ISYE 6501 Questions 9 and 10

Loading the Libraries

```r
library(tidyverse)
library(tree)
library(rpart)
library(randomForest)
library(gtools)
library(gmodels)
library(pscl)
library(ROCR)
```

Question 9.1

Reading the Dataset

```r
uscrime <- read.table("uscrime.txt", stringsAsFactors = FALSE, header =
TRUE)
head(uscrime)
```

```
##      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
```

```r
uscrime_mu <- colMeans(uscrime[,-16]) # to acquire the mean
uscrime_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
```

```
uscrime_std <- apply(uscrime[,-16], 2, sd) # to acquire the standard deviation
uscrime_std
```

```
##            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
```

Computing Eigenvalues/Eigenvectors

```
mx_us_crime <- as.matrix(uscrime)
US_XTX <- t(mx_us_crime)%*%mx_us_crime # Transpose
uscrime_eig <- eigen(US_XTX)
uscrime_eig$vectors
```

```
##                [,1]          [,2]          [,3]          [,4]
##  [1,] -2.488396e-03 -2.869448e-04 -0.0225839777 -0.1534630299
##  [2,] -5.203481e-05 -2.750586e-04 -0.0005216608 -0.0181246906
##  [3,] -1.944345e-03  3.877277e-04 -0.0141162919 -0.0456035654
##  [4,] -1.621460e-03 -3.045730e-03  0.0214880041  0.0284684426
##  [5,] -1.530797e-03 -2.646528e-03  0.0195901264  0.0279110059
##  [6,] -1.022576e-04  1.341849e-05 -0.0007631936 -0.0040132934
##  [7,] -1.786130e-02  2.461697e-03 -0.1374162781 -0.7756073341
##  [8,] -7.046714e-03 -2.420865e-02  0.9866366490 -0.0948613997
##  [9,] -1.644189e-03 -9.258303e-03  0.0210501559 -0.4250459148
## [10,] -1.735235e-05  7.757484e-06 -0.0001098753 -0.0007463730
## [11,] -6.195534e-04 -2.499495e-04  0.0022554451 -0.0268027785
## [12,] -9.851364e-01  1.706846e-01 -0.0004672221  0.0168446238
## [13,] -3.408100e-03 -1.894275e-03 -0.0267329647 -0.3269365874
## [14,] -8.126420e-06  1.571792e-05 -0.0002017833 -0.0009373122
## [15,] -4.830394e-03 -2.461746e-03  0.0656957538 -0.2706339413
## [16,] -1.705419e-01 -9.849683e-01 -0.0251032088  0.0084879970
##                [,5]          [,6]          [,7]          [,8]
##  [1,] -0.0203627672  0.0061250356  0.0195978863  0.1175015155
##  [2,]  0.0216249191 -0.0128565410  0.0539479198 -0.0602323234
##  [3,] -0.0627450258 -0.0183563443 -0.0743596731  0.1853176640
##  [4,]  0.0598036098 -0.0511924947 -0.5526342490 -0.4568692653
##  [5,]  0.0618556445 -0.0594209434 -0.5249705389 -0.4216684691
##  [6,] -0.0030158987 -0.0002146271 -0.0003536561  0.0056386910
##  [7,] -0.4595015917 -0.1650199697 -0.2605463963  0.1821602994
##  [8,] -0.0989649274 -0.0789032293  0.0067295319  0.0285926690
##  [9,]  0.8572149475 -0.2619260534  0.0093226928  0.1052097806
## [10,] -0.0008529719 -0.0004728415 -0.0002593961  0.0008238155
## [11,] -0.0072457484 -0.0101685126  0.0200672179 -0.0942884130
## [12,]  0.0081949893 -0.0006382427  0.0045296618 -0.0008079730
## [13,] -0.0931702612 -0.0342832520  0.5783541440 -0.7124081456
## [14,]  0.0003008058 -0.0018286851  0.0005765411 -0.0009844370
## [15,]  0.1522482589  0.9433767931 -0.0850849262 -0.0127747632
```

```
## [16,] -0.0059288832  0.0019014804  0.0020453897  0.0026508481
##                [,9]          [,10]         [,11]         [,12]
##  [1,]  0.6442051690 -6.951187e-01  0.2330193171 -0.0884965999
##  [2,]  0.0272529296 -9.675903e-02  0.0124705095  0.9852971948
##  [3,]  0.1990778543  4.968556e-01  0.8081807438  0.0381177180
##  [4,]  0.0569931408 -2.109145e-02 -0.0327165297 -0.0696468892
##  [5,]  0.1037337057  1.942041e-03  0.1220505450  0.0669673938
##  [6,]  0.0108378820  2.263280e-02 -0.0067442453 -0.0411648855
##  [7,] -0.1273578114  4.031239e-02 -0.1440006373  0.0301360195
##  [8,]  0.0038582304 -1.606861e-03 -0.0035096387  0.0011072042
##  [9,] -0.0389789002  4.818095e-02 -0.0058305162 -0.0184677175
## [10,] -0.0135241198 -7.029711e-03  0.0092031432 -0.0089729724
## [11,] -0.7063554696 -4.908599e-01  0.4968893704 -0.0447327817
## [12,]  0.0003425533 -2.309926e-04 -0.0002219537 -0.0002117256
## [13,]  0.1136488914  1.170936e-01  0.0926785864 -0.0725861804
## [14,]  0.0008968368 -1.159858e-03  0.0043064821  0.0110769182
## [15,] -0.0265734307  3.109846e-02  0.0019296603  0.0123066890
## [16,] -0.0005321070 -4.567439e-05 -0.0005464009  0.0001058019
##               [,13]         [,14]         [,15]         [,16]
##  [1,] -3.072458e-02  4.558800e-03 -2.001634e-03  1.121384e-03
##  [2,] -9.760686e-02  3.942863e-02 -1.321279e-02  1.313414e-02
##  [3,] -1.110117e-01 -9.811285e-03 -5.605126e-03 -3.654047e-03
##  [4,] -6.843468e-01 -1.558464e-02 -1.344233e-02 -4.048838e-03
##  [5,]  7.122544e-01  2.026607e-02  1.294297e-02  6.519428e-03
##  [6,] -2.310811e-02  9.845771e-01  2.620492e-02  1.641943e-01
##  [7,]  1.650683e-02 -4.414053e-03  2.272292e-04 -1.834733e-03
##  [8,]  3.165408e-04 -7.517914e-05  2.506794e-05 -3.177725e-05
##  [9,] -2.765968e-03 -1.213514e-03 -7.797279e-04 -4.077780e-04
## [10,] -1.746065e-04 -1.590247e-01 -1.441647e-01  9.764875e-01
## [11,] -1.753098e-02  2.297932e-02  1.063887e-03 -1.446106e-02
## [12,] -2.587347e-05 -1.508966e-05 -1.037458e-06  3.307259e-06
## [13,] -2.517970e-03 -3.806091e-03 -1.259917e-03  4.739281e-04
## [14,] -2.003237e-02 -4.929824e-02  9.889217e-01  1.380333e-01
## [15,]  6.256945e-03 -2.095904e-04  1.451310e-03  5.104355e-04
## [16,]  2.428929e-04 -9.469304e-06  2.948580e-05  3.784198e-06
```

```
for (e in 1:ncol(uscrime)){
  print(det(US_XTX - uscrime_eig$values[e]*diag(ncol(uscrime)))%*
%uscrime_eig$vectors[,e])
}
```

```
##                [,1]           [,2]           [,3]          [,4]
## [1,] -1.066148e+129 -2.229421e+127 -8.330509e+128 -6.94711e+128
##                [,5]           [,6]           [,7]          [,8]
## [1,] -6.558668e+128 -4.381207e+127 -7.652637e+129 -3.01915e+129
##                [,9]          [,10]          [,11]          [,12]
## [1,] -7.044493e+128 -7.434581e+126 -2.654464e+128 -4.220797e+131
##               [,13]          [,14]          [,15]          [,16]
## [1,] -1.460194e+129 -3.481748e+126 -2.069573e+129 -7.306832e+130
##               [,1]          [,2]         [,3]          [,4]
```

```
[,5]
## [1,] -1.221147e+91 -1.170563e+91 1.650048e+91 -1.296167e+92
-1.126279e+92
##              [,6]         [,7]          [,8]          [,9]
[,10]
## [1,] 5.71049e+89 1.047621e+92 -1.030244e+93 -3.940044e+92
3.301342e+89
##               [,11]        [,12]          [,13]        [,14]
[,15]
## [1,] -1.063707e+91 7.263803e+93 -8.061442e+91 6.689054e+89
-1.047642e+92
##               [,16]
## [1,] -4.191717e+94
##                [,1]          [,2]          [,3]         [,4]
[,5]
