Week 4 Homework

loading packages and setting seed for homework

```
library('tidyverse')
library('randomForest')
library('rpart')
library('ROCR')

#Set seed for reproducibility
set.seed(156)
```

Question 1

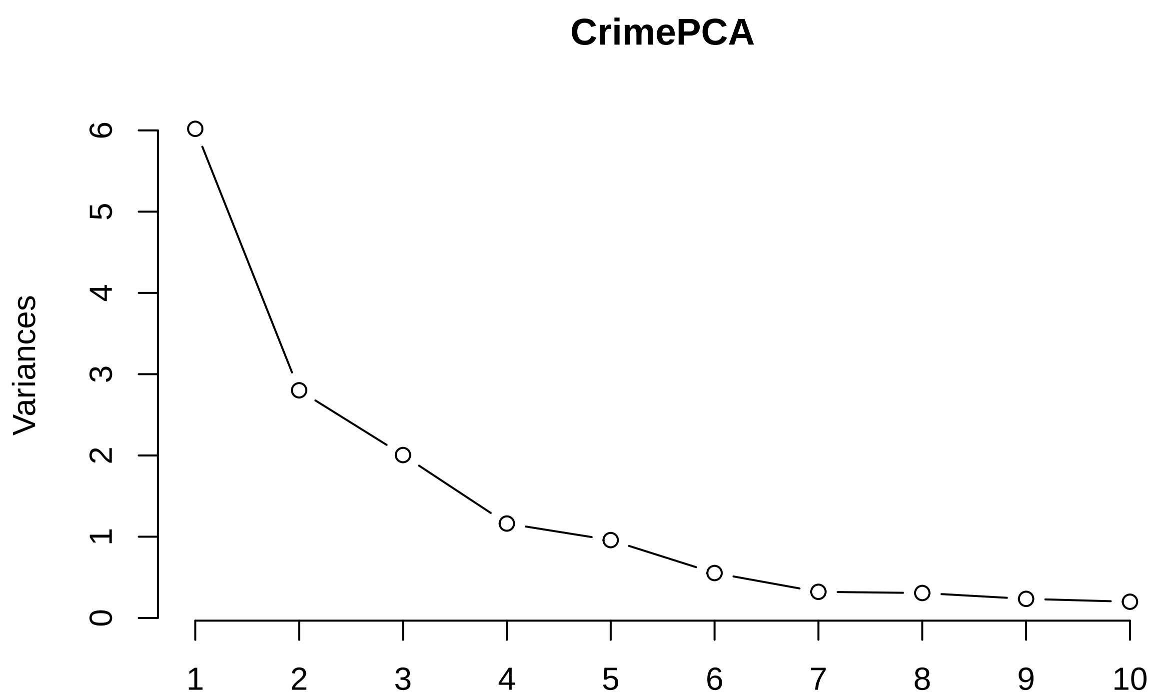
apply Principal Component Analysis and then create a regression model using the first 4 principal components. Specify your new model in terms of the original variables (not the principal components), and compare its quality to that of your solution to Homework 3 Question 4.

