HW 6

Question 8.2 Summary From HW 5

Question 8.2 was using linear regression to estimate the crime rate of a test data point. First was running a linear regression all the data points. From the model I picked 4 variables that carried the most significance. Then used a test data point and found how well that model fit the test data point. I got a R Squared value of 0.902. #I picked the 4 variables with the lowest P Value. The closer P VALUE is to 0 the more significant the coefficient is. I used 0.05 as the cutoff.The 4 variables with the highest significance are M, Ed, Ineq, Prob

Question 9.1

Using PCA to predict the crime rate of the same test data point from HW 5.

First part of the code is clearing any global data, importing the data, calling on the required packages.

```
remove(list=ls())
set.seed(15)
dir()
```

```
##  [1] "credit_card_data.csv"          "credit_card_data.txt"
##  [3] "credit_card_with_headers.csv"  "credit_card_with_headers.txt"
##  [5] "credit_card_with_headers.xlsx" "HW 2.Rmd"
##  [7] "HW 3.R"                        "HW 3.Rmd"
##  [9] "HW 4.R"                        "HW 4.Rmd"
## [11] "HW 5.R"                        "HW 5.Rmd"
## [13] "HW 6.R"                        "HW 6.Rmd"
## [15] "HW_2.docx"                     "HW_3.docx"
## [17] "HW_4.docx"                     "HW_5.docx"
## [19] "HW_6.Rmd"                      "Iris.csv"
## [21] "Iris.txt"                      "Question 6.2.xlsx"
## [23] "Temps.csv"                     "Temps.txt"
## [25] "Temps.xlsx"                    "US_Crime.csv"
## [27] "US_Crime.txt"
```

```
library(ggplot2)
library(GGally)
```

```
crime <- read.table("US_Crime.txt", header=TRUE, stringsAsFactors = FALSE,fil
eEncoding = "UTF-8-BOM")
head(crime)
```

```
##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7
```

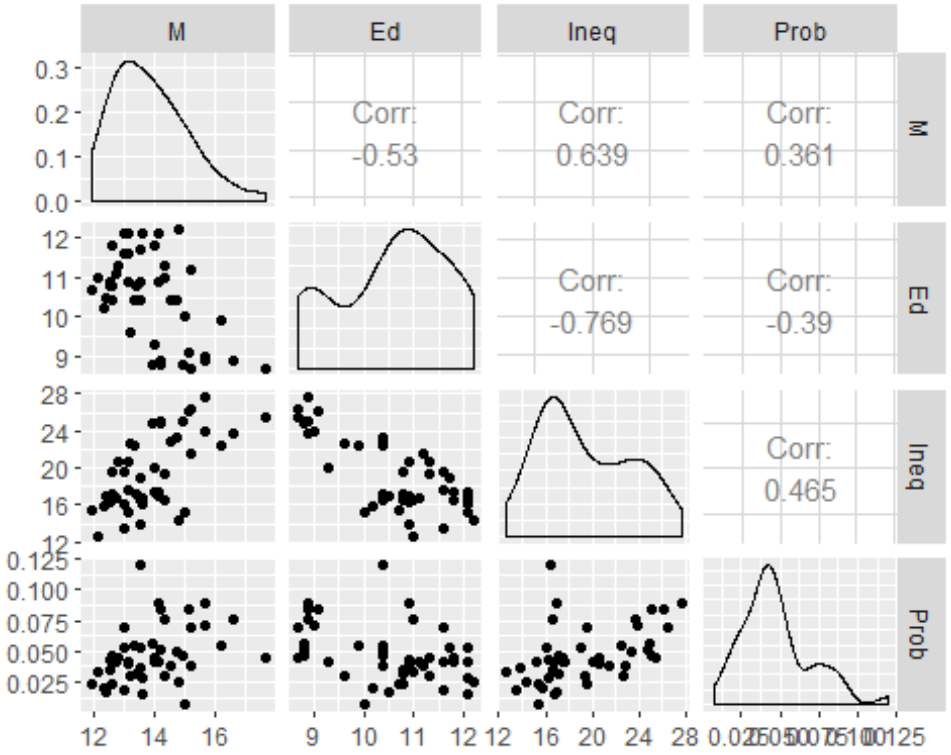
```
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6
##       Prob    Time Crime
## 1 0.084602 26.2011   791
## 2 0.029599 25.2999  1635
## 3 0.083401 24.3006   578
## 4 0.015801 29.9012  1969
## 5 0.041399 21.2998  1234
## 6 0.034201 20.9995   682
```

```
str(crime)
```

```
## 'data.frame':    47 obs. of  16 variables:
##  $ M     : num  15.1 14.3 14.2 13.6 14.1 12.1 12.7 13.1 15.7 14 ...
##  $ So    : int  1 0 1 0 0 0 1 1 1 0 ...
##  $ Ed    : num  9.1 11.3 8.9 12.1 12.1 11 11.1 10.9 9 11.8 ...
##  $ Po1   : num  5.8 10.3 4.5 14.9 10.9 11.8 8.2 11.5 6.5 7.1 ...
##  $ Po2   : num  5.6 9.5 4.4 14.1 10.1 11.5 7.9 10.9 6.2 6.8 ...
##  $ LF    : num  0.51 0.583 0.533 0.577 0.591 0.547 0.519 0.542 0.553 0.632
...
##  $ M.F   : num  95 101.2 96.9 99.4 98.5 ...
##  $ Pop   : int  33 13 18 157 18 25 4 50 39 7 ...
##  $ NW    : num  30.1 10.2 21.9 8 3 4.4 13.9 17.9 28.6 1.5 ...
##  $ U1    : num  0.108 0.096 0.094 0.102 0.091 0.084 0.097 0.079 0.081 0.1
...
##  $ U2    : num  4.1 3.6 3.3 3.9 2 2.9 3.8 3.5 2.8 2.4 ...
##  $ Wealth: int  3940 5570 3180 6730 5780 6890 6200 4720 4210 5260 ...
##  $ Ineq  : num  26.1 19.4 25 16.7 17.4 12.6 16.8 20.6 23.9 17.4 ...
##  $ Prob  : num  0.0846 0.0296 0.0834 0.0158 0.0414 ...
##  $ Time  : num  26.2 25.3 24.3 29.9 21.3 ...
##  $ Crime : int  791 1635 578 1969 1234 682 963 1555 856 705 ...
```