## [1,] -4.733115e+65 -1.093289e+64 -2.958471e+65 4.503423e+65
4.105668e+65
##                [,6]          [,7]         [,8]         [,9]
[,10]
## [1,] -1.599489e+64 -2.879949e+66 2.067778e+67 4.411659e+65
-2.30275e+63
##              [,11]         [,12]          [,13]         [,14]
[,15]
## [1,] 4.726927e+64 -9.791968e+63 -5.602653e+65 -4.228943e+63
1.376841e+66
##               [,16]
## [1,] -5.261092e+65
##                [,1]          [,2]          [,3]         [,4]
[,5]
## [1,] -1.391497e+62 -1.643422e+61 -4.135016e+61 2.581322e+61
2.530777e+61
##                [,6]          [,7]          [,8]          [,9]
[,10]
## [1,] -3.638977e+60 -7.032672e+62 -8.601377e+61 -3.854023e+62
-6.767595e+59
##               [,11]        [,12]          [,13]       [,14]
[,15]
## [1,] -2.430291e+61 1.527354e+61 -2.964435e+62 -8.4989e+59
-2.453922e+62
##             [,16]
## [1,] 7.69633e+60
##                [,1]         [,2]          [,3]         [,4]
[,5]
## [1,] -1.178118e+49 1.251142e+49 -3.630207e+49 3.460027e+49
3.578751e+49
##                [,6]          [,7]          [,8]         [,9]
[,10]
## [1,] -1.744893e+48 -2.658515e+50 -5.725764e+49 4.959545e+50
-4.934997e+47
##               [,11]        [,12]          [,13]         [,14]
```

```
[,15]
## [1,] -4.192136e+48 4.741333e+48 -5.390505e+49 1.740357e+47
8.80855e+49
##              [,16]
## [1,] -3.430244e+48
##              [,1]          [,2]          [,3]          [,4]
[,5]
## [1,] 1.959908e+47 -4.113877e+47 -5.873721e+47 -1.638074e+48
-1.90137e+48
##               [,6]          [,7]         [,8]          [,9]
[,10]
## [1,] -6.867705e+45 -5.280361e+48 -2.52477e+48 -8.381192e+48
-1.513013e+46
##              [,11]         [,12]         [,13]         [,14]
[,15]
## [1,] -3.253753e+47 -2.042269e+46 -1.097006e+48 -5.851484e+46
3.018647e+49
##             [,16]
## [1,] 6.084417e+46
##             [,1]         [,2]          [,3]          [,4]
[,5]
## [1,] 3.78299e+38 1.041359e+39 -1.435369e+39 -1.066753e+40
-1.013353e+40
##               [,6]         [,7]         [,8]         [,9]
[,10]
## [1,] -6.826642e+36 -5.02934e+39 1.299005e+38 1.799564e+38
-5.007137e+36
##             [,11]        [,12]      [,13]      [,14]          [,15]
## [1,] 3.873585e+38 8.743629e+37 1.1164e+40 1.1129e+37 -1.642399e+39
##             [,16]
## [1,] 3.948226e+37
##              [,1]          [,2]        [,3]          [,4]
[,5]
## [1,] 3.748328e+37 -1.921426e+37 5.91168e+37 -1.457424e+38
-1.345133e+38
##              [,6]         [,7]         [,8]         [,9]
[,10]
## [1,] 1.798757e+36 5.810959e+37 9.121133e+36 3.356219e+37
2.627992e+35
##              [,11]         [,12]       [,13]         [,14]
[,15]
## [1,] -3.007824e+37 -2.577454e+35 -2.2726e+38 -3.140379e+35
-4.075182e+36
##             [,16]
## [1,] 8.456272e+35
##               [,1]          [,2]          [,3]          [,4]
[,5]
## [1,] -2.871713e+35 -1.214871e+34 -8.874416e+34 -2.540618e+34
-4.624201e+34
##              [,6]         [,7]          [,8]         [,9]
```

```
[,10]
## [1,] -4.83127e+33 5.677308e+34 -1.719907e+33 1.737586e+34
6.02873e+33
##              [,11]         [,12]         [,13]         [,14]
[,15]
## [1,] 3.148764e+35 -1.527021e+32 -5.066197e+34 -3.997885e+32
1.18458e+34
##              [,16]
## [1,] 2.372006e+32
##                [,1]          [,2]         [,3]          [,4]
[,5]
## [1,] -1.347816e+35 -1.876131e+34 9.633895e+34 -4.089573e+33
3.765564e+32
##               [,6]         [,7]          [,8]         [,9]
[,10]
## [1,] 4.388438e+33 7.816462e+33 -3.115659e+32 9.342154e+33
-1.363042e+33
##               [,11]         [,12]        [,13]         [,14]
[,15]
## [1,] -9.517639e+34 -4.478882e+31 2.270413e+34 -2.248933e+32
6.029906e+33
##               [,16]
## [1,] -8.856139e+30
##                [,1]          [,2]          [,3]         [,4]
[,5]
## [1,] -7.195422e+31 -3.850779e+30 -2.495588e+32 1.010256e+31
-3.768809e+31
##               [,6]         [,7]         [,8]        [,9]
[,10]
## [1,] 2.082561e+30 4.446607e+31 1.083744e+30 1.80041e+30
-2.841846e+30
##               [,11]        [,12]        [,13]         [,14]
[,15]
## [1,] -1.534349e+32 6.853727e+28 -2.86183e+31 -1.329802e+30
-5.958614e+29
##              [,16]
## [1,] 1.687236e+29
##                [,1]         [,2]         [,3]          [,4]
[,5]
## [1,] -1.475098e+32 1.642334e+33 6.353619e+31 -1.160903e+32
1.11624e+32
##                [,6]         [,7]         [,8]          [,9]
[,10]
## [1,] -6.861533e+31 5.023196e+31 1.845534e+30 -3.078276e+31
-1.495652e+31
##               [,11]        [,12]         [,13]        [,14]
[,15]
## [1,] -7.456245e+31 -3.52913e+29 -1.209896e+32 1.846347e+31
2.05133e+31
##             [,16]
```

```
## [1,] 1.76355e+29
##              [,1]         [,2]         [,3]        [,4]
[,5]
## [1,] 2.949667e+29 9.370601e+29 1.065751e+30 6.56997e+30
-6.837893e+30
##              [,6]          [,7]          [,8]         [,9]
[,10]
## [1,] 2.218459e+29 -1.584714e+29 -3.038903e+27 2.655426e+28
1.676284e+27
##             [,11]        [,12]       [,13]        [,14]
[,15]
## [1,] 1.683036e+29 2.483944e+26 2.41734e+28 1.923178e+29
-6.006887e+28
##              [,16]
## [1,] -2.331857e+27
##              [,1]         [,2]          [,3]          [,4]
[,5]
## [1,] 2.653696e+24 2.295156e+25 -5.711189e+24 -9.071885e+24
1.179696e+25
##              [,6]          [,7]          [,8]          [,9]
[,10]
## [1,] 5.731263e+26 -2.569438e+24 -4.376208e+22 -7.063911e+23
-9.256891e+25
##             [,11]         [,12]         [,13]        [,14]
[,15]
## [1,] 1.337636e+25 -8.783751e+21 -2.215541e+24 -2.86967e+25
-1.220034e+23
##             [,16]
## [1,] -5.51212e+21
##              [,1]         [,2]         [,3]         [,4]
[,5]
## [1,] 3.183593e+22 2.101491e+23 8.914937e+22 2.137999e+23
-2.058576e+23
##               [,6]          [,7]          [,8]         [,9]
[,10]
## [1,] -4.167886e+23 -3.614074e+21 -3.987049e+20 1.240155e+22
2.292935e+24
##              [,11]        [,12]        [,13]         [,14]
[,15]
## [1,] -1.692109e+22 1.650073e+19 2.003894e+22 -1.572877e+25
-2.308304e+22
##              [,16]
## [1,] -4.689708e+20
##              [,1]         [,2]          [,3]          [,4]
[,5]
## [1,] 2.127842e+22 2.492222e+23 -6.933607e+22 -7.682729e+22
1.237071e+23
##              [,6]          [,7]          [,8]          [,9]
[,10]
## [1,] 3.115612e+24 -3.481433e+22 -6.02978e+20 -7.737648e+21
```

```
1.852899e+25
##               [,11]        [,12]        [,13]        [,14]
[,15]
## [1,] -2.744008e+23 6.275572e+19 8.992855e+21 2.619203e+24
9.685588e+21
##             [,16]
## [1,] 7.18057e+19
```