The results I obtained indicate that my original model from Week 3 was a more accurate model.

```
#Crime data file
crimeData <- read.table("http://www.statsci.org/data/general/uscrime.txt", header = TRUE)

#Performing PCA on original crime dataset, removing response variable.
CrimePCA <- prcomp(crimeData[,-16], center = TRUE, scale. = TRUE)

#The first four principal components account for 79% of the variance in the data.
# This can be seen with the screeplot and summary.
summary(CrimePCA)
```

```
## 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
```

```
screeplot(CrimePCA, type = "lines")
```




```r
#rotation matrix of first 4 Principal Components
CrimePCA$rotation[,1:4]
```

```
##                PC1         PC2           PC3         PC4
## M      -0.30371194  0.06280357  0.1724199946 -0.02035537
## So     -0.33088129 -0.15837219  0.0155433104  0.29247181
## Ed      0.33962148  0.21461152  0.0677396249  0.07974375
## Po1     0.30863412 -0.26981761  0.0506458161  0.33325059
## Po2     0.31099285 -0.26396300  0.0530651173  0.35192809
## LF      0.17617757  0.31943042  0.2715301768 -0.14326529
## M.F     0.11638221  0.39434428 -0.2031621598  0.01048029
## Pop     0.11307836 -0.46723456  0.0770210971 -0.03210513
## NW     -0.29358647 -0.22801119  0.0788156621  0.23925971
## U1      0.04050137  0.00807439 -0.6590290980 -0.18279096
## U2      0.01812228 -0.27971336 -0.5785006293 -0.06889312
## Wealth  0.37970331 -0.07718862  0.0100647664  0.11781752
## Ineq   -0.36579778 -0.02752240 -0.0002944563 -0.08066612
## Prob   -0.25888661  0.15831708 -0.1176726436  0.49303389
## Time   -0.02062867 -0.38014836  0.2235664632 -0.54059002
```

```r
