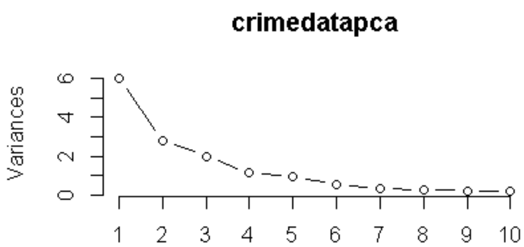
HW 4 (Included my code and comments in #)

Question 9.1

Using the same crime data set uscrime.txt as in Question 8.2, apply Principal Component Analysis and then create a regression model using the first few principal components. Specify your new model in terms of the original variables (not the principal components), and compare its quality to that of your solution to Question 8.2. You can use the R function prcomp for PCA.

#Read Table. Since there are 15 predictor variables, there should be 15 principle components.
crimedata <- read.table("9.1uscrimeSummer2018.txt", header = TRUE, stringsAsFactors = FALSE)

#Perform Principle Component Analysis. PCA tells what is the effect of a variable on a whole data set.
crimedatapca <- prcomp(crimedata[,1:15], center = TRUE, scale. = TRUE)
plot(crimedatapca, type = "l") #Screen Plot. Shows variance explained by each PC's. The first pc1 accounts for the most variation in the original data.



summary(crimedatapca) #How much of the variance is explained by each PC's.

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

#Make regression models based on PCAs to predict crime.
pccrime <- cbind(crimedatapca$x[,1:10], crimedata[,16])
pccrime
model2 <- lm(pccrime[,11]~.,data = as.data.frame(pccrime[,1:10]))
summary(model2)
#Using the first 10 PCs, I get a R2 of .6963 and adjusted R2 of .6119

```
Residuals:
    Min      1Q  Median      3Q     Max
-428.85 -146.39    9.56  148.94  424.17

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   905.09      35.14  25.753  < 2e-16 ***
PC1            65.22      14.48   4.504 6.77e-05 ***
PC2           -70.08      21.22  -3.302  0.00217 **
PC3            25.19      25.09   1.004  0.32198
PC4            69.45      32.95   2.107  0.04211 *
PC5          -229.04      36.29  -6.312 2.67e-07 ***
PC6           -60.21      47.76  -1.261  0.21553
PC7           117.26      62.62   1.872  0.06928 .
PC8            28.72      64.07   0.448  0.65670
PC9           -37.18      73.26  -0.507  0.61492
PC10           56.32      79.46   0.709  0.48303
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 240.9 on 36 degrees of freedom
Multiple R-squared:  0.6963,	Adjusted R-squared:  0.6119
F-statistic: 8.253 on 10 and 36 DF,  p-value: 9.127e-07
```

#scaled * (x-mean)/sd =unscaled * x+b
#Below are the coefficients in terms of original variable
coeff2 <-
model2$coefficients[2:length(model2$coefficients)]%*%t(crimedatapca$rotation[,1:(length(model2$coefficients)-1)])

```
            M       So       Ed      Po1      Po2       LF      M.F      Pop      NW        U1       U2
[1,] 46.95501 101.3146 33.88377 118.5441 120.1252 30.92128 118.4317 24.86952 84.9771 -12.27977 22.87038
       Wealth     Ineq      Prob     Time
[1,] 32.60724 51.88229 -142.0114 -17.8218
```

#Below are the unscaled coefficients and intercepts.
intercept <- model2$coefficients[1] - sum(coeff2*sapply(crimedata[,1:15],mean)/sapply(crimedata[,1:15], sd))
coeff3 <- coeff2/sapply(crimedata[,1:15], sd)

```
            M       So       Ed      Po1      Po2       LF     M.F       Pop       NW         U1       U2
[1,] 37.36186 211.5236 30.28852 39.88836 42.96121 765.1546 40.1908 0.6532373 8.263938 -681.1206 27.08012
         Wealth     Ineq      Prob      Time
[1,] 0.03379305 13.00436 -6245.838 -2.514755
```

#Compute R2 and adjusted R2
SSE = sum((estimates - crimedata[,16])^2)
SStot = sum((crimedata[,16] - mean(crimedata[,16]))^2)
R2 = 1-SSE/SStot #.681
R2adjusted = (1-R2)*10/(nrow(crimedata)-10-1) #.065
#PCA model seems much worse than a non-pca model. In the original regression model from 8.2, the R2 value is .803 but PCA gives a result much lower than that which is .681.This may be due to overfitting.

Question 10.1

Using the same crime data set uscrime.txt as in Questions 8.2 and 9.1, find the best model you can using a regression tree model and a random forest model.

