Mixture of Opensource datasets

Vaibhav Walvekar

Due: Tuesday, December 13, 2016

```
#Libraries Required for setup
library(tidyverse)
library(MASS)
library(ISLR)
library(class)
library(tree)
library(gbm)
library(randomForest)
library(AER)
library(bestglm)
library(car)
library(data.table)
library(boot)
```

1. In this problem we will use data about infidelitys, known as the Fair's Affairs dataset. The Affairs dataset is available as part of the AER package in R. This data comes from a survey conducted by Psychology Today in 1969, see Greene (2003) and Fair (1978) for more information. The dataset contains various self-reported characteristics of 601 participants, including how often the respondent engaged in extramarital sexual intercourse during the past year, as well as their gender, age, year married, whether they had children, their religiousness (on a 5-point scale, from 1=anti to 5=very), education, occupation (Hillinghead 7-point classification with reverse numbering), and a numeric self-rating of their marriage (from 1=very unhappy to 5=very happy).

(a) Describe the participants. Use descriptive, summarization, and exploratory techniques to describe the participants in the study. For example, what proportion of respondents are female? What is the average age of respondents?

```
#Reading Affairs Dataset
data(Affairs)
affairs_data <- Affairs

#Exploring ffairs Dataset
dim(affairs_data)
```

```
## [1] 601   9
```

```
sapply (affairs_data, class)
```

```
##       affairs        gender           age  yearsmarried      children
##     "numeric"      "factor"     "numeric"     "numeric"      "factor"
## religiousness     education    occupation        rating
##     "integer"     "numeric"     "integer"     "integer"
```

```
summary(affairs_data)
```

```
##     affairs          gender         age         yearsmarried    children
##  Min.   : 0.000   female:315   Min.   :17.50   Min.   : 0.125   no :171
##  1st Qu.: 0.000   male  :286   1st Qu.:27.00   1st Qu.: 4.000   yes:430
```

```
##  Median : 0.000                Median :32.00   Median : 7.000
##  Mean   : 1.456                Mean   :32.49   Mean   : 8.178
##  3rd Qu.: 0.000                3rd Qu.:37.00   3rd Qu.:15.000
##  Max.   :12.000                Max.   :57.00   Max.   :15.000
##  religiousness     education       occupation        rating
##  Min.   :1.000   Min.   : 9.00   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:14.00   1st Qu.:3.000   1st Qu.:3.000
##  Median :3.000   Median :16.00   Median :5.000   Median :4.000
##  Mean   :3.116   Mean   :16.17   Mean   :4.195   Mean   :3.932
##  3rd Qu.:4.000   3rd Qu.:18.00   3rd Qu.:6.000   3rd Qu.:5.000
##  Max.   :5.000   Max.   :20.00   Max.   :7.000   Max.   :5.000
```

The Affairs dataset consists of 601 rows and 9 columns. The participants are asked about number of extramarital sexual intercourse affairs that they have been involved during the past year and some demographic data like age, sex, education and occupation have also been captured. There are 315 female participants and 286 male participants. The average age of participants is in the range of 30 to 34 years. Among the participants, 430 subjects have children and 171 dont. The average married years is between 8 to 9 years.

(b) Suppose we want to explore the characteristics of participants who engage in extramarital sexual intercourse (i.e. affairs). Instead of modeling the number of affairs, we will consider the binary outcome - had an affair versus didn't have an affair. Create a new variable to capture this response variable of interest.

```
#Creating  a vector to add the new variable
affairs_binary <-  rep(0, nrow(affairs_data))

#Assigning 0 for no affairs and 1 for more than 0 affairs
affairs_binary [affairs_data$affairs >0]=1

#Making the new variable factor
affairs_binary = as.factor(affairs_binary)

#Attaching new variable to the dataset
affairs_data =data.frame(affairs_data ,affairs_binary)
```

The variable affairs_binary captures information if there has been an extra martial affair or not. 0 value indicates no affair in past year and 1 value indicates atleast 1 affair extra martial affair in past year.