#Checking to ensure that principal components are orthogonal -
# perfect correlation across the diagonal confirms this.
cor(CrimePCA$x[,1:4])
```

```
##               PC1           PC2           PC3          PC4
## PC1  1.000000e+00 -1.273307e-16 -1.825724e-16 2.298165e-16
## PC2 -1.273307e-16  1.000000e+00 -5.694249e-16 3.269637e-16
```

```
## PC3 -1.825724e-16 -5.694249e-16  1.000000e+00 1.177395e-16
## PC4  2.298165e-16  3.269637e-16  1.177395e-16 1.000000e+00
```

```r
# creating dataframe with principal components and response variable
CrimePCAData <- cbind(crimeData[,16],data.frame(CrimePCA$x[,1:4]))
colnames(CrimePCAData)[1] <- 'Crime'

#creating linear model using 1st 4 principal components -
# R squared value is only 0.3. PC3 and PC4 seem to be statistically insignificant to the model
# which tells me that the first 4 principal components do not produce a
# better result from the original model.
CrimePCA.lm <- lm(Crime ~., data = CrimePCAData)
summary(CrimePCA.lm)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = CrimePCAData)
##
## 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
```

```r
#converting rotation matrix to model coefficients
betas <- CrimePCA$rotation[,1:4] %*% CrimePCA.lm$coefficients[-1]
colnames(betas)[1] <- 'coefficients'
betas
```

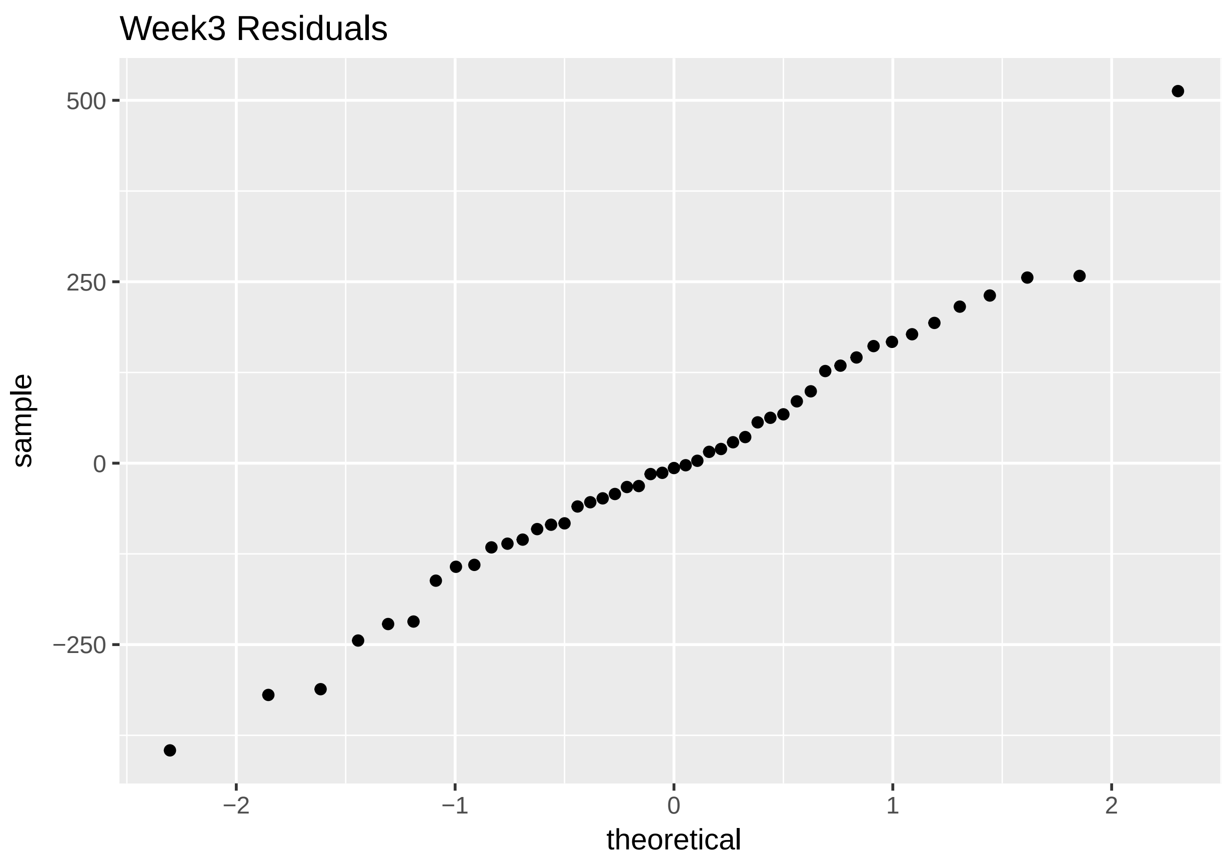
```
##        coefficients
## M        -21.277963
## So        10.223091
## Ed        14.352610
## Po1       63.456426
## Po2       64.557974
## LF       -14.005349
## M.F      -24.437572
## Pop       39.830667
## NW        15.434545
## U1       -27.222281
## U2         1.425902
## Wealth    38.607855
## Ineq     -27.536348
## Prob       3.295707
```

```
## Time      -6.612616
```