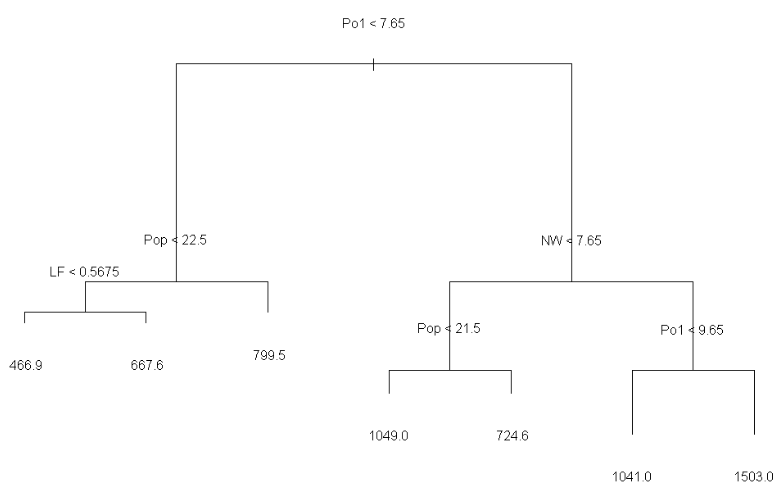
#Fit regression tree function to US crime data
tree.data <- tree(Crime~., data=data)
#Only 4 predictors were used in the construction of the tree. The deviance value is a measure of how significant the variables are. There are 7 terminal nodes.

summary(tree.data)

```
Regression tree:
tree(formula = Crime ~ ., data = data)
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
```

#More information on how the tree was split
tree.data$frame
#Plotting the regression tree yields us the below graph
plot(tree.data)
text(tree.data)

```
      var  n        dev      yval splits.cutleft splits.cutright
1     Po1 47 6880927.66  905.0851          <7.65           >7.65
2     Pop 23  779243.48  669.6087          <22.5           >22.5
4      LF 12  243811.00  550.5000        <0.5675         >0.5675
8  <leaf>  7   48518.86  466.8571
9  <leaf>  5   77757.20  667.6000
5  <leaf> 11  179470.73  799.5455
3      NW 24 3604162.50 1130.7500          <7.65           >7.65
6     Pop 10  557574.90  886.9000          <21.5           >21.5
12 <leaf>  5  146390.80 1049.2000
13 <leaf>  5  147771.20  724.6000
7     Po1 14 2027224.93 1304.9286          <9.65           >9.65
14 <leaf>  6  170828.00 1041.0000
15 <leaf>  8 1124984.88 1502.8750
```



#Calculate R2
yhat <- predict(tree.data)
ssres <- sum((yhat-data$Crime)^2)
sstot <- sum((data$Crime -mean(data$Crime))^2)
R2 <- 1-(ssres/sstot)
R2 #.724

#Under regression tree Po1 is good predictor of crime.
#Grow random tree and set the number of predictors that want to consider at each split of tree
numperd <- 4 #How many variables to split or branch
rf.data <- randomForest(Crime~., data = data, mtry = numperd, importance = TRUE)
rf.data

```
Call:
 randomForest(formula = Crime ~ ., data = data, mtry = numperd,      importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 84174.71
                    % Var explained: 42.5
```

#Calculate R2
yhat <- predict(rf.data)
ssres <- sum((yhat-data$Crime)^2)
sstot <- sum((data$Crime -mean(data$Crime))^2)
R2 <- 1-(ssres/sstot)
R2 #.416 (Same as %Var explained)

importance(rf.data) #In both metrics higer values mean it's more important. Just like regression tree, random forest is also a very important predictor of crime. It gives a better predictive quality for data points where Po1<7.65 than it is for >7.65. unlike regression tree, random forest does have predictive value even for Po1>7.65.

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.

Before launching any new product, it is important to do proper market research to get an understanding on the demand for your product. Let's say that you are trying to launch a new vegan organic shampoo product. A logistic regression model could be used to identify if a customer would be willing to buy your product. Some possible predictors could be: customer's age, income, attitude towards organic/vegan products, gender, and average yearly spend on relevant products.

Question 10.3

1. Using the GermanCredit data set germancredit.txt from http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/german / (description at http://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29), 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. You can use the glm function in R. To get a logistic regression (logit) model on data where the response is either zero or one, use family=binomial(link="logit") in your glm function call.

```
data <- read.table("10.3germancreditSummer2018.txt", sep = " ")
str(data)

#Binomial family of glm recognises 0 and 1 as the calssification values, 1 and 2 to 0 and 1 for the response variable
data$V21[data$V21==1]<-0
data$V21[data$V21==2]<-1

#0 is good and 1 is bad
#Part 1: Use all the variables
reg <- glm(V21~., family = binomial(link = "logit"), data=data)
summary(reg)

#Here are all the coefficients for the model with AIC 993.82
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
```

2. Because the model gives a result between 0 and 1, it requires setting a threshold probability to separate between “good” and “bad” answers. In this data set, they estimate that incorrectly identifying a bad customer as good, is 5 times worse than incorrectly classifying a good customer as bad. Determine a good threshold probability based on your model

A good threshold to use for this model would be .5.

```
confusionmatrix<-(table(data$V21, pred > 0.5))
confusionmatrix
#   FALSE TRUE
# 0  623   77
# 1  149  151
```