(c) Use an appropriate regression model to explore the relationship between having an affair and other personal characteristics. Comment on which covariates seem to be predictive of having an affair and which do not.

```
#Fitting a logistic regression
glm_affairs <- glm(affairs_binary~gender+age+yearsmarried+children+religiousness+
                     education + occupation +rating,
                   data = affairs_data, family="binomial")
summary(glm_affairs)
```

```
##
## Call:
## glm(formula = affairs_binary ~ gender + age + yearsmarried +
##     children + religiousness + education + occupation + rating,
##     family = "binomial", data = affairs_data)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -1.5713  -0.7499  -0.5690  -0.2539   2.5191
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.37726    0.88776   1.551 0.120807
## gendermale     0.28029    0.23909   1.172 0.241083
## age           -0.04426    0.01825  -2.425 0.015301 *
## yearsmarried   0.09477    0.03221   2.942 0.003262 **
## childrenyes    0.39767    0.29151   1.364 0.172508
## religiousness -0.32472    0.08975  -3.618 0.000297 ***
## education      0.02105    0.05051   0.417 0.676851
## occupation     0.03092    0.07178   0.431 0.666630
## rating        -0.46845    0.09091  -5.153 2.56e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 675.38  on 600  degrees of freedom
## Residual deviance: 609.51  on 592  degrees of freedom
## AIC: 627.51
##
## Number of Fisher Scoring iterations: 4
```

From the summary of the logistic model generated using the covariates, we can see that pvalue for age, yearsmarried, religiousness and rating are below 0.05. Thus these variables are significant in predicting if a participant has had an affair or not. Higher the age lower is the log odds of affair as they have negative relationship. Higher the number of yearsmarried higher is the log odds of affair and higher the religiousness lower is the log odds of affair. Higher the rating lower is the log odds of an affair. Gender, Education and having children doesnt have effect on a participant having an affair.

(d) Use an all subsets model selection procedure to obtain a "best" fit model. Is the model different from the full model you fit in part (c)? Which variables are included in the 'best" fit model? You might find the bestglm() function available in the bestglm package helpful.

```
#Converting affairs_binary to numeric as the regression requires it to be numeric
affairs_data$affairs_binary = as.numeric(affairs_data$affairs_binary)

#Best subset using AIC
bestglm_affairs <- bestglm(affairs_data, IC="AIC")
```

```
## Note: binary categorical variables converted to 0-1 so 'leaps' could be used.
```

```
bestglm_affairs$BestModel
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Coefficients:
## (Intercept)      affairs   gendermale          age  childrenyes
##    1.195116     0.098777     0.042180    -0.002858     0.066825
##      rating
```

```
##   -0.016371
```

Yes, the model fit using bestglm is different to the one fitted using logistic regression. The variables included in the bestglm are affairs, gendermale, age, childrenyes and rating. Only age and rating are the common variables in both the models.

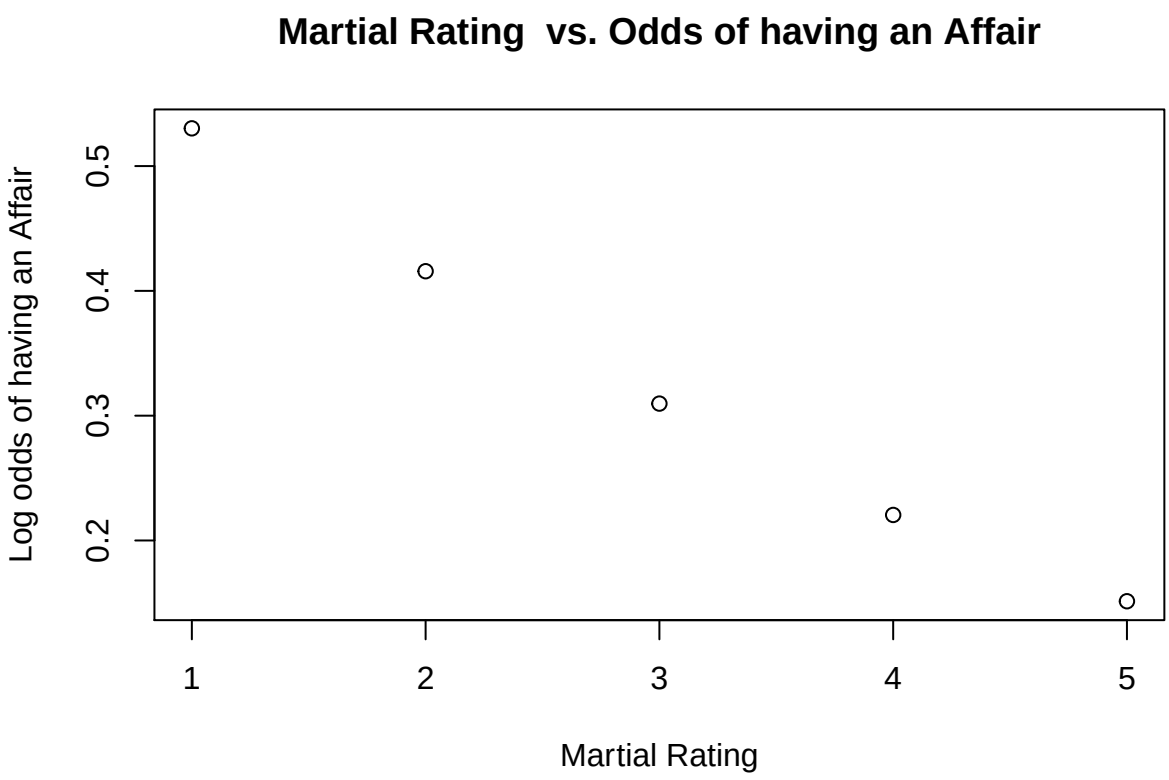
(e) Interpret the model parameters using the model from part (d).
From the model parameters we can conclude that variables affairs, gendermale and childrenyes have a direct positive relationship with a participant having an affair. This means an increase in any of thes variables causes a increase in the chances of a person having an affair. On the other hand, variables age and rating have a direct ngative relationship with the participant having an affair. This means lower the values of these variables higher is the chances of a person having an affair. The factor variables like gender and children have one other possibility such as female and no children respectively. But both of these have no effect on the chances of person having an affair.