By doing the computations for the eigenvalues and eigenvectors, there are sixteen eigenvector columns in the matrix. After running the loop functions, the values in the eigenvalues are much closer to zero and lower than one.

Setting the Seed

```r
set.seed(2018)
```

Running the Principal Component Analysis

```r
crime_pca <-
prcomp(~M+So+Ed+Po1+Po2+LF+M.F+Pop+NW+U1+U2+Wealth+Ineq+Prob+Time,uscri
me, 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
```

```r
crime_pca$x
```

```
##            PC1         PC2         PC3         PC4          PC5
PC6
## 1  -4.1992835 -1.09383120 -1.11907395  0.67178115  0.055283376
0.30733835
## 2   1.1726630  0.67701360 -0.05244634 -0.08350709 -1.173199821
-0.58323731
## 3  -4.1737248  0.27677501 -0.37107658  0.37793995  0.541345246
0.71872230
## 4   3.8349617 -2.57690596  0.22793998  0.38262331 -1.644746496
```

```
0.72948841
## 5   1.8392999  1.33098564  1.27882805  0.71814305  0.041590320
-0.39409015
## 6   2.9072336 -0.33054213  0.53288181  1.22140635  1.374360960
-0.69225131
## 7   0.2457752 -0.07362562 -0.90742064  1.13685873  0.718644387
-0.93107472
## 8  -0.1301330 -1.35985577  0.59753132  1.44045387 -0.222781388
0.04912052
## 9  -3.6103169 -0.68621008  1.28372246  0.55171150 -0.324292990
0.12683417
## 10  1.1672376  3.03207033  0.37984502 -0.28887026 -0.646056610
0.33130781
## 11  2.5384879 -2.66771358  1.54424656 -0.87671210 -0.324083561
0.44365740
## 12  1.0065920 -0.06044849  1.18861346 -1.31261964  0.358087724
0.25696957
## 13  0.5161143  0.97485189  1.83351610 -1.59117618  0.599881946
1.04761756
## 14  0.4265556  1.85044812  1.02893477 -0.07789173  0.741887592
0.61569775
## 15 -3.3435299  0.05182823 -1.01358113  0.08840211  0.002969448
0.17074576
## 16 -3.0310689 -2.10295524 -1.82993161  0.52347187 -0.387454246
-0.20965321
## 17 -0.2262961  1.44939774 -1.37565975  0.28960865  1.337784608
-0.25633983
## 18 -0.1127499 -0.39407030 -0.38836278  3.97985093  0.410914404
0.09317136
## 19  2.9195668 -1.58646124  0.97612613  0.78629766  1.356288600
-0.89044651
## 20  2.2998485 -1.73396487 -2.82423222 -0.23281758 -0.653038858
0.68615337
## 21  1.1501667  0.13531015  0.28506743 -2.19770548  0.084621572
0.45958300
## 22 -5.6594827 -1.09730404  0.10043541 -0.05245484 -0.689327990
0.13338054
## 23 -0.1011749 -0.57911362  0.71128354 -0.44394773  0.689939865
0.54002731
## 24  1.3836281  1.95052341 -2.98485490 -0.35942784 -0.744371276
0.01453851
## 25  0.2727756  2.63013778  1.83189535  0.05207518  0.803692524
1.52313508
## 26  4.0565577  1.17534729 -0.81690756  1.66990720 -2.895110075
-0.47766314
## 27  0.8929694  0.79236692  1.26822542 -0.57575615  1.830793964
-1.11656766
## 28  0.1514495  1.44873320  0.10857670 -0.51040146 -1.023229895
-0.74149513
## 29  3.5592481 -4.76202163  0.75080576  0.64692974  0.309946510
```

```
0.72486153
## 30 -4.1184576 -0.38073981  1.43463965  0.63330834 -0.254715638
-0.42316550
## 31 -0.6811731  1.66926027 -2.88645794 -1.30977099 -0.470913997
-0.45866080
## 32  1.7157269 -1.30836339 -0.55971313 -0.70557980  0.331277622
1.30802615
## 33 -1.8860627  0.59058174  1.43570145  0.18239089  0.291863659
-0.13885903
## 34  1.9526349  0.52395429 -0.75642216  0.44289927  0.723474420
-0.42036754
## 35  1.5888864 -3.12998571 -1.73107199 -1.68604766  0.665406182
0.54144206
## 36  1.0709414 -1.65628271  0.79436888 -1.85172698  0.020031154
-2.43356674
## 37 -4.1101715  0.15766712  2.36296974 -0.56868399 -2.469679496
0.07239996
## 38 -0.7254706  2.89263339 -0.36348376 -0.50612576  0.028157162
1.06465126
## 39 -3.3451254 -0.95045293  0.19551398 -0.27716645  0.487259213
-0.20571166
## 40 -1.0644466 -1.05265304  0.82886286 -0.12042931 -0.645884788
0.63320546
## 41  1.4933989  1.86712106  1.81853582 -1.06112429  0.009855774
-1.03480444
## 42 -0.6789284  1.83156328 -1.65435992  0.95121379  2.115630145
-0.02332805
## 43 -2.4164258 -0.46701087  1.42808323  0.41149015 -0.867397522
-1.13982198
## 44  2.2978729  0.41865689 -0.64422929 -0.63462770 -0.703116983
-0.65215040
## 45 -2.9245282 -1.19488555 -3.35139309 -1.48966984  0.806659622
-0.48157983
## 46  1.7654525  0.95655926  0.98576138  1.05683769  0.542466034
0.71712602
## 47  2.3125056  2.56161119 -1.58223354  0.59863946 -1.140712406
0.39563373
##              PC7          PC8          PC9         PC10          PC11
## 1  -0.566408161 -0.007801727  0.223509947  0.452743650 -0.0847454174
## 2   0.195611187  0.154566472  0.436777195  0.212085890 -0.0339166059
## 3   0.103306929  0.351138883  0.062992321 -0.067190215 -0.4814915573
## 4   0.266994985 -1.547460841 -0.379541806  0.229223052  0.1098495110
## 5   0.070507664 -0.543237437  0.224632448  0.477690842 -0.3295818584
## 6   0.226482092  0.562323186  0.417722172  0.091009390  0.0102296864
## 7   0.307507661  1.056861503 -1.160218292  0.791683164  0.2829470570
## 8   0.911404993  0.693339330 -0.421314146  0.613278523 -0.3211719754
## 9  -0.417420968 -0.053270500  0.232662026  0.065541569  0.1212937342
## 10  0.009579488 -0.329270845 -0.123629746  0.200126861 -0.0005664179
## 11 -0.182961180  0.587179568 -0.070907596 -0.556615080 -0.1727018439
## 12 -0.462577031  0.307351101 -0.105197263 -0.132898969  0.2984659116
```