The next part of the code is seeing the correlation between the 4 most significant predictors from HW 5. M, Ed, Ineq, and Prob.

```
#Use ggpairs to see if there is correlation between the 4 most significant va
riables.
ggpairs(crime, columns=c("M", "Ed", "Ineq","Prob"))
```



You can see how each predictor compared with each other. It's plotted data and its correlation value.

Next I ran a PCA on all the predictors in the crime index.

```
#Run PCA on Crime Data for Scaled Predictors
crime_PCA <- prcomp(crime[,1:15], scale. = TRUE)
summary(crime_PCA)

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

You can see the principle componenets in order of importance of all 15 predictors from the data.

Within the PCA output you can also see standard deviation, the center, rotation, scale, and X. I'm specificaly calling to see the rotation value which is the Eigenvectors.

```
crime_PCA$rotation #See Eigenvectors of each predictor variable of each princ
iple component
```

```
##                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
## M      -0.449132706 -0.15707378 -0.55367691  0.15474793 -0.01443093
## So     -0.100500743  0.19649727  0.22734157 -0.65599872  0.06141452
## Ed     -0.008571367 -0.23943629 -0.14644678 -0.44326978  0.51887452
## Po1    -0.095776709  0.08011735  0.04613156  0.19425472 -0.14320978
## Po2    -0.119524780  0.09518288  0.03168720  0.19512072 -0.05929780
## LF      0.504234275 -0.15931612  0.25513777  0.14393498  0.03077073
## M.F    -0.074501901  0.15548197 -0.05507254 -0.24378252 -0.35323357
## Pop     0.547098563  0.09046187 -0.59078221 -0.20244830 -0.03970718
## NW      0.051219538 -0.31154195  0.20432828  0.18984178  0.49201966
## U1      0.017385981 -0.17354115 -0.20206312  0.02069349  0.22765278
## U2      0.048155286 -0.07526787  0.24369650  0.05576010 -0.04750100
## Wealth -0.154683104 -0.14859424  0.08630649 -0.23196695 -0.11219383
## Ineq    0.272027031  0.37483032  0.07184018 -0.02494384 -0.01390576
## Prob    0.283535996 -0.56159383 -0.08598908 -0.05306898 -0.42530006
## Time   -0.148203050 -0.44199877  0.19507812 -0.23551363 -0.29264326
##               PC11        PC12        PC13        PC14          PC15
## M      0.39446657  0.16580189 -0.05142365  0.04901705  0.0051398012
## So     0.23397868 -0.05753357 -0.29368483 -0.29364512  0.0084369230
## Ed    -0.11821954  0.47786536  0.19441949  0.03964277 -0.0280052040
## Po1   -0.13042001  0.22611207 -0.18592255 -0.09490151 -0.6894155129
## Po2   -0.13885912  0.19088461 -0.13454940 -0.08259642  0.7200270100
## LF     0.38532827  0.02705134 -0.27742957 -0.15385625  0.0336823193
## M.F   -0.28029732 -0.23925913  0.31624667 -0.04125321  0.0097922075
## Pop    0.05849643 -0.18350385  0.12651689 -0.05326383  0.0001496323
## NW    -0.20695666 -0.36671707  0.22901695  0.13227774 -0.0370783671
## U1    -0.17857891 -0.09314897 -0.59039450 -0.02335942  0.0111359325
## U2     0.47021842  0.28440496  0.43292853 -0.03985736  0.0073618948
## Wealth 0.31955631 -0.32172821 -0.14077972  0.70031840 -0.0025685109
## Ineq  -0.18278697  0.43762828 -0.12181090  0.59279037  0.0177570357
```