(f) Create an artificial test dataset where martial rating varies from 1 to 5 and all other variables are set to their means. Use this test dataset and the predict function to obtain predicted probabilities of having an affair for case in the test data. Interpret your results and use a visualization to support your interpretation.

```
#Creating dummy dataset as asked in the question
dummydata_affairs <- data.frame(rating=c(1, 2, 3, 4, 5),age=mean(affairs_data$age),
                                yearsmarried=mean(affairs_data$yearsmarried),
                                religiousness=mean(affairs_data$religiousness))

#Fitting model only based on variables which have significance
#This will help in predicting for the test dataset created above
#Also converting affairs_binary back to factor as logistic requires it to be a factor
affairs_data$affairs_binary = as.factor(affairs_data$affairs_binary)
glm_significant_variables_affairs <- glm(affairs_binary ~ age +
                                          yearsmarried + religiousness + rating,
                                        data=affairs_data, family=binomial())

yhat_affairs <- predict(glm_significant_variables_affairs,
                        newdata=dummydata_affairs, type="response")
yhat_affairs
```

```
##         1         2         3         4         5
## 0.5302296 0.4157377 0.3096712 0.2204547 0.1513079
```

```
plot(dummydata_affairs$rating,yhat_affairs, xlab = 'Martial Rating' ,
     ylab = 'Log odds of having an Affair',
     main = 'Martial Rating  vs. Odds of having an Affair')
```



The log odds of a participant having an affair reduces as the Martia rating increases. The martial rating is defined in a way where 1 = very unhappy, 2 = somewhat unhappy, 3 = average, 4 = happier than average, 5 = very happy. Thus it is expected that with increase in happiness in marriage the odds of affairs would reduce and that is the same result we see in the above graphic.