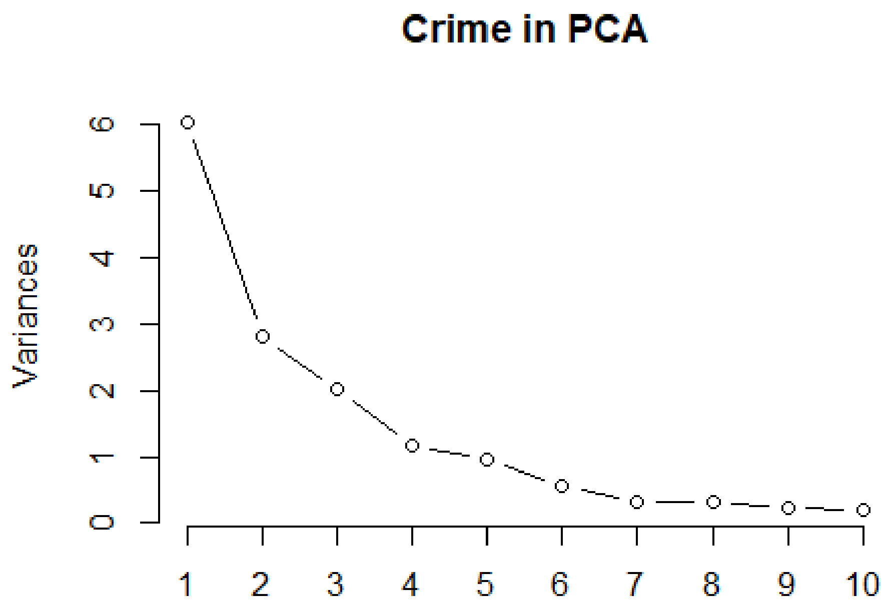
```
## 13 -0.494631320  0.753702337 -0.384056907 -0.340154686 -0.3093005372
## 14 -0.087093101 -0.046931419 -0.159138488  0.280005792  0.1705829803
## 15  1.040213660 -0.139392628 -0.147546022 -1.024276227  0.7966941694
## 16  0.262430717  0.641818600  0.526895635  0.828407330 -0.2016395195
## 17 -0.754882880 -0.959968310  0.351808733 -0.046049514  0.1106976222
## 18 -1.227238054  0.280226677 -0.412734008 -1.074780984  0.1309449295
## 19  0.387161139 -0.002276046  0.555855685  0.598093089  0.3873076362
## 20 -0.401936004  0.240456772  0.341543809  0.229195572  0.7640552201
## 21 -0.179283176  0.772072202 -0.344317021 -0.192047623 -0.2491916653
## 22 -1.337728458  0.261648468  0.225568667  0.361253314 -1.2502555533
## 23  0.995827754  0.371597176  1.073655584  0.033997150 -0.0148920689
## 24  0.042135169 -0.210603749 -0.111463892  0.570729260 -0.2891751385
## 25 -0.341012092  0.390172476 -0.015090214 -0.107776581  0.0126408264
## 26 -0.110906098  0.991890307  0.232407672 -0.727397771 -0.1821057801
## 27 -0.199196211 -0.044269305 -0.015729946 -0.046457518 -0.2413405035
## 28  0.113082804 -0.677219677  0.151930973  0.076617716 -0.4139560352
## 29  0.248081636 -0.844089307  0.230269486 -0.342149453 -0.8429456727
## 30 -0.116127247 -0.891169193 -0.011731985 -0.435636015  0.0144413727
## 31  0.704852096 -0.538600585  0.439137868 -0.709658521 -0.5740441221
## 32 -0.786980332 -0.067086938 -0.169888285  0.072917031  0.6056884273
## 33  0.767856496  0.027448832 -0.773125607  0.126124015  0.1459949892
## 34  0.181257930  0.115379461 -0.101718594  0.321007813 -0.4060548228
## 35 -0.449541256 -0.276891496  0.007657702  0.202491328  0.0936192141
## 36 -0.333843509  0.384707595  0.642612190 -0.727991803  0.1824929850
## 37 -0.343611407  0.157984131  0.915881371  0.481641023  1.1919120577
## 38  0.863051754 -0.058247210  0.341385143 -0.133649827 -0.5185529852
## 39  0.966860079  0.059557654  0.039345212  0.034036490  0.2185933062
## 40  0.767470212 -0.704833575 -1.109887730  0.106827471  0.1951224135
## 41 -0.589160590 -0.468876595 -0.528478950  0.430811630  0.1829897714
## 42 -0.557413301 -0.963360913  0.485515025  0.007295728  0.4739341401
## 43  0.041128192 -0.573696577 -0.773992630 -0.447789368 -0.1172352964
## 44 -0.442990964 -0.093002011 -0.515838387  0.241578722 -0.1363783451
## 45  0.233636019  0.379908278 -0.815127937 -0.541397364  0.2642920144
## 46  0.847914876  0.172381544  0.657987377 -0.480124036  0.1175554086
## 47 -0.171412192  0.327844331 -0.167078790 -0.002371858  0.2888983375
##           PC12         PC13         PC14          PC15
## 1   0.22096639 -0.112616798  0.326964861  0.0233840087
## 2   0.35686524  0.297516509  0.252356741 -0.0607636781
## 3  -0.04701948  0.052160542 -0.486551130  0.0421174952
## 4   0.17727101  0.088381306  0.149678420  0.0291749700
## 5   0.41807551 -0.722152235  0.131027187 -0.0751493967
## 6  -0.70661980 -0.135172709  0.194925675  0.0155861048
## 7  -0.65196573  0.168327740  0.145473719 -0.0654492790
## 8   0.49089082  0.218057687 -0.623230400 -0.0259344691
## 9  -0.29249322 -0.242429444  0.026476592  0.0252300906
## 10 -0.21063943 -0.257769674 -0.276967642  0.0232404560
## 11 -0.33472808  0.238074383  0.255472039  0.0992321732
## 12 -0.26641418  0.171319693  0.094123766  0.0190525547
## 13  0.59785665 -0.132203906  0.027925309 -0.0148583070
## 14  0.18719968  0.571485989  0.250689865  0.0127642083
```

```
## 15  0.56068471  0.217331625  0.037229143  0.0452385996
## 16 -0.16367226 -0.082957159  0.137971468 -0.0210413021
## 17  0.33986466 -0.128534101 -0.246396571 -0.0073811334
## 18 -0.16259339 -0.474477655  0.096820598  0.0107830419
## 19  0.49141798  0.110318335 -0.185686144  0.1027680411
## 20  0.05854928  0.173991982  0.041243802 -0.0108009160
## 21  0.03436398 -0.407556122  0.094462966 -0.0062668835
## 22  0.15171519  0.319206246  0.003834903 -0.0005073113
## 23  0.08607424 -0.037204214  0.545497655  0.0129578778
## 24 -0.20783571 -0.240516367 -0.122497400 -0.0342080182
## 25  0.37619331  0.117057471 -0.105183565 -0.0510978767
## 26  0.30036333  0.137225797 -0.134072192 -0.1184870411
## 27 -0.51580918  0.066145794 -0.186576416  0.0791823778
## 28  0.24306271 -0.140043507  0.629391628 -0.0354269136
## 29  0.03561083 -0.229673348 -0.234477116  0.0387679658
## 30 -0.36730664  0.388569856 -0.025869303 -0.0300544785
## 31 -0.79220655  0.007892720 -0.201914013  0.0766956405
## 32 -0.34195913  0.154638372  0.085491563 -0.0800132601
## 33  0.25911938 -0.316086918 -0.024206874  0.1045722437
## 34  0.25952688  0.166191625  0.152140934  0.0830313640
## 35 -0.33281300  0.047752123 -0.312239740 -0.1013067365
## 36  0.47165172  0.049320737 -0.382422475 -0.0704633747
## 37 -0.31784996 -0.395326593 -0.238009619  0.0858414347
## 38 -0.25514910  0.169135060 -0.013058191 -0.0353381517
## 39  0.08796506  0.030789317 -0.067516845 -0.1026461875
## 40 -0.05840207 -0.137544171 -0.177710919 -0.0704026331
## 41 -0.26187866 -0.058757893 -0.113235908 -0.0939372094
## 42  0.33534399  0.291642167  0.013605734 -0.0399895760
## 43 -0.26398492  0.427157629  0.266115989 -0.0276514754
## 44  0.17238472  0.005592707  0.142206916  0.1612571077
## 45  0.39144866 -0.508852301  0.223930669  0.0073779464
## 46 -0.56753437 -0.172018049  0.056680914 -0.0850410458
## 47  0.01440895  0.246609753 -0.223916593  0.1659609523
```

By utilizing the Principal Component Analysis for all the variables, it was determined that the first element on the data that is given shows the highest standard deviation. However, the cumulative proportion gives a higher number when it goes down through further components.

Creating the Screeplot

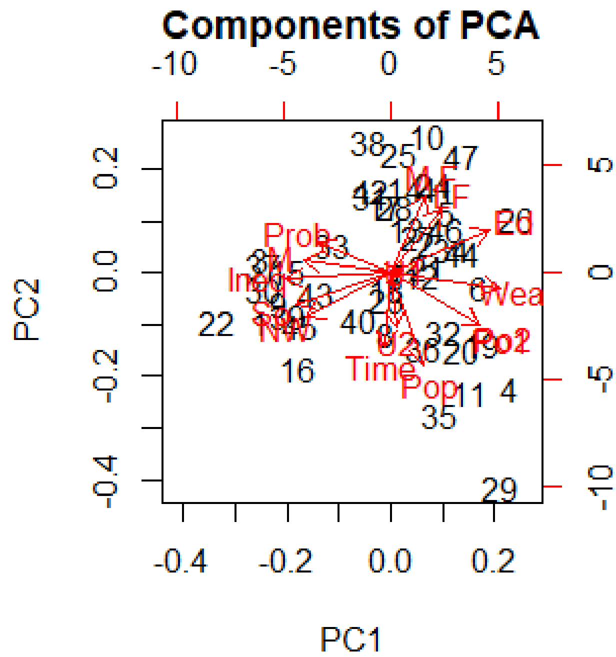
```
screeplot(crime_pca, type="lines", main = "Crime in PCA")
```



By looking at this plot, 3 is the most optimal clusters to use on the PCA.

Biplot

```
biplot(crime_pca, main = "Components of PCA")
```



The biplot shows a much more consistent set of components of the factors that determines the rank in the PCA in comparison to the above plot.

Concentrating on the First Four Components

```
crime_four <-
lm(uscrime[,16]~crime_pca$x[,1]+crime_pca$x[,2]+crime_pca$x[,3]+crime_p
ca$x[,4], uscrime)
summary(crime_four)

##
## Call:
## lm(formula = uscrime[, 16] ~ crime_pca$x[, 1] + crime_pca$x[,
##     2] + crime_pca$x[, 3] + crime_pca$x[, 4], data = uscrime)
##
## 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 ***
## crime_pca$x[, 1]    65.22      20.22   3.225  0.00244 **
## crime_pca$x[, 2]   -70.08      29.63  -2.365  0.02273 *
## crime_pca$x[, 3]    25.19      35.03   0.719  0.47602
## crime_pca$x[, 4]    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
```

By looking at the first four components in the Principal Component Analysis for the regression model, the r-squared value comes out to 0.3091, which is a fairly poor model with less variance. However, p-value is within the standards at < 5% threshold.

