PCA analysis on crime data

```r
crime <- read.delim("http://www.statsci.org/data/general/uscrime.txt")
data_scaled <- as.data.frame(scale(crime))
```

We are doing principal component analysis is reducing dimensionality of large data sets into smaller ones so we can create a simpler model. We are trading a little accuracy for simplicity as smaller data sets are easier to explore and visualize. Machine learning algos can also run faster on smaller data sets.

I'm doing a PCA on the crime data set.

```r
head(crime)
```

```
##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.083401
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.015801
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.041399
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.034201
##      Time Crime
## 1 26.2011   791
## 2 25.2999  1635
## 3 24.3006   578
## 4 29.9012  1969
## 5 21.2998  1234
## 6 20.9995   682
```

```r
PCA <- prcomp(crime[,1:15], scale=TRUE)
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
```

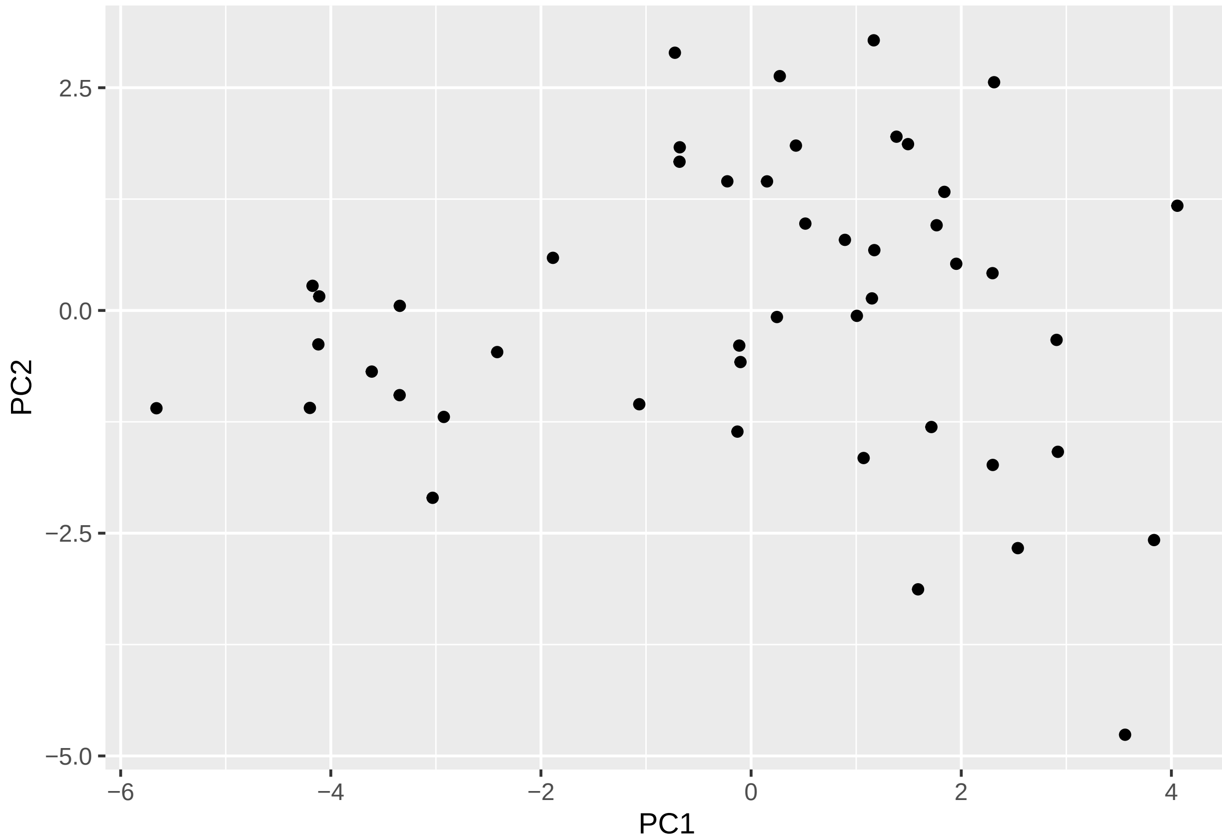
```
## U1      0.04050137  0.00807439 -0.6590290980 -0.18279096 -0.13136873
## U2      0.01812228 -0.27971336 -0.5785006293 -0.06889312 -0.13499487
## Wealth  0.37970331 -0.07718862  0.0100647664  0.11781752  0.01167683
## Ineq   -0.36579778 -0.02752240 -0.0002944563 -0.08066612 -0.21672823
## Prob   -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
```

```r
names(PCA)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

Above are the eigenvector for the principal components from the PCA analysis.