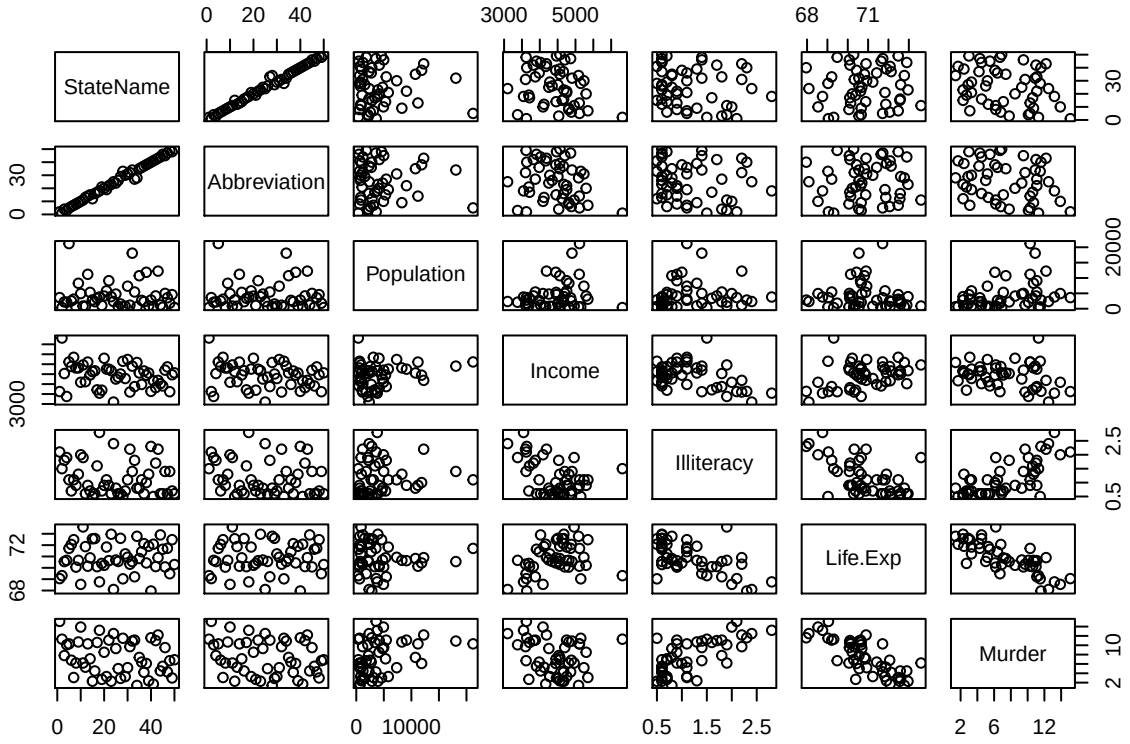
2. In this problem we will revisit the state dataset. This data, available as part of the base R package, contains various data related to the 50 states of the United States of America. Suppose you want to explore the relationship between a state's Murder rate and other characteristics of the state, for example population, illiteracy rate, and more. Follow the questions below to perform this analysis.

```
state <- state.x77
Abbreviation <- state.abb
Area <- state.area
Region <- state.region
state_center <- state.center
Division <- state.division
StateName <- state.name

#Combining state dataframe with Abbreviation, Region and Division vector.
#Area vector is redundant as the data is already present in the state df,
#thus removing it.
state_final <- data.frame(state, Abbreviation, Region, Division,StateName)
Area<-NULL
#state_name<-NULL
#Creating one single column Center_location which will have (latitude,longitude)
state_final <-  mutate(state_final, Center_location =
                            paste("(",state.center$x,",", state.center$y,")", sep = ''))
```

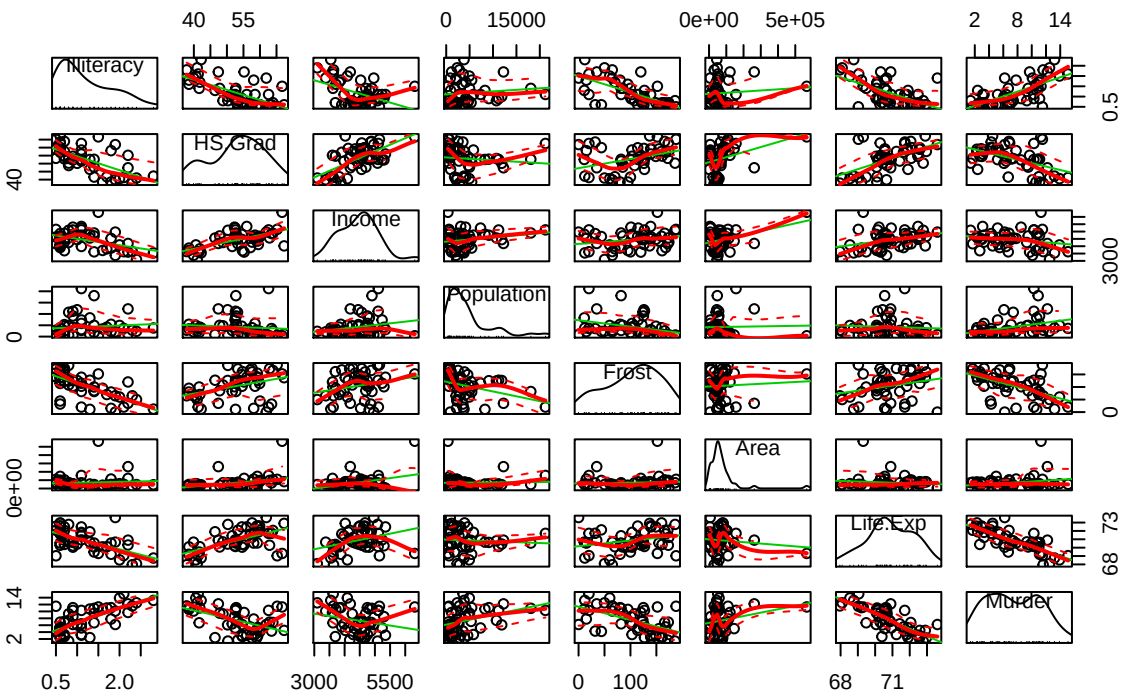
```
#Deleting state_center, as it is redundant now
state_center <-NULL

#Arranging the columns in logical order
state_final <- state_final[c(12,9,1,2,3,4,5,6,7,8,13,11,10)]
```