Rerunning the Original Components

```
uscrime_origin <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
names(uscrime_origin) <-c("M", "So", "Ed", "Po1", "Po2", "LF", "M.F",
"Pop","NW", "U1", "U2", "Wealth", "Ineq", "Prob", "Time")
```

Looking at the Original Components through the Loop

```
for (p in 4) {
  for (name in names(uscrime_origin)) {
    altogether <- crime_four$coefficients[p+1] * crime_pca$rotation[,p]
[[name]]
    uscrime_origin[[name]] <- uscrime_origin[[name]] + altogether
  }
}

uscrime_origin

##            M          So          Ed         Po1         Po2
LF
##   -1.4135995  20.3110062   5.5378867  23.1429304  24.4400090
-9.9492059
##          M.F         Pop          NW          U1          U2
Wealth
##    0.7278145  -2.2295740  16.6156369 -12.6941068  -4.7843539
8.1819594
##         Ineq        Prob        Time
##   -5.6019421  34.2392465 -37.5418312
```

Scaling the Original Components

```
(crime_zero <- coef(crime_four)[1])

## (Intercept)
##    905.0851

(crime_s <- crime_pca$rotation[,1:4] %*% as.matrix(coef(crime_four)[-
1]))

##                [,1]
## M      -21.277963
```

```
## So       10.223091
## Ed       14.352610
## Po1      63.456426
## Po2      64.557974
## LF      -14.005349
## M.F     -24.437572
## Pop      39.830667
## NW       15.434545
## U1      -27.222281
## U2        1.425902
## Wealth   38.607855
## Ineq    -27.536348
## Prob      3.295707
## Time     -6.612616
```

By looking at the scaled version of the crime counts, it is totaled to 905.08.Let's delve to further analysis.

Unscaling the Originals

```
total <- 0
unscale_crime <- rep(0,15)
for(p in 1:15){
  total <- total + crime_s[p]*uscrime_mu[p]/uscrime_std[p]
  unscale_crime[p] <- crime_s[1]/uscrime_std[p]
}
(crime_zero_origin <- crime_zero - total)

## (Intercept)
##    1666.485

crime_zero_origin

## (Intercept)
##    1666.485
```

By scaling back the model to the originals, it has been determined that it comes out to 1648.5 crimes. However, given that the data is binary, PCA does not work too well to determine its factors.

Question 10.1

Reading the Dataset

```
crime_tree <- read.table("uscrime.txt", stringsAsFactors = FALSE,
header = TRUE)
head(crime_tree)

##      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
```

Changing a Different Seed

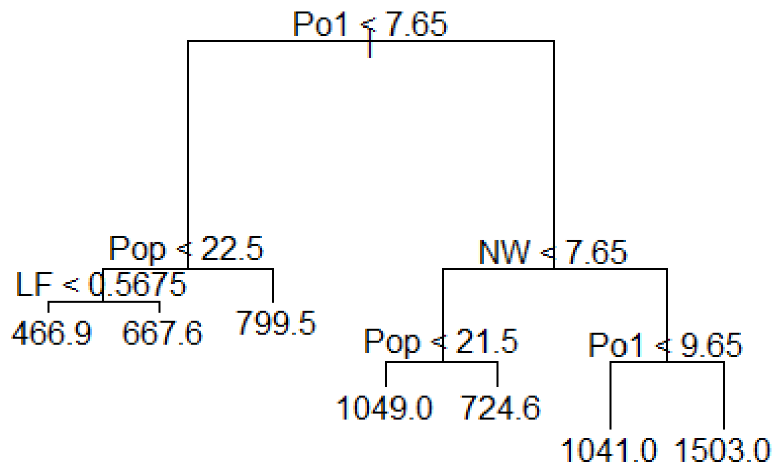
```
set.seed(510)
```

Building the Regression Tree Model

```
crime_regression <- tree(Crime~
  M+So+Ed+Po1+Po2+LF+M.F+Pop+NW+U1+U2+Wealth+Ineq+Prob+Time,crime_tree)
summary(crime_regression)

##
## Regression tree:
## tree(formula = Crime ~ M + So + Ed + Po1 + Po2 + LF + M.F + Pop +
##     NW + U1 + U2 + Wealth + Ineq + Prob + Time, data = crime_tree)
## Variables actually used in tree construction:
## [1] "Po1" "Pop" "LF"  "NW"
## Number of terminal nodes:  7
## Residual mean deviance:  47390 = 1896000 / 40
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -573.900  -98.300   -1.545    0.000  110.600  490.100

plot(crime_regression, main = "Regression Tree for Crime")
text(crime_regression)
```



Calculate Regression Tree Model's R-Square

```
crime_yhat <- predict(crime_regression)
crime_SSres <- sum((crime_yhat - crime_tree$Crime)^2)
crime_SStot <- sum((crime_tree$Crime - mean(crime_tree$Crime))^2)
crime_R2 <- 1 - crime_SSres/crime_SStot
crime_R2

## [1] 0.7244962
```

By computing the R-square for the Regression tree model, it comes out to 0.724, which is a fairly good model to use. Although, let's move on and compute the random forest model.

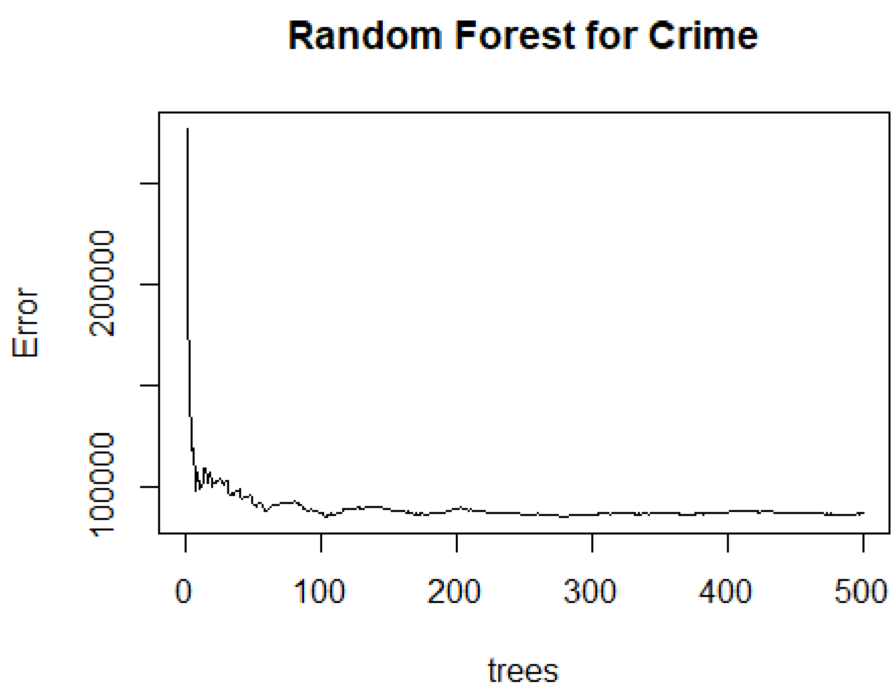
Building the Random Forest Model

```
crime_forest <- randomForest(Crime~
  M+So+Ed+Po1+Po2+LF+M.F+Pop+NW+U1+U2+Wealth+Ineq+Prob+Time,crime_tree)
summary(crime_forest)

##                 Length Class  Mode
## call              3    -none- call
## type              1    -none- character
## predicted        47    -none- numeric
## mse             500    -none- numeric
## rsq             500    -none- numeric
## oob.times        47    -none- numeric
```

```
## importance        15     -none- numeric
## importanceSD       0     -none- NULL
## localImportance    0     -none- NULL
## proximity          0     -none- NULL
## ntree              1     -none- numeric
## mtry               1     -none- numeric
## forest            11     -none- list
## coefs              0     -none- NULL
## y                 47     -none- numeric
## test               0     -none- NULL
## inbag              0     -none- NULL
## terms              3     terms  call
```

```
plot(crime_forest, main = "Random Forest for Crime")
```




```
crime_forest$ntree
```

```
## [1] 500
```

```
crime_forest$importance
```

```
##        IncNodePurity
## M          242497.33
## So          12402.02
## Ed         200946.07
## Po1       1245161.92
## Po2       1157759.09
```

```
## LF         300670.14
## M.F        243807.34
## Pop        390916.39
## NW         560406.13
## U1         130456.17
## U2         167428.60
## Wealth     571954.33
## Ineq       199120.31
## Prob       744765.77
## Time       209504.75
```

Calculate Random Forest R-Squared

```
rf_yhat <- predict(crime_forest)
rf_SSres <- sum((rf_yhat - crime_tree$Crime)^2)
rf_SStot <- sum((crime_tree$Crime - mean(crime_tree$Crime))^2)
rf_R2 <- 1 - rf_SSres/rf_SStot
rf_R2

## [1] 0.4099392
```

By comparing the random forest and regression tree models, there are much more impurities in the random forest analysis. In addition, the random forest has computed to 500 different trees in the model. By calculating the R-squared for the random forest model, it comes out to 0.41, which is a fairly poor model than the regression tree.

Question 10.2

Describe a situation or problem from your job, everyday life, current events, etc., for which a logistic regression model would be appropriate. List some (up to 5) predictors that you might use.