```r
ggplot(as.data.frame(PCA$x), aes(x = PC1, y = PC2)) + geom_point()
```




```
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
```

You can see that the first component accounts for 40% of variance as shown by the proportion of variance value. Rest of components have less importance as you can see from proportion of variance. We can conclude that this data set is governed by crime variable.

```
#Building linear regression model with first 5 terms
PC <- PCA$x[,1:5]
crimePC <- cbind(PC,crime[,16])
modelPCA <- lm(V6~., data = as.data.frame(crimePC))
summary(modelPCA)
```

```
##
## Call:
## lm(formula = V6 ~ ., data = as.data.frame(crimePC))
##
## 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
```

Looking at the linear regression model, all terms except PC3 have a p value below 0.05. This shows 4 of the 5 terms are relevant factors. The adjusted R squared value is really low at 0.6.

Below we are reconstructing model in terms of original variables.

```r
#b0 is our intercept
b0 <- modelPCA$coefficients[1]

#beta vector is created using the coefficients
betas <- modelPCA$coefficients[2:6]

b0
```

```
## (Intercept)
##    905.0851
```

```r
#alpha vector is calculated below
alphas <- PCA$rotation[,1:5]%*%betas

#unscaling data below for beta and alpha and then calculating estimates
unscaledalpha <- alphas/sapply(crime[,1:15],sd)
betaunscaled <- b0 - sum(alphas*sapply(crime[,1:15],mean)/sapply(crime[,1:15],sd))

 est <- as.matrix(crime[,1:15]) %*% unscaledalpha + betaunscaled
 est
```

```
##            [,1]
##  [1,]  713.6803
##  [2,] 1195.7066
##  [3,]  506.4008
```

```
##  [4,] 1744.8151
##  [5,] 1004.3223
##  [6,]  901.3083
##  [7,]  817.7618
##  [8,] 1158.0158
##  [9,]  862.6600
## [10,]  906.1942
## [11,] 1309.8473
## [12,]  831.7397
## [13,]  668.7175
## [14,]  653.8079
## [15,]  663.3242
## [16,]  933.7860
## [17,]  467.7924
## [18,] 1097.8331
## [19,]  975.2212
## [20,] 1238.8452
## [21,]  805.7895
## [22,]  769.6724
## [23,]  768.1369
## [24,]  928.9523
## [25,]  604.2355
## [26,] 1845.7567
## [27,]  480.4270
## [28,] 1015.0839
## [29,] 1463.7936
## [30,]  801.6455
## [31,]  687.8542
## [32,]  969.6941
## [33,]  722.6822
## [34,]  841.7013
## [35,]  914.9564
## [36,]  977.8353
## [37,] 1211.6890
## [38,]  604.2928
## [39,]  627.6148
## [40,] 1069.8938
## [41,]  841.4929
## [42,]  272.2545
## [43,] 1043.4520
## [44,] 1126.3430
## [45,]  425.4541
## [46,]  927.1627
## [47,] 1139.3538
```

```
SSE = sum((est - crime[,16])^2)
SStot = sum((crime[,16] - mean(crime[,16]))^2)
R2 <- 1-SSE/SStot


R2adjust <- R2 - (1-R2)*5/(nrow(crime)-5-1)
R2adjust
```

```
## [1] 0.601925
```

Looking at calculated R2 adjusted values you can see that the value is pretty decent at 0.6 which shows the model fit well but not as well as the linear regression from 8.2 which was higher at 0.7.

```
a <- 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)

pred_a <- data.frame(predict(PCA,a))

pred_model<-predict(modelPCA,pred_a)

pred_model
```

```
##        1
## 1388.926
```

This prediction is within the range of the data set so it is a reasonable data point.