(a) Examine the bivariate relationships present in the data. Briefly discuss notable results. You might find the scatterplotMatrix() function available in the car package helpful.

```
#Ploting pairs and scatterplotmatrix to understand bivariate relationships
pairs(state_final[0:7])
```




```
scatterplotMatrix(~Illiteracy+HS.Grad+Income+Population+
                    Frost+Area+Life.Exp+Murder, data=state_final,
                  main="Scatter plot for Bivariate Analysis")
```



From the above graphics we can see that Murder rate is directly proportional to illiteracy. It is inversely proportional to HS Grad. Thus these two components are important in forming a model to explain modification in murder rates. We can also see a minor direct relationship between population and murder rate but just by eyeballing the visuals we cannot predict any relationship. Although we see a trend in LifeExp and Murder or Frost and Murder but they logically make little sense to be considered for our model. But to get an holistic model for the dataset we can try considering them and if they are statistically insignificant we can discard them. There is no particular relationship between Area and Murder. Other variables in the dateset are irrelevant for building our model.

(b) Fit a multiple linear regression model. How much variance in the murder rate across states do the predictor variables explain?

```
fit_states <- lm(Murder ~ Illiteracy + HS.Grad + Income + Population +
                   Area + Frost+ Life.Exp, data=state_final)
summary(fit_states)
```

```
##
## Call:
## lm(formula = Murder ~ Illiteracy + HS.Grad + Income + Population +
##     Area + Frost + Life.Exp, data = state_final)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4452 -1.1016 -0.0598  1.1758  3.2355
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

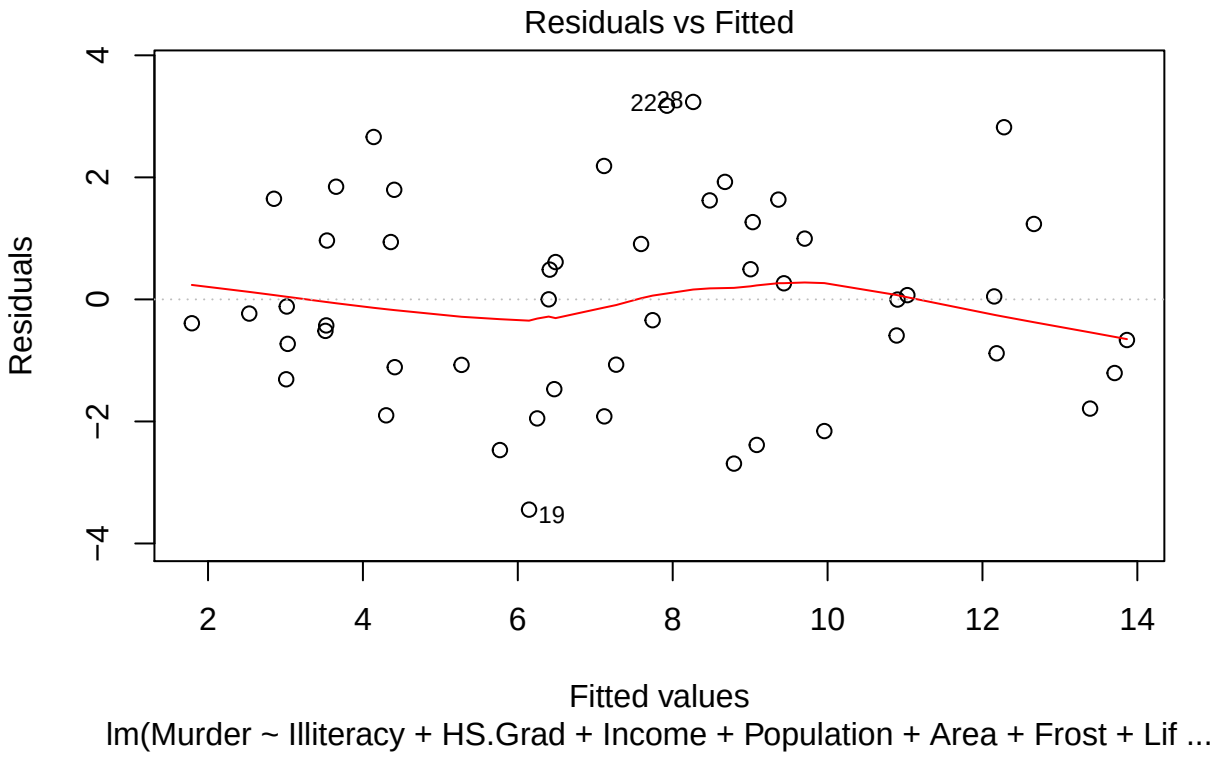
```
## (Intercept)  1.222e+02  1.789e+01   6.831 2.54e-08 ***
## Illiteracy   1.373e+00  8.322e-01   1.650  0.10641
## HS.Grad      3.234e-02  5.725e-02   0.565  0.57519
## Income      -1.592e-04  5.725e-04  -0.278  0.78232
## Population   1.880e-04  6.474e-05   2.905  0.00584 **
## Area         5.967e-06  3.801e-06   1.570  0.12391
## Frost       -1.288e-02  7.392e-03  -1.743  0.08867 .