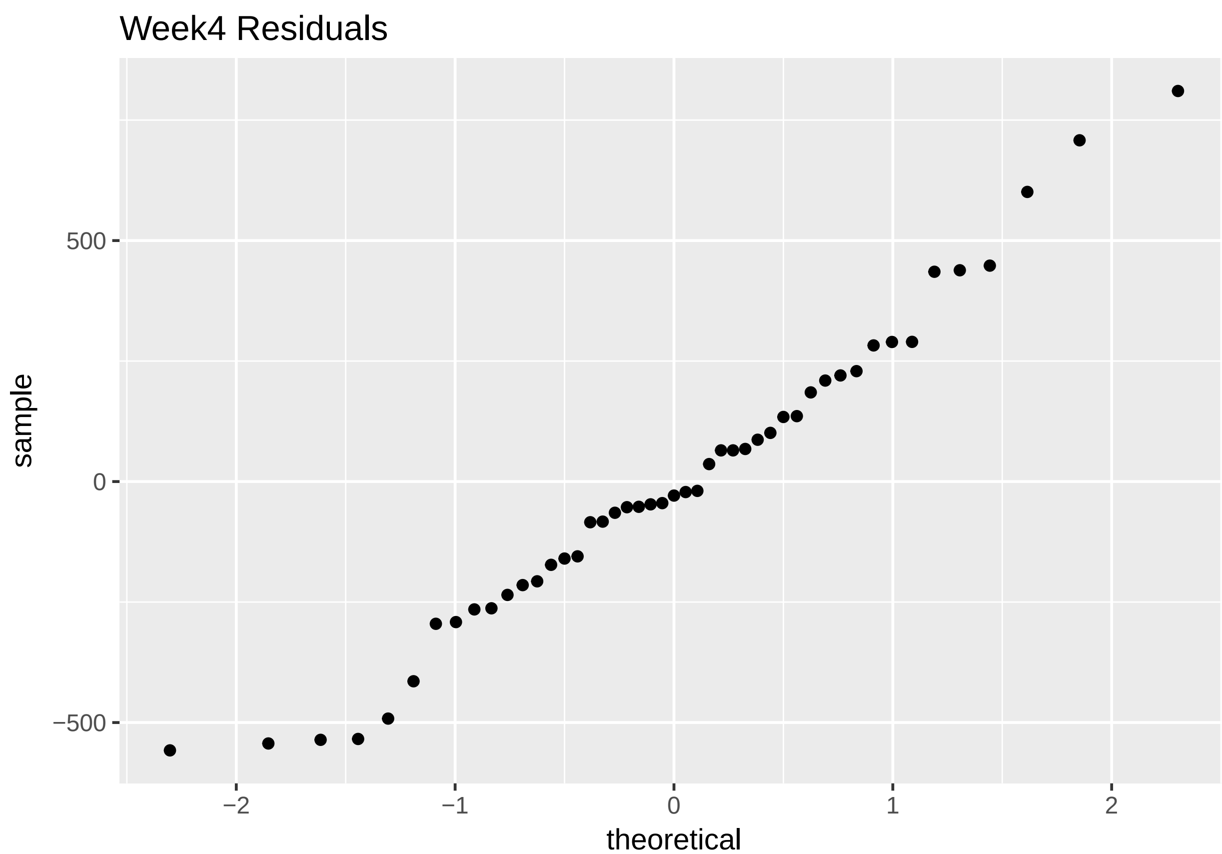
```r
#original linear model from Week 3
crimeModel <- lm(Crime ~ ., data = crimeData)

#using ANOVA to compare the PCA model vs the original model.
# This shows the P value being 7.857e-06 which is less than .05 so
# we can reject the null hypothesis. The models are different.
anova(CrimePCA.lm,crimeModel)
```

```
## Analysis of Variance Table
##
## Model 1: Crime ~ PC1 + PC2 + PC3 + PC4
## Model 2: Crime ~ M + So + Ed + Po1 + Po2 + LF + M.F + Pop + NW + U1 +
##     U2 + Wealth + Ineq + Prob + Time
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     42 4753950
## 2     31 1354946 11   3399004 7.0697 7.857e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
crimeData$predictedWeek3 <-  predict(crimeModel)
crimeData$predictedWeek4 <- predict(CrimePCA.lm)

# Obtain  residual values
crimeData$residualsWeek3 <- residuals(crimeModel)
crimeData$residualsWeek4 <- residuals(CrimePCA.lm)

#Creating residual df for plotitng
modelResiduals <- data.frame(data=(cbind(residuals(crimeModel),residuals(CrimePCA.lm))))
colnames(modelResiduals) <- c('Week3', 'Week4')

#qqplots of the residuals of both models show they are fairly normally distributed.
modelResiduals %>%
  ggplot(aes(sample=modelResiduals$Week3)) +
  stat_qq() +
  labs(title = "Week3 Residuals")
```




```
modelResiduals %>%
  ggplot(aes(sample=modelResiduals$Week4)) +
  stat_qq() +
  labs(title = "Week4 Residuals")
```



