://rpubs.com/JanpuHou/278584
#The following code will generate four plots
pcaPlots <- function(x) {
    x.var <- x$sdev ^ 2
    x.pvar <- x.var/sum(x.var)
    print("Proportions of variance:")
    print(x.pvar)

    par(mfrow=c(2,2))
    plot(x.pvar,xlab="Principal component", ylab="Proportion of variance expl
ained", ylim=c(0,1), type='b')
    plot(cumsum(x.pvar),xlab="Principal component", ylab="Cumulative Proporti
on of variance explained", ylim=c(0,1), type='b')
    screeplot(x)
    screeplot(x,type="l")
    par(mfrow=c(1,1))
}

#Generate PCA plots and a scree plot
pcaPlots(data_PCA)

## [1] "Proportions of variance:"
##  [1] 0.401263510 0.186789802 0.133662956 0.077480520 0.063886598 0.0368795
93
##  [7] 0.021454579 0.020493418 0.015677019 0.013325395 0.011712360 0.0085460
07
## [13] 0.004622779 0.003897851 0.000307611
```




```
screeplot(data_PCA, main="Scree Plot", type="line")
abline(h=1, col="blue")
```



Using a threshold of one, the scree plot seems to indicate that the first five principal components are the most important. Looking at the other generated plots, it appears that the elbow point is at 5 principal components as well. Therefore, using all of these observations, I will use pc = five for my model.

```
PC_chosen <- 5
crimePCA <- cbind(data_PCA$x[,1:PC_chosen], data[,16])
crimePCA
```

```
##              PC1         PC2         PC3         PC4          PC5
##  [1,] -4.1992835 -1.09383120 -1.11907395  0.67178115  0.055283376  791
##  [2,]  1.1726630  0.67701360 -0.05244634 -0.08350709 -1.173199821 1635
##  [3,] -4.1737248  0.27677501 -0.37107658  0.37793995  0.541345246  578
##  [4,]  3.8349617 -2.57690596  0.22793998  0.38262331 -1.644746496 1969
##  [5,]  1.8392999  1.33098564  1.27882805  0.71814305  0.041590320 1234
##  [6,]  2.9072336 -0.33054213  0.53288181  1.22140635  1.374360960  682
##  [7,]  0.2457752 -0.07362562 -0.90742064  1.13685873  0.718644387  963
##  [8,] -0.1301330 -1.35985577  0.59753132  1.44045387 -0.222781388 1555
##  [9,] -3.6103169 -0.68621008  1.28372246  0.55171150 -0.324292990  856
## [10,]  1.1672376  3.03207033  0.37984502 -0.28887026 -0.646056610  705
## [11,]  2.5384879 -2.66771358  1.54424656 -0.87671210 -0.324083561 1674
## [12,]  1.0065920 -0.06044849  1.18861346 -1.31261964  0.358087724  849
## [13,]  0.5161143  0.97485189  1.83351610 -1.59117618  0.599881946  511
## [14,]  0.4265556  1.85044812  1.02893477 -0.07789173  0.741887592  664
## [15,] -3.3435299  0.05182823 -1.01358113  0.08840211  0.002969448  798
## [16,] -3.0310689 -2.10295524 -1.82993161  0.52347187 -0.387454246  946
## [17,] -0.2262961  1.44939774 -1.37565975  0.28960865  1.337784608  539
## [18,] -0.1127499 -0.39407030 -0.38836278  3.97985093  0.410914404  929
## [19,]  2.9195668 -1.58646124  0.97612613  0.78629766  1.356288600  750
## [20,]  2.2998485 -1.73396487 -2.82423222 -0.23281758 -0.653038858 1225
## [21,]  1.1501667  0.13531015  0.28506743 -2.19770548  0.084621572  742
## [22,] -5.6594827 -1.09730404  0.10043541 -0.05245484 -0.689327990  439
## [23,] -0.1011749 -0.57911362  0.71128354 -0.44394773  0.689939865 1216
## [24,]  1.3836281  1.95052341 -2.98485490 -0.35942784 -0.744371276  968
## [25,]  0.2727756  2.63013778  1.83189535  0.05207518  0.803692524  523
## [26,]  4.0565577  1.17534729 -0.81690756  1.66990720 -2.895110075 1993
## [27,]  0.8929694  0.79236692  1.26822542 -0.57575615  1.830793964  342
## [28,]  0.1514495  1.44873320  0.10857670 -0.51040146 -1.023229895 1216
## [29,]  3.5592481 -4.76202163  0.75080576  0.64692974  0.309946510 1043
## [30,] -4.1184576 -0.38073981  1.43463965  0.63330834 -0.254715638  696
## [31,] -0.6811731  1.66926027 -2.88645794 -1.30977099 -0.470913997  373
## [32,]  1.7157269 -1.30836339 -0.55971313 -0.70557980  0.331277622  754
## [33,] -1.8860627  0.59058174  1.43570145  0.18239089  0.291863659 1072
## [34,]  1.9526349  0.52395429 -0.75642216  0.44289927  0.723474420  923
## [35,]  1.5888864 -3.12998571 -1.73107199 -1.68604766  0.665406182  653
## [36,]  1.0709414 -1.65628271  0.79436888 -1.85172698  0.020031154 1272
## [37,] -4.1101715  0.15766712  2.36296974 -0.56868399 -2.469679496  831
## [38,] -0.7254706  2.89263339 -0.36348376 -0.50612576  0.028157162  566
## [39,] -3.3451254 -0.95045293  0.19551398 -0.27716645  0.487259213  826
## [40,] -1.0644466 -1.05265304  0.82886286 -0.12042931 -0.645884788 1151
```