## Life.Exp    -1.655e+00  2.562e-01  -6.459 8.68e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.746 on 42 degrees of freedom
## Multiple R-squared:  0.8083, Adjusted R-squared:  0.7763
## F-statistic: 25.29 on 7 and 42 DF,  p-value: 3.872e-13
```

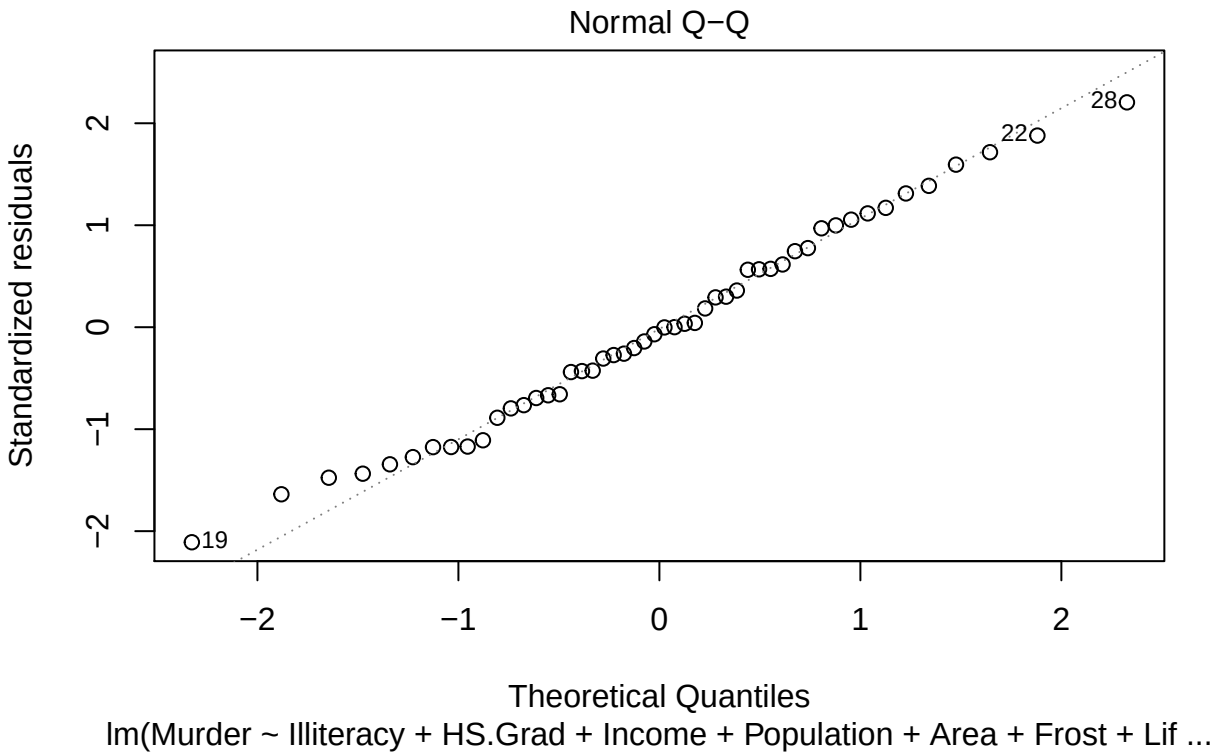
The most commonly used measures to measure the goodness of fit are the R^2 and RSE. These also tell us about the amount of variance explained by the predictors. Hence from the above summary we can say that 0.7763 or 77.63% of variance has been explanied by the predictors while predicting Murder. The ideal value for R^2 is 1, which would mean that model explains a large portion of variance in the response variables.

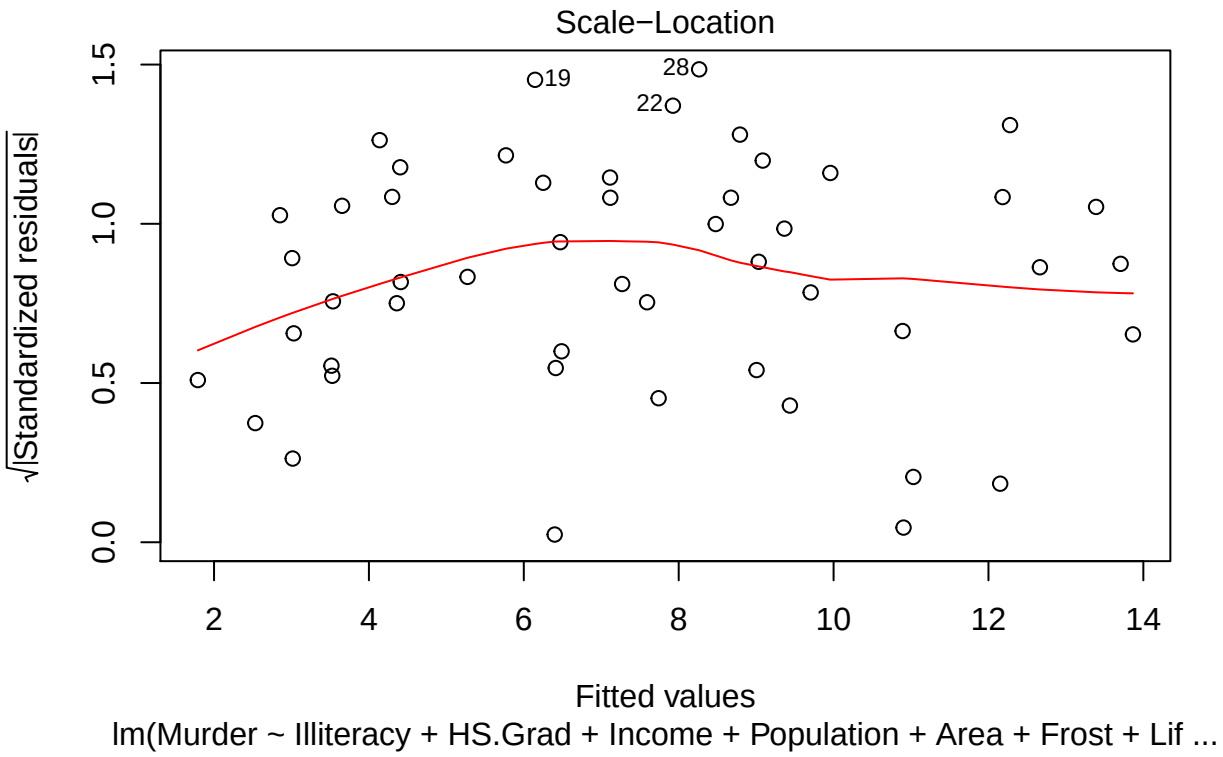
From the fitted model we can observe that only Life Expectancy and Population are statistically significant in predicting Murder rate as there pvalue is less than 0.05.

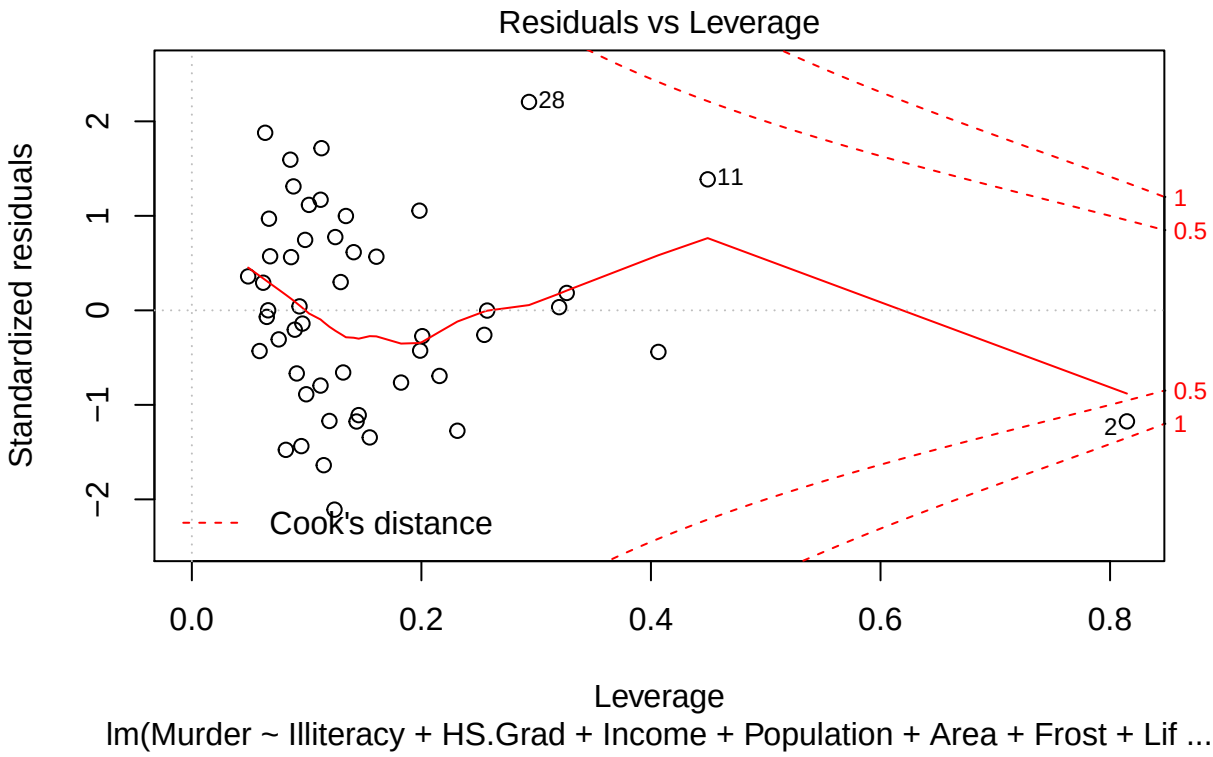
(c) Evaluate the statistical assumptions in your regression analysis from part (b) by performing a basic analysis of model residuals and any unusual observations. Discuss any concerns you have about your model.