Question 2

Using the same crime data set as in Homework 3 Question 4, find the best model you can using (a) a regression tree model, and (b) a random forest model. For each model, describe one or two qualitative takeaways you get from analyzing the results

Rpart Insights: The variable importance from the model is telling me that Po1 and Po2 are both very important, but very similar. Meaning only one is actually needed. Wealth and Ineq are also very similar but importance. Prob and M are the next two most importance variables. RPart chose very similar variables as most important compared to randomForest.

Random Forest Insights: The plot showing MSE tells me that Po1 and Po2 are very similar in importance so only one is really needed in the model. NW is the next most important variable followed by Prob, then Wealth. Node purity also tells me that Po1 and Po2 are very similar in importance, followed by Prob, and Wealth and NW are very similar, followed by Pop. Both of these show that most of the variance can be explained with Po1, Prob, Wealth, and NW. It also shows that there are many variables that are similar to each other in importance and we could throw some out if need be while still maintaining the same level of accuracy.

My random forest model is able to predict on average 73% of the actual crime values.

Another insight that I take from these models is that they show collinearity of predictors. This is a very useful tool which can help reduce dimensionality in models. The collinearity that is being shown in these models is also what PCA and a VIF test are showing as well. This is a universal takeaway from the tree family models that can be applied to modeling the same dataset with other methods.

```r
CrimeData.2 <- read.table("http://www.statsci.org/data/general/uscrime.txt", header = TRUE)
crimeDataRF <- CrimeData.2
crimeDataRpart <- CrimeData.2

#(b) regression tree model
Crime.Rpart <- rpart(Crime ~ ., data = crimeDataRpart, method = "anova")
summary(Crime.Rpart)
```

```
## Call:
## rpart(formula = Crime ~ ., data = crimeDataRpart, method = "anova")
##   n= 47
##
##           CP nsplit rel error    xerror      xstd
## 1 0.36296293      0 1.0000000 1.0085866 0.2534839
## 2 0.14814320      1 0.6370371 0.8648446 0.1874900
## 3 0.05173165      2 0.4888939 1.0148268 0.2337991
## 4 0.01000000      3 0.4371622 0.9076756 0.2252342
##
## Variable importance
##    Po1    Po2 Wealth   Ineq   Prob      M     NW    Pop   Time     Ed
##     17     17     11     11     10     10      9      5      4      4
##     LF     So
##      1      1
##
## Node number 1: 47 observations,    complexity param=0.3629629
##   mean=905.0851, MSE=146402.7
##   left son=2 (23 obs) right son=3 (24 obs)
##   Primary splits:
##       Po1    < 7.65      to the left,  improve=0.3629629, (0 missing)
##       Po2    < 7.2       to the left,  improve=0.3629629, (0 missing)
##       Prob   < 0.0418485 to the right, improve=0.3217700, (0 missing)
##       NW     < 7.65      to the left,  improve=0.2356621, (0 missing)
##       Wealth < 6240      to the left,  improve=0.2002403, (0 missing)
##   Surrogate splits:
##       Po2    < 7.2       to the left,  agree=1.000, adj=1.000, (0 split)
##       Wealth < 5330      to the left,  agree=0.830, adj=0.652, (0 split)
##       Prob   < 0.043598  to the right, agree=0.809, adj=0.609, (0 split)
##       M      < 13.25     to the right, agree=0.745, adj=0.478, (0 split)
##       Ineq   < 17.15     to the right, agree=0.745, adj=0.478, (0 split)
##
## Node number 2: 23 observations,    complexity param=0.05173165
##   mean=669.6087, MSE=33880.15
##   left son=4 (12 obs) right son=5 (11 obs)
##   Primary splits:
##       Pop < 22.5      to the left,  improve=0.4568043, (0 missing)
##       M   < 14.5      to the left,  improve=0.3931567, (0 missing)
##       NW  < 5.4       to the left,  improve=0.3184074, (0 missing)
##       Po1 < 5.75      to the left,  improve=0.2310098, (0 missing)
##       U1  < 0.093     to the right, improve=0.2119062, (0 missing)
##   Surrogate splits:
##       NW   < 5.4       to the left,  agree=0.826, adj=0.636, (0 split)
##       M    < 14.5      to the left,  agree=0.783, adj=0.545, (0 split)
##       Time < 22.30055  to the left,  agree=0.783, adj=0.545, (0 split)
##       So   < 0.5       to the left,  agree=0.739, adj=0.455, (0 split)
```

```
##       Ed   < 10.85     to the right, agree=0.739, adj=0.455, (0 split)
##
## Node number 3: 24 observations,    complexity param=0.1481432
##   mean=1130.75, MSE=150173.4
##   left son=6 (10 obs) right son=7 (14 obs)
##   Primary splits:
##       NW   < 7.65      to the left,  improve=0.2828293, (0 missing)
##       M    < 13.05     to the left,  improve=0.2714159, (0 missing)
##       Time < 21.9001   to the left,  improve=0.2060170, (0 missing)
##       M.F  < 99.2      to the left,  improve=0.1703438, (0 missing)
##       Po1  < 10.75     to the left,  improve=0.1659433, (0 missing)
##   Surrogate splits:
##       Ed   < 11.45     to the right, agree=0.750, adj=0.4, (0 split)
##       Ineq < 16.25     to the left,  agree=0.750, adj=0.4, (0 split)
##       Time < 21.9001   to the left,  agree=0.750, adj=0.4, (0 split)
##       Pop  < 30        to the left,  agree=0.708, adj=0.3, (0 split)
##       LF   < 0.5885    to the right, agree=0.667, adj=0.2, (0 split)
##
## Node number 4: 12 observations
##   mean=550.5, MSE=20317.58
##
## Node number 5: 11 observations
##   mean=799.5455, MSE=16315.52
##
## Node number 6: 10 observations
##   mean=886.9, MSE=55757.49
##
## Node number 7: 14 observations
##   mean=1304.929, MSE=144801.8
```