I have heard that logistic regression can be used in many industries. Examples include:

- Getting approved for a loan, who will pay the mortgage off in time vs. who will default
- Voting outcomes: who will vote vs. who will not vote
- Willing to pay: who will buy for a specific item vs. who will not buy for that specific item

Question 10.3

Reading the Dataset for the German Credit

```
gce <- read.table("germancredit.txt", stringsAsFactors = FALSE, header
= FALSE)
head(gce)

##    V1 V2  V3  V4   V5  V6  V7 V8  V9  V10 V11  V12 V13  V14  V15 V16
V17
## 1 A11  6 A34 A43 1169 A65 A75  4 A93 A101   4 A121  67 A143 A152   2
A173
## 2 A12 48 A32 A43 5951 A61 A73  2 A92 A101   2 A121  22 A143 A152   1
A173
## 3 A14 12 A34 A46 2096 A61 A74  2 A93 A101   3 A121  49 A143 A152   1
A172
## 4 A11 42 A32 A42 7882 A61 A74  2 A93 A103   4 A122  45 A143 A153   1
A173
## 5 A11 24 A33 A40 4870 A61 A73  3 A93 A101   4 A124  53 A143 A153   2
A173
## 6 A14 36 A32 A46 9055 A65 A73  2 A93 A101   4 A124  35 A143 A153   1
A172
##   V18  V19  V20 V21
## 1   1 A192 A201   1
## 2   1 A191 A201   2
## 3   2 A191 A201   1
## 4   2 A191 A201   1
## 5   2 A191 A201   2
## 6   2 A192 A201   1
```

Selecting the Random Seed

```
set.seed(818)
```

Logistic Regression Conversion

```
gce$V21 <- as.integer(as.logical(gce$V21 < 2))
```

Creating the Logistic Regression Model

```
gce_log <- glm(V21 ~., family = binomial(link = "logit"), gce)
summary(gce_log)

##
## Call:
## glm(formula = V21 ~ ., family = binomial(link = "logit"), data =
gce)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6116  -0.7095   0.3752   0.6994   2.3410
##
## Coefficients:
```

```
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.005e-01  1.084e+00  -0.369 0.711869
## V1A12        3.749e-01  2.179e-01   1.720 0.085400 .
## V1A13        9.657e-01  3.692e-01   2.616 0.008905 **
## V1A14        1.712e+00  2.322e-01   7.373 1.66e-13 ***
## V2          -2.786e-02  9.296e-03  -2.997 0.002724 **
## V3A31       -1.434e-01  5.489e-01  -0.261 0.793921
## V3A32        5.861e-01  4.305e-01   1.362 0.173348
## V3A33        8.532e-01  4.717e-01   1.809 0.070470 .
## V3A34        1.436e+00  4.399e-01   3.264 0.001099 **
## V4A41        1.666e+00  3.743e-01   4.452 8.51e-06 ***
## V4A410       1.489e+00  7.764e-01   1.918 0.055163 .
## V4A42        7.916e-01  2.610e-01   3.033 0.002421 **
## V4A43        8.916e-01  2.471e-01   3.609 0.000308 ***
## V4A44        5.228e-01  7.623e-01   0.686 0.492831
## V4A45        2.164e-01  5.500e-01   0.393 0.694000
## V4A46       -3.628e-02  3.965e-01  -0.092 0.927082
## V4A48        2.059e+00  1.212e+00   1.699 0.089297 .
## V4A49        7.401e-01  3.339e-01   2.216 0.026668 *
## V5          -1.283e-04  4.444e-05  -2.887 0.003894 **
## V6A62        3.577e-01  2.861e-01   1.250 0.211130
## V6A63        3.761e-01  4.011e-01   0.938 0.348476
## V6A64        1.339e+00  5.249e-01   2.551 0.010729 *
## V6A65        9.467e-01  2.625e-01   3.607 0.000310 ***
## V7A72        6.691e-02  4.270e-01   0.157 0.875475
## V7A73        1.828e-01  4.105e-01   0.445 0.656049
## V7A74        8.310e-01  4.455e-01   1.866 0.062110 .
## V7A75        2.766e-01  4.134e-01   0.669 0.503410
## V8          -3.301e-01  8.828e-02  -3.739 0.000185 ***
## V9A92        2.755e-01  3.865e-01   0.713 0.476040
## V9A93        8.161e-01  3.799e-01   2.148 0.031718 *
## V9A94        3.671e-01  4.537e-01   0.809 0.418448
## V10A102     -4.360e-01  4.101e-01  -1.063 0.287700
## V10A103      9.786e-01  4.243e-01   2.307 0.021072 *
## V11         -4.776e-03  8.641e-02  -0.055 0.955920
## V12A122     -2.814e-01  2.534e-01  -1.111 0.266630
## V12A123     -1.945e-01  2.360e-01  -0.824 0.409743
## V12A124     -7.304e-01  4.245e-01  -1.721 0.085308 .
## V13          1.454e-02  9.222e-03   1.576 0.114982
## V14A142      1.232e-01  4.119e-01   0.299 0.764878
## V14A143      6.463e-01  2.391e-01   2.703 0.006871 **
## V15A152      4.436e-01  2.347e-01   1.890 0.058715 .
## V15A153      6.839e-01  4.770e-01   1.434 0.151657
## V16         -2.721e-01  1.895e-01  -1.436 0.151109
## V17A172     -5.361e-01  6.796e-01  -0.789 0.430160
## V17A173     -5.547e-01  6.549e-01  -0.847 0.397015
## V17A174     -4.795e-01  6.623e-01  -0.724 0.469086
## V18         -2.647e-01  2.492e-01  -1.062 0.288249
## V19A192      3.000e-01  2.013e-01   1.491 0.136060
## V20A202      1.392e+00  6.258e-01   2.225 0.026095 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1221.73  on 999  degrees of freedom
## Residual deviance:  895.82  on 951  degrees of freedom
## AIC: 993.82
##
## Number of Fisher Scoring iterations: 5
```

By looking at the summary of the Logistic Regression model, there are five or six variables that are significant. What it shows is that there is a strong correlation between those variables as well. Let's delve deeper to the different percentage of the models.

Cocentrating at the p-values < 0.05 Level

```
gce0.05_log<- glm(V21~V1+V2+V3+V4+V5+V6+V8+V9+V10+V14+V20,
family=binomial(link="logit"), gce)
summary(gce0.05_log)
```

```
##
## Call:
## glm(formula = V21 ~ V1 + V2 + V3 + V4 + V5 + V6 + V8 + V9 + V10 +
##     V14 + V20, family = binomial(link = "logit"), data = gce)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6952  -0.7637   0.3938   0.7062   2.2211
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.650e-01  6.387e-01  -1.354 0.175659
## V1A12        4.110e-01  2.097e-01   1.960 0.049986 *
## V1A13        1.069e+00  3.583e-01   2.982 0.002861 **
## V1A14        1.779e+00  2.264e-01   7.858 3.89e-15 ***
## V2          -2.805e-02  8.836e-03  -3.175 0.001499 **
## V3A31        1.361e-01  5.198e-01   0.262 0.793395
## V3A32        8.588e-01  4.046e-01   2.123 0.033784 *
## V3A33        9.800e-01  4.607e-01   2.127 0.033421 *
## V3A34        1.587e+00  4.267e-01   3.720 0.000199 ***
## V4A41        1.556e+00  3.599e-01   4.323 1.54e-05 ***
## V4A410       1.575e+00  7.540e-01   2.089 0.036690 *
## V4A42        6.612e-01  2.487e-01   2.659 0.007837 **
## V4A43        8.968e-01  2.393e-01   3.747 0.000179 ***
## V4A44        5.635e-01  7.428e-01   0.759 0.448091
## V4A45        1.782e-01  5.383e-01   0.331 0.740600
## V4A46       -1.800e-01  3.846e-01  -0.468 0.639890
## V4A48        2.133e+00  1.224e+00   1.742 0.081531 .
## V4A49        8.093e-01  3.226e-01   2.509 0.012110 *
```

```
## V5          -1.116e-04  4.092e-05  -2.726 0.006409 **
## V6A62        2.671e-01  2.739e-01   0.975 0.329488
## V6A63        4.271e-01  3.924e-01   1.089 0.276349
## V6A64        1.331e+00  5.038e-01   2.641 0.008255 **
## V6A65        9.677e-01  2.543e-01   3.805 0.000142 ***
## V8          -3.038e-01  8.429e-02  -3.605 0.000312 ***
## V9A92        1.243e-01  3.661e-01   0.340 0.734175
## V9A93        7.705e-01  3.600e-01   2.140 0.032331 *
## V9A94        2.786e-01  4.345e-01   0.641 0.521342
## V10A102     -5.323e-01  4.001e-01  -1.330 0.183402
## V10A103      1.022e+00  4.128e-01   2.476 0.013278 *
## V14A142      1.417e-01  4.007e-01   0.354 0.723700
## V14A143      6.536e-01  2.333e-01   2.802 0.005083 **
## V20A202      1.290e+00  6.220e-01   2.074 0.038064 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1221.73  on 999  degrees of freedom
## Residual deviance:  920.95  on 968  degrees of freedom
## AIC: 984.95
##
## Number of Fisher Scoring iterations: 5
```