```
## [41,]  1.4933989  1.86712106  1.81853582 -1.06112429  0.009855774  880
## [42,] -0.6789284  1.83156328 -1.65435992  0.95121379  2.115630145  542
## [43,] -2.4164258 -0.46701087  1.42808323  0.41149015 -0.867397522  823
## [44,]  2.2978729  0.41865689 -0.64422929 -0.63462770 -0.703116983 1030
## [45,] -2.9245282 -1.19488555 -3.35139309 -1.48966984  0.806659622  455
## [46,]  1.7654525  0.95655926  0.98576138  1.05683769  0.542466034  508
## [47,]  2.3125056  2.56161119 -1.58223354  0.59863946 -1.140712406  849
```

```
#Create lm model
lm_model <- lm(V6~., data = as.data.frame(crimePCA))
summary(lm_model)
```

```
##
## Call:
## lm(formula = V6 ~ ., data = as.data.frame(crimePCA))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -420.79 -185.01   12.21  146.24  447.86
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      35.59  25.428  < 2e-16 ***
## PC1            65.22      14.67   4.447 6.51e-05 ***
## PC2           -70.08      21.49  -3.261  0.00224 **
## PC3            25.19      25.41   0.992  0.32725
## PC4            69.45      33.37   2.081  0.04374 *
## PC5          -229.04      36.75  -6.232 2.02e-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:  0.6452, Adjusted R-squared:  0.6019
## F-statistic: 14.91 on 5 and 41 DF,  p-value: 2.446e-08
```

Here, I generated a linear regression model using the number of principal components chosen before and the original crime data.

```
#Transformation steps and estimation
intercept <- lm_model$coefficients[1]
beta_vec <- lm_model$coefficients[2:(PC_chosen+1)]
alpha_vec <- data_PCA$rotation[,1:PC_chosen] %*% beta_vec

mu <- sapply(data[,1:15], mean)
sigma <- sapply(data[,1:15], sd)

og_alpha <- alpha_vec/sigma
og_beta <- intercept - sum(alpha_vec*mu/sigma)

estimate <- as.matrix(data[,1:15]) %*% og_alpha + og_beta
```

```r
#Calculate R^2 metrics
SSE <- sum((estimate-data[,16])^2)
SStot <- sum((data[,16] - mean(data[,16]))^2)

R2 <- 1- (SSE/SStot)
R2

## [1] 0.6451941

adj_R2 <- R2 - (1-R2)*PC_chosen/(nrow(data) - PC_chosen - 1)
adj_R2

## [1] 0.601925
```

Here, I found the intercept and created the alpha and beta vectors. I also obtained the original alpha and beta values using the calculated values for mu and sigma. Following this, I made some estimations. As can be seen, the form of the “estimate” resembles the equation of a line (y = aX + b where a = og_alpha and b = og_beta). The estimates were then used to calculate the R^2 and adjusted R^2 values. As a quick note, in regards to the R^2 values, it can be seen that they are lower than the values I obtained for last week’s homework (for the model using only select predictors): R-squared: 0.7659, Adjusted R-squared: 0.7307. As an additional note, these R^2 values are also the same as the R^2 values listed in the output of ‘summary(lm_model)’ shown above.

```r
#Test data from last week
test <- 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)

test_pred <- data.frame(predict(data_PCA, test))
test_pred_model <- predict(lm_model, test_pred)
test_pred_model

##        1
## 1388.926
```

Here, I used the test data given last week to see how the new linear regression model fares compared to the one from last week. The predicted value of crime I obtained for last week’s homework was 1304 while the value I obtained with the new model is 1389. So we can see that the two values are similar; however, when we flatly compare the R^2 values, I would say that this new model using PCA (R^2 = 0.6451941; adj R^2 = 0.601925) is worse than the previous week’s model (R^2 = 0.7659; adj R^2 = 0.7307). In favour of the PCA model though, it did obtain a similar prediction with less predictors, so it shows that there is quite a bit of merit to it. I would be curious to see how the results change with a larger data set as well.

Finally, my specified model using the original alpha (obtained by doing t(og_alpha)) and beta (obtained by printing og_beta) values for the first five principal components is:

Crime ~ 48.37374M + 79.01922So + 17.8312Ed + 39.48484Po1 + 39.85892Po2 + 1886.946LF + 36.69366M.F + 1.546583Pop + 9.537384NW + 159.0115U1 + 38.29933U2 + 0.03724014Wealth + 5.540321Ineq - 1523.521Prob + 3.838779Time - 5933.837