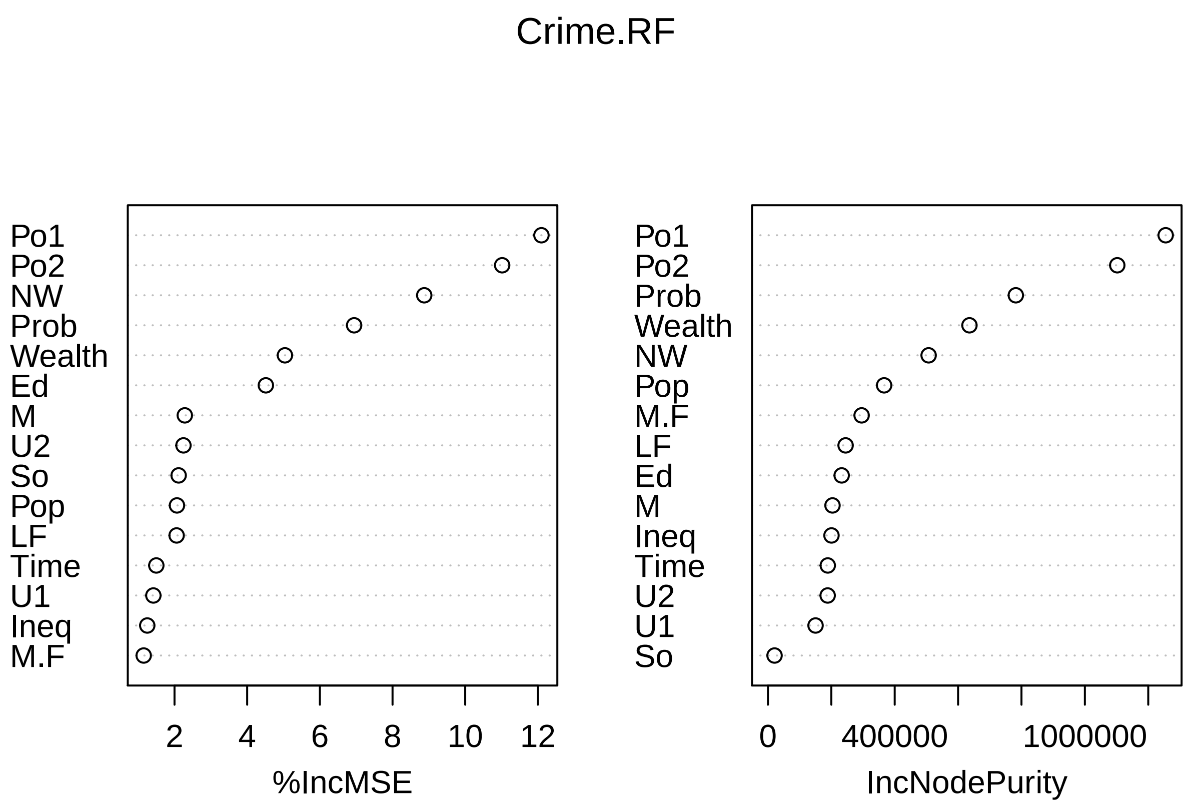
```
#variable importance
Crime.Rpart$variable.importance
```

```
##       Po1       Po2    Wealth      Ineq      Prob         M        NW
## 2497521.7 2497521.7 1628818.5 1602212.0 1520230.6 1388627.8 1245883.8
##       Pop      Time        Ed        LF        So
##  661770.6  601906.0  569545.9  203872.5  161800.8
```

```
#-----------------------------------------------------------------------------
#(a) RandomForest Model
Crime.RF <- randomForest(Crime ~ ., data = crimeDataRpart, importance = TRUE)
Crime.RF
```

```
##
## Call:
##  randomForest(formula = Crime ~ ., data = crimeDataRpart, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##           Mean of squared residuals: 84928.76
##                     % Var explained: 41.99
```

```r
#plotting the importance of variables in the Randomforest model.
varImpPlot(Crime.RF)
```




```r
RFpred <- predict(Crime.RF)

crimeDataRF[,17] <- predict(Crime.RF)
colnames(crimeDataRF)[17] <- 'RFprediction'

#showing predictions and percent correct to prediction
crimeDataRF$predictionVariance <- abs(crimeDataRF$RFprediction - crimeDataRF$Crime)
crimeDataRF$predictionPrcntCorrect <- 1 - (round(crimeDataRF$predictionVariance / crimeDataRF$Crime,2))

#This tells me that my random Forest is on average 73% correct in predicting crime.
mean(crimeDataRF$predictionPrcntCorrect)
```

```
## [1] 0.7374468
```

Question 3

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.