Concentrating at the p-values at 0.1 Level

```
gce0.1_log<- glm(V21~V1+V2+V3+V4+V5+V6+V8+V9+V10+V14+V20,
family=binomial(link="logit"), gce)
summary(gce0.1_log)

##
## Call:
## glm(formula = V21 ~ V1 + V2 + V3 + V4 + V5 + V6 + V8 + V9 + V10 +
##     V14 + V20, family = binomial(link = "logit"), data = gce)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6952  -0.7637   0.3938   0.7062   2.2211
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.650e-01  6.387e-01  -1.354 0.175659
## V1A12        4.110e-01  2.097e-01   1.960 0.049986 *
## V1A13        1.069e+00  3.583e-01   2.982 0.002861 **
## V1A14        1.779e+00  2.264e-01   7.858 3.89e-15 ***
## V2          -2.805e-02  8.836e-03  -3.175 0.001499 **
## V3A31        1.361e-01  5.198e-01   0.262 0.793395
## V3A32        8.588e-01  4.046e-01   2.123 0.033784 *
## V3A33        9.800e-01  4.607e-01   2.127 0.033421 *
## V3A34        1.587e+00  4.267e-01   3.720 0.000199 ***
```

```
## V4A41          1.556e+00  3.599e-01   4.323 1.54e-05 ***
## V4A410         1.575e+00  7.540e-01   2.089 0.036690 *
## V4A42          6.612e-01  2.487e-01   2.659 0.007837 **
## V4A43          8.968e-01  2.393e-01   3.747 0.000179 ***
## V4A44          5.635e-01  7.428e-01   0.759 0.448091
## V4A45          1.782e-01  5.383e-01   0.331 0.740600
## V4A46         -1.800e-01  3.846e-01  -0.468 0.639890
## V4A48          2.133e+00  1.224e+00   1.742 0.081531 .
## V4A49          8.093e-01  3.226e-01   2.509 0.012110 *
## V5            -1.116e-04  4.092e-05  -2.726 0.006409 **
## V6A62          2.671e-01  2.739e-01   0.975 0.329488
## V6A63          4.271e-01  3.924e-01   1.089 0.276349
## V6A64          1.331e+00  5.038e-01   2.641 0.008255 **
## V6A65          9.677e-01  2.543e-01   3.805 0.000142 ***
## V8            -3.038e-01  8.429e-02  -3.605 0.000312 ***
## V9A92          1.243e-01  3.661e-01   0.340 0.734175
## V9A93          7.705e-01  3.600e-01   2.140 0.032331 *
## V9A94          2.786e-01  4.345e-01   0.641 0.521342
## V10A102       -5.323e-01  4.001e-01  -1.330 0.183402
## V10A103        1.022e+00  4.128e-01   2.476 0.013278 *
## V14A142        1.417e-01  4.007e-01   0.354 0.723700
## V14A143        6.536e-01  2.333e-01   2.802 0.005083 **
## V20A202        1.290e+00  6.220e-01   2.074 0.038064 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1221.73  on 999  degrees of freedom
## Residual deviance:  920.95  on 968  degrees of freedom
## AIC: 984.95
##
## Number of Fisher Scoring iterations: 5
```

By looking at the 0.05 and 0.1 levels for the p-values in the models, it has shown that the AIC at 984.95 is a slight decrease from the original at 993.82. The importance of the results have seem that there isn't much change at all just by using the results from deviance at all. Let's continue further with the analysis.

Looking for the R-Square in the Model

```
pR2(gce_log)

##          llh      llhNull            G2     McFadden          r2ML
## -447.9088927 -610.8643021   325.9108186    0.2667620     0.2781304
##         r2CU
##    0.3943548

pR2(gce0.05_log)
```

```
##          llh      llhNull           G2     McFadden         r2ML
## -460.4738644 -610.8643021  300.7808753    0.2461929    0.2597600
##         r2CU
##    0.3683078
```

```
pR2(gce0.1_log)
```

```
##          llh      llhNull           G2     McFadden         r2ML
## -460.4738644 -610.8643021  300.7808753    0.2461929    0.2597600
##         r2CU
##    0.3683078
```

For looking at the r-squared models, it has performed poorly for all three of them at 0.39. This has said that r-squared isn't the necessary best predictor.

Coefficients of the Model

```
coef(gce0.05_log)
```

```
##   (Intercept)         V1A12         V1A13         V1A14
V2
## -0.8649760417  0.4110493365  1.0685579853  1.7789448463
-0.0280539537
##         V3A31         V3A32         V3A33         V3A34
V4A41
##  0.1361367748  0.8588098332  0.9800032493  1.5872248890
1.5558030723
##        V4A410         V4A42         V4A43         V4A44
V4A45
##  1.5752471303  0.6611900907  0.8967984326  0.5634994460
0.1781948947
##         V4A46         V4A48         V4A49            V5
V6A62
## -0.1799533082  2.1326439263  0.8092803910 -0.0001115638
0.2670947555
##         V6A63         V6A64         V6A65            V8
V9A92
##  0.4271406756  1.3307166352  0.9676558814 -0.3038309609
0.1243044417
##         V9A93         V9A94       V10A102       V10A103
V14A142
##  0.7704937215  0.2786486137 -0.5322808104  1.0221620025
0.1416502625
##       V14A143       V20A202
##  0.6536033184  1.2901550120
```

```
sum(gce[,21])
```

```
## [1] 700
```

Comparing the Data by Using Half for Training and Testing

```
indexes = sample(1:nrow(gce), size=0.5*nrow(gce))
gce_train <- gce[indexes,] # Training Data
gce_test <- gce[-indexes,] # Testing Data
```

Using Logistic Regression on the Training Data

```
gce_train_half <- glm(V21 ~ ., family=binomial(link="logit"), data =
gce_train)
summary(gce_train_half)

##
## Call:
## glm(formula = V21 ~ ., family = binomial(link = "logit"), data =
gce_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5010  -0.5950   0.2890   0.6604   2.1947
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.806e+00  1.893e+00   0.954 0.339962
## V1A12       -3.790e-03  3.346e-01  -0.011 0.990961
## V1A13        9.900e-01  5.792e-01   1.709 0.087395 .
## V1A14        1.881e+00  3.488e-01   5.392 6.98e-08 ***
## V2          -1.699e-02  1.368e-02  -1.242 0.214161
## V3A31       -7.416e-02  8.863e-01  -0.084 0.933309
## V3A32        3.762e-01  7.039e-01   0.534 0.593053
## V3A33        8.745e-01  7.489e-01   1.168 0.242907
## V3A34        1.175e+00  7.048e-01   1.667 0.095611 .
## V4A41        2.064e+00  6.057e-01   3.407 0.000657 ***
## V4A410       8.229e-01  1.058e+00   0.778 0.436732
## V4A42        5.297e-02  4.016e-01   0.132 0.895085
## V4A43        6.805e-01  3.833e-01   1.775 0.075830 .
## V4A44       -2.078e+00  1.388e+00  -1.497 0.134318
## V4A45        9.203e-02  9.579e-01   0.096 0.923462
## V4A46        4.696e-01  5.482e-01   0.857 0.391638
## V4A48        1.027e+00  1.283e+00   0.800 0.423658
## V4A49        4.503e-01  5.183e-01   0.869 0.384902
## V5          -1.946e-04  6.497e-05  -2.996 0.002738 **
## V6A62        7.930e-01  4.363e-01   1.818 0.069130 .
## V6A63        6.518e-01  5.428e-01   1.201 0.229796
## V6A64        1.537e+00  9.079e-01   1.693 0.090391 .
## V6A65        4.502e-01  3.939e-01   1.143 0.253095
## V7A72       -1.394e+00  7.870e-01  -1.771 0.076585 .
## V7A73       -1.335e+00  7.540e-01  -1.771 0.076608 .
## V7A74       -7.798e-01  7.807e-01  -0.999 0.317851
## V7A75       -1.162e+00  7.600e-01  -1.529 0.126266
## V8          -4.617e-01  1.362e-01  -3.389 0.000702 ***
```

```
## V9A92         2.359e-01  5.338e-01   0.442 0.658490
## V9A93         1.280e+00  5.366e-01   2.385 0.017083 *
## V9A94         5.342e-01  6.415e-01   0.833 0.405040
## V10A102      -3.096e-01  6.718e-01  -0.461 0.644914
## V10A103       7.232e-01  5.863e-01   1.234 0.217377
## V11          -6.490e-02  1.301e-01  -0.499 0.617769
## V12A122      -3.565e-01  3.796e-01  -0.939 0.347701
## V12A123      -2.321e-01  3.714e-01  -0.625 0.532018
## V12A124      -1.887e+00  6.681e-01  -2.825 0.004735 **
## V13           3.054e-02  1.453e-02   2.101 0.035644 *
## V14A142      -1.172e-01  6.664e-01  -0.176 0.860418
## V14A143       9.448e-01  3.755e-01   2.516 0.011864 *
## V15A152       3.602e-01  3.429e-01   1.050 0.293512
## V15A153       6.519e-01  7.516e-01   0.867 0.385712
## V16          -5.774e-01  3.087e-01  -1.870 0.061428 .
## V17A172      -6.480e-01  1.473e+00  -0.440 0.659872
## V17A173      -4.087e-01  1.453e+00  -0.281 0.778456
## V17A174      -3.691e-01  1.437e+00  -0.257 0.797320
## V18          -5.023e-01  3.875e-01  -1.296 0.194900
## V19A192       4.679e-01  2.992e-01   1.564 0.117829
## V20A202       1.293e+00  7.043e-01   1.836 0.066297 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 609.16  on 499  degrees of freedom
## Residual deviance: 411.41  on 451  degrees of freedom
## AIC: 509.41
##
## Number of Fisher Scoring iterations: 6
```