```
#Plotting the generated model which shows 4 different plots to analyze statistical assumptions
plot(fit_states)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Murder ~ Illiteracy + HS.Grad + Income + Population + Area + Frost + Lif ...
2
19
4
2
0
−2
−4
2
4
6
8
10
12
14


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Murder ~ Illiteracy + HS.Grad + Income + Population + Area + Frost + Lif ...
19
22
28
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
19
28
22
2
4
6
8
10
12
14
Fitted values
lm(Murder ~ Illiteracy + HS.Grad + Income + Population + Area + Frost + Lif ...



Some of the assumptions made for regression analysis are as below: 1. Linearity : The relationship betwen the two variables should be linear. If this assumption is not satisfied then the model generated is a good representation of the relationship between the variables. Linearity can simply be tested by using a scatter plot. From the Residuals vs. Fitted values plot as the plot is quite symettrical across the red line, we can say that data is linear for the state dataset.
2. Multivariate Normality : In this we assume that the distribution is normal, if not we would need to transform the data to be normal by using logarithm, etc. From the Normal Q-Q plot we can see that the error between observed and predicted values is normal.
3. No MultiCollinearity : We assume that there is no multicollinearity in the data which means that variables are independent of each other and there is no dependency among them.
4. No AutoCorrelation : We assume that there is no autocorrelation in the data which means that residuals are independent of each other.
5. Homoscedastic : In this we assume that the error terms along the regression is equal. The Scale-Location plot helps us check this assumption. Because the residuals are more towards the bottom end of the line, this assumption is not completely fulfilled and hence is a concern.
From the Residuals vs. Leverage graph we can say that there are no influential case causing for any outliers. Thus Homoscedastic is the only assumption which is a cause of concern.

(d) Use a stepwise model selection procedure of your choice to obtain a "best" fit model. Is the model different from the full model you fit in part (b)? If yes, how so?

```
#Backward stepwise
step(lm(Murder ~ Illiteracy + HS.Grad + Income + Population +Area +
          Frost+ Life.Exp, data=state_final),
     direction = "backward")
```

```
## Start:  AIC=63.01
```

```
## Murder ~ Illiteracy + HS.Grad + Income + Population + Area +
##     Frost + Life.