Answer:

Logistic regression would be useful at work to determine the probability of successfully winning a bank charge back. Predictors that would be good in this model would be: PrincipalAmount, CustomerTenure, BIN, OrderingMethod, ReasonForChargeBack

Question 4

use logistic regression to find a good predictive model for whether credit applicants are good credit risks or not. Show your model (factors used and their coefficients), the software output, and the quality of fit.

```r
#German Credit Data
germanCreditData <-as.data.frame(read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/s

#v21 is the response variable, it is supposed to be logical.
# Converting to 0 and 1 rather than 1 and 2. new data: 0 is bad 1 is good
germanCreditData$V21 <- as.integer(ifelse(germanCreditData$V21 == 2,1,0))

#Creating the initial logisitc model
germanGLM <- glm(V21 ~ .,family=binomial(link='logit'), data = germanCreditData)

summary(germanGLM)
```

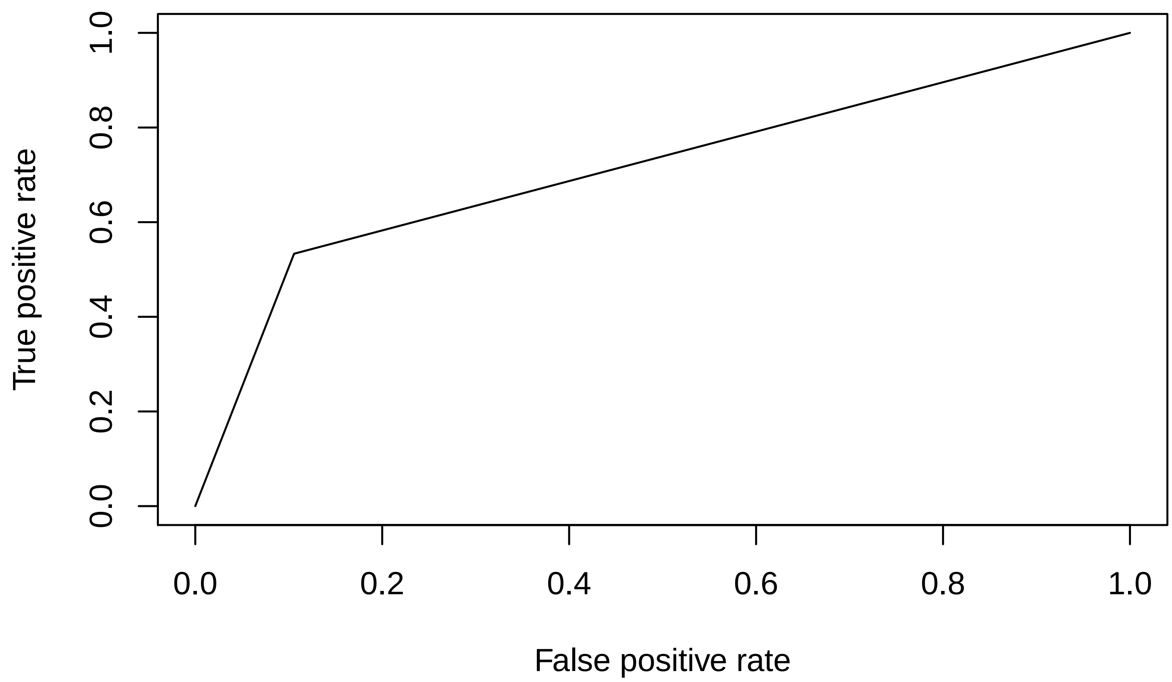
```
##
## Call:
## glm(formula = V21 ~ ., family = binomial(link = "logit"), data = germanCreditData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3410  -0.6994  -0.3752   0.7095   2.6116
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  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
```

```r
Pgerman <- predict(germanGLM, newdata = germanCreditData[,-21], type="response")
#converting probabilities to logical responses
Pgerman <- ifelse(Pgerman > 0.5,1,0)

#prediction accuracy of glm model
predVariance <- mean(Pgerman != germanCreditData[,21])
cat('Prediction Accuracy:',(1-predVariance))
```

```
## Prediction Accuracy: 0.786
```

```r
#Creating ROC Curve and plotting using the ROCR package
predictions <- ROCR::prediction(Pgerman, germanCreditData$V21)
perf <- performance(predictions, measure = "tpr", x.measure = "fpr")
plot(perf)
```




```r
auc <- performance(predictions, measure = "auc")
#grabbing only the auc value
auc <- auc@y.values[[1]]
#This tells me that the model is doing well as the auc number is greater than .5 and near 1
auc
```

```
## [1] 0.7138095
```