Model 1 and Looking at the 0.1 p-value

```
gce_train_half_one <- glm(V21 ~ V1+V2+V3+V4+V6+V8+V10+V12+V14+V16,
family=binomial(link="logit"), data = gce_train)
summary(gce_train_half_one)

##
## Call:
## glm(formula = V21 ~ V1 + V2 + V3 + V4 + V6 + V8 + V10 + V12 +
##     V14 + V16, family = binomial(link = "logit"), data = gce_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5234  -0.7646   0.3685   0.7248   2.3356
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.66453    0.95765   0.694 0.487732
## V1A12        0.06224    0.30329   0.205 0.837408
```

```
## V1A13        1.04559    0.53633   1.950 0.051233 .
## V1A14        1.84050    0.32590   5.647 1.63e-08 ***
## V2          -0.03733    0.01037  -3.601 0.000317 ***
## V3A31        0.03906    0.80790   0.048 0.961438
## V3A32        0.46134    0.60902   0.758 0.448739
## V3A33        0.90235    0.66800   1.351 0.176752
## V3A34        1.45333    0.61909   2.348 0.018898 *
## V4A41        1.91182    0.54881   3.484 0.000495 ***
## V4A410       0.91338    0.87672   1.042 0.297500
## V4A42       -0.11813    0.35604  -0.332 0.740058
## V4A43        0.56950    0.34729   1.640 0.101040
## V4A44       -1.29498    1.26674  -1.022 0.306642
## V4A45        0.29594    0.81743   0.362 0.717324
## V4A46        0.18333    0.52045   0.352 0.724653
## V4A48        0.69567    1.21410   0.573 0.566649
## V4A49        0.52518    0.47710   1.101 0.270994
## V6A62        0.57980    0.38855   1.492 0.135639
## V6A63        0.65586    0.50040   1.311 0.189972
## V6A64        1.15571    0.80165   1.442 0.149399
## V6A65        0.57172    0.36501   1.566 0.117276
## V8          -0.22497    0.10866  -2.070 0.038412 *
## V10A102     -0.64913    0.61004  -1.064 0.287290
## V10A103      0.84931    0.54916   1.547 0.121970
## V12A122     -0.32792    0.34484  -0.951 0.341643
## V12A123     -0.30886    0.33362  -0.926 0.354570
## V12A124     -1.01214    0.43928  -2.304 0.021219 *
## V14A142      0.12218    0.63372   0.193 0.847115
## V14A143      0.80415    0.35015   2.297 0.021644 *
## V16         -0.44896    0.27275  -1.646 0.099750 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 609.16  on 499  degrees of freedom
## Residual deviance: 454.66  on 469  degrees of freedom
## AIC: 516.66
##
## Number of Fisher Scoring iterations: 5
```

After looking at the first model, there are three significant factors on the equation, which are V1A14, V2, and V4A41.

Model 2 at the 0.1 p-value

```
gce_train_half_two <- glm(V21 ~ V1+V2+V3+V4+V6+V10+V14+V16,
family=binomial(link="logit"), data = gce_train)
summary(gce_train_half_two)

##
## Call:
```

```
## glm(formula = V21 ~ V1 + V2 + V3 + V4 + V6 + V10 + V14 + V16,
##     family = binomial(link = "logit"), data = gce_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7037  -0.8315   0.3784   0.7445   2.0258
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.486593   0.853825  -0.570 0.568747
## V1A12        0.207466   0.289207   0.717 0.473152
## V1A13        1.131365   0.532442   2.125 0.033598 *
## V1A14        1.897703   0.319668   5.936 2.91e-09 ***
## V2          -0.043251   0.009694  -4.462 8.14e-06 ***
## V3A31        0.029827   0.786937   0.038 0.969765
## V3A32        0.498937   0.594055   0.840 0.400974
## V3A33        0.907492   0.655210   1.385 0.166040
## V3A34        1.451394   0.607869   2.388 0.016955 *
## V4A41        1.752438   0.520431   3.367 0.000759 ***
## V4A410       0.944230   0.901698   1.047 0.295022
## V4A42       -0.019571   0.348062  -0.056 0.955160
## V4A43        0.530618   0.337015   1.574 0.115380
## V4A44       -1.277560   1.241625  -1.029 0.303507
## V4A45        0.068638   0.796917   0.086 0.931364
## V4A46       -0.129230   0.498628  -0.259 0.795503
## V4A48        0.682818   1.233511   0.554 0.579883
## V4A49        0.656455   0.468924   1.400 0.161538
## V6A62        0.384922   0.371586   1.036 0.300254
## V6A63        0.662058   0.494319   1.339 0.180462
## V6A64        1.216326   0.792246   1.535 0.124713
## V6A65        0.588550   0.356634   1.650 0.098884 .
## V10A102     -0.713770   0.596221  -1.197 0.231246
## V10A103      1.026558   0.537100   1.911 0.055967 .
## V14A142      0.278350   0.613977   0.453 0.650292
## V14A143      0.898862   0.339618   2.647 0.008128 **
## V16         -0.362713   0.265488  -1.366 0.171872
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 609.16  on 499  degrees of freedom
## Residual deviance: 465.08  on 473  degrees of freedom
## AIC: 519.08
##
## Number of Fisher Scoring iterations: 5
```

```
pR2(gce_train_half_two)
```

```
##          llh      llhNull           G2     McFadden         r2ML 
## -232.5392682 -304.5800852  144.0816341    0.2365250    0.2503608 
##         r2CU 
##    0.3554880
```

Using a Different Threshold at 50%

```r
gce_fit <- fitted.values(gce_train_half_two)
tf <- rep(0,500)

for (j in 1:500){
  if(gce_fit[j] >= 0.5) tf[j] <- 1
}

CrossTable(gce_train$V21, tf, digits=1, prop.r=F, prop.t=F,
prop.chisq=F, chisq=F, data=gce_train)

## 
## 
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
## 
## 
## Total Observations in Table:  500 
## 
## 
##              | tf 
## gce_train$V21 |         0 |         1 | Row Total | 
## -------------|-----------|-----------|-----------|
##            0 |        76 |        73 |       149 | 
##              |       0.7 |       0.2 |           | 
## -------------|-----------|-----------|-----------|
##            1 |        33 |       318 |       351 | 
##              |       0.3 |       0.8 |           | 
## -------------|-----------|-----------|-----------|
##  Column Total |       109 |       391 |       500 | 
##              |       0.2 |       0.8 |           | 
## -------------|-----------|-----------|-----------|
## 
## 
```

Total Cost

```r
gce_tc50 <- 76* 5 + 33 * 1
gce_tc50

## [1] 413
```

Another Threshold Scenario, Concentrating on 70%

```
gce_fit <- fitted.values(gce_train_half_two)
tf <- rep(0,500)

for (j in 1:500){
  if(gce_fit[j] >= 0.7) tf[j] <- 1
}

CrossTable(gce_train$V21, tf, digits=1, prop.r=F, prop.t=F,
prop.chisq=F, chisq=F, data=gce_train)

##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Col Total |
## |-------------------------|
##
##
## Total Observations in Table:  500
##
##
##              | tf
## gce_train$V21 |         0 |         1 | Row Total |
## --------------|-----------|-----------|-----------|
##             0 |       112 |        37 |       149 |
##               |       0.5 |       0.1 |           |
## --------------|-----------|-----------|-----------|
##             1 |        96 |       255 |       351 |
##               |       0.5 |       0.9 |           |
## --------------|-----------|-----------|-----------|
##  Column Total |       208 |       292 |       500 |
##               |       0.4 |       0.6 |           |
## --------------|-----------|-----------|-----------|
##
##
```

Total Cost Part 2

```
gce_tc70 <- 37* 5 + 96 * 1
gce_tc70

## [1] 281
```

By doing the analyses for the different thresholds on the datasets between the 50-70% thresholds, it has been determined that the 50% threshold has a higher total cost than the 70% threshold by looking at the tables that are given. When the 50% threshold was used, it seems that there are more people woulkd be accepted for credit at 83% vs. at 70% where it was at 56%.