```
## Prob   -0.08978385  0.15567100 -0.03547596  0.04761011  0.0293376260
## Time   -0.26363121  0.13536989 -0.05738113 -0.04488401  0.0376754405
```

You can see the breakdown of each Principal Component based on each predictors Eigenvector.

The HW specifically asks to use the first 4 PCs. The next part of the code is running a PCA model only on those 4 PCs.

```
#Find the first 4 PCs from the Crime Data. HW asks for the first 4
PC_4 <- crime_PCA$x[,1:4]
PC_4
```

```
##              PC1         PC2         PC3         PC4
##  [1,] -4.1992835 -1.09383120 -1.11907395  0.67178115
##  [2,]  1.1726630  0.67701360 -0.05244634 -0.08350709
##  [3,] -4.1737248  0.27677501 -0.37107658  0.37793995
##  [4,]  3.8349617 -2.57690596  0.22793998  0.38262331
##  [5,]  1.8392999  1.33098564  1.27882805  0.71814305
##  [6,]  2.9072336 -0.33054213  0.53288181  1.22140635
##  [7,]  0.2457752 -0.07362562 -0.90742064  1.13685873
##  [8,] -0.1301330 -1.35985577  0.59753132  1.44045387
##  [9,] -3.6103169 -0.68621008  1.28372246  0.55171150
## [10,]  1.1672376  3.03207033  0.37984502 -0.28887026
## [11,]  2.5384879 -2.66771358  1.54424656 -0.87671210
## [12,]  1.0065920 -0.06044849  1.18861346 -1.31261964
## [13,]  0.5161143  0.97485189  1.83351610 -1.59117618
## [14,]  0.4265556  1.85044812  1.02893477 -0.07789173
## [15,] -3.3435299  0.05182823 -1.01358113  0.08840211
## [16,] -3.0310689 -2.10295524 -1.82993161  0.52347187
## [17,] -0.2262961  1.44939774 -1.37565975  0.28960865
## [18,] -0.1127499 -0.39407030 -0.38836278  3.97985093
## [19,]  2.9195668 -1.58646124  0.97612613  0.78629766
## [20,]  2.2998485 -1.73396487 -2.82423222 -0.23281758
## [21,]  1.1501667  0.13531015  0.28506743 -2.19770548
## [22,] -5.6594827 -1.09730404  0.10043541 -0.05245484
## [23,] -0.1011749 -0.57911362  0.71128354 -0.44394773
## [24,]  1.3836281  1.95052341 -2.98485490 -0.35942784
## [25,]  0.2727756  2.63013778  1.83189535  0.05207518
## [26,]  4.0565577  1.17534729 -0.81690756  1.66990720
## [27,]  0.8929694  0.79236692  1.26822542 -0.57575615
## [28,]  0.1514495  1.44873320  0.10857670 -0.51040146
## [29,]  3.5592481 -4.76202163  0.75080576  0.64692974
## [30,] -4.1184576 -0.38073981  1.43463965  0.63330834
## [31,] -0.6811731  1.66926027 -2.88645794 -1.30977099
## [32,]  1.7157269 -1.30836339 -0.55971313 -0.70557980
## [33,] -1.8860627  0.59058174  1.43570145  0.18239089
## [34,]  1.9526349  0.52395429 -0.75642216  0.44289927
## [35,]  1.5888864 -3.12998571 -1.73107199 -1.68604766
## [36,]  1.0709414 -1.65628271  0.79436888 -1.85172698
```

```
## [37,] -4.1101715  0.15766712  2.36296974 -0.56868399
## [38,] -0.7254706  2.89263339 -0.36348376 -0.50612576
## [39,] -3.3451254 -0.95045293  0.19551398 -0.27716645
## [40,] -1.0644466 -1.05265304  0.82886286 -0.12042931
## [41,]  1.4933989  1.86712106  1.81853582 -1.06112429
## [42,] -0.6789284  1.83156328 -1.65435992  0.95121379
## [43,] -2.4164258 -0.46701087  1.42808323  0.41149015
## [44,]  2.2978729  0.41865689 -0.64422929 -0.63462770
## [45,] -2.9245282 -1.19488555 -3.35139309 -1.48966984
## [46,]  1.7654525  0.95655926  0.98576138  1.05683769
## [47,]  2.3125056  2.56161119 -1.58223354  0.59863946
```

Next I ran a linear regression model using those 4 PCs.

```
#Linear Regression of first 4 PC variables
crime_PC <- cbind(PC_4, crime[,16])
model_PCA <- lm(V5~., data=as.data.frame(crime_PC))
summary(model_PCA)

##
## Call:
## lm(formula = V5 ~ ., data = as.data.frame(crime_PC))
##
## 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

#R2 much lower compared to last weeks hw
```

From this summary step I realized the R squared value was lower than last weeks homework.

From here the next step is to convert the PCA model in terms of the original data not the scaled PC values. You need a variety of values to do that. You need the Y intercept, the beta values of the PC components, sigma, and Mu. From there you can get the beta and alpha values of the original data. Then you can get an estimate of the test data point's crime rate and see the quality of the fit (R squared).

```r
#Converting model in terms of original crime data
#Y intercept
beta_int <- model_PCA$coefficients[1]
beta_int
```

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

```r
#Get the rest of Beta Values
beta_rest <- model_PCA$coefficients[2:5]
beta_rest
```

```
##       PC1       PC2       PC3       PC4
##  65.21593 -70.08312  25.19408  69.44603
```

```r
#Get Alpha values. Multiply Coefficients by rotation values
alpha <- (crime_PCA$rotation[,1:4])*(beta_rest)
alpha
```

```
##                PC1         PC2           PC3         PC4
## M      -19.8068566   4.3614585   4.343962799   1.4265676
## So      23.1891930 -10.3283897   1.079421210   7.3685576
## Ed       8.5564501 -15.0406449   4.417702643   5.5378867
## Po1     21.4334144  -6.7978060  -3.549416737  21.7332469
## Po2     20.2816882 -18.3311826   1.336926706 -24.6642181
## LF     -12.3470739  20.8319521  18.856693019  -3.6094369
## M.F      2.9321426 -27.6368772 -13.249409219   0.7278145
## Pop      7.8528436 -11.7715439  -5.397878677  -2.0937661
## NW     -19.1465148 -15.8344721   1.985687941 -16.7680664
## U1      -2.8384622   0.5265788 -45.766955033  -4.6052498
## U2       0.4565742  19.6031847 -37.727456624  -4.7843539
## Wealth  26.3688874  -1.9446961  -0.705370216   7.6835794
## Ineq   -23.8558426  -1.9113212  -0.007418555   5.6533335
## Prob    18.1435807  10.3247954  -8.171898035  12.4215342
## Time    -0.5197204  26.6419828  14.580094844 -37.5418312
```

```r
#Get orginal data alpha and beta values
#Original Alpha is is alpha divided by sigma. Original Beta is intercept- (al
pha*mu)/sigma)
mu <- sapply(crime[,1:15],mean) #sapply funciton returns the average value of
all 15 predictors in the original data
mu
```

```
##            M           So           Ed          Po1          Po2
## 1.385745e+01 3.404255e-01 1.056383e+01 8.500000e+00 8.023404e+00
##           LF          M.F          Pop           NW           U1
## 5.611915e-01 9.830213e+01 3.661702e+01 1.011277e+01 9.546809e-02
##           U2       Wealth         Ineq         Prob         Time
## 3.397872e+00 5.253830e+03 1.940000e+01 4.709138e-02 2.659792e+01
```

```
sigma <- sapply(crime[,1:15],sd) #sapply function returns the standard deviat
ion of all 15 predictors in the original data
sigma
```

```
##            M           So           Ed          Po1          Po2
##   1.25676339   0.47897516   1.11869985   2.97189736   2.79613186
##           LF          M.F          Pop           NW           U1
##   0.04041181   2.94673654  38.07118801  10.28288187   0.01802878
##           U2       Wealth         Ineq         Prob         Time
##   0.84454499 964.90944200   3.98960606   0.02273697   7.08689519
```

```
alpha_orig <- alpha/sigma
alpha_orig
```

```
##                  PC1           PC2           PC3           PC4
## M       -15.76021133   3.470389520  3.456468e+00  1.135112e+00
## So       48.41418685 -21.563518429  2.253606e+00  1.538401e+01
## Ed        7.64856641 -13.444754471  3.948961e+00  4.950288e+00
## Po1       7.21203049  -2.287362310 -1.194327e+00  7.312920e+00
## Po2       7.25348062  -6.555907787  4.781344e-01 -8.820835e+00
## LF     -305.53129511 515.491635526  4.666134e+02 -8.931638e+01
## M.F       0.99504741  -9.378808327 -4.496299e+00  2.469900e-01
## Pop       0.20626736  -0.309198230 -1.417838e-01 -5.499608e-02
## NW       -1.86197946  -1.539886612  1.931062e-01 -1.630678e+00
## U1     -157.44059363  29.207676452 -2.538549e+03 -2.554388e+02
## U2        0.54061555  23.211533875 -4.467193e+01 -5.665008e+00
## Wealth    0.02732784  -0.002015418 -7.310222e-04  7.963006e-03
## Ineq     -5.97949829  -0.479075160 -1.859471e-03  1.417015e+00
## Prob    797.97699438 454.097198772 -3.594101e+02  5.463144e+02
## Time     -0.07333542   3.759330717  2.057332e+00 -5.297359e+00
```

```
beta_orig <- beta_int - sum((alpha*mu)/sigma)
beta_orig
```

```
## (Intercept)
##    2120.063
```

After getting all of these values I can estimate the equation of the line.

```
#Estimate equation of the line in PCA model
#slope is the scaled alpha value and y intercept is original beta
line_est <- crime[,1:15]*(alpha_orig+beta_orig)
```

Using that estimate you can calculate the R squared value.

```
#Find R2 of estimated line
SSE <- sum((line_est-crime[,16]^2))
SS <- sum((crime[,16]-mean(crime[,16]))^2)
R2 <- 1-(SSE/SS)
R2
```

```
## [1] 21.54757
```

Next part is inserting the test data from HW 5.

```
#Test point from HW 5
test_data <- 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)
```

The last part is using the first 4 PCs and applying the test data to predict the crime rate of the test city.

```
#Predict using PCA crime in the test point using first 4 predictors
pred_model <- data.frame(predict(crime_PCA,test_data))
pred <- predict(model_PCA, pred_model)
pred

##        1
## 1112.678
```

Last week I got a predicted crime rate of the test city of 1644 using the 4 most significant predictors not just the first four. Using PCA and first 4 predictors got 1113 as a predicted crime rate. That quite a bit loewr but R2 last week was .90 this week was 0.2155. This shows how picking the best predictors can influence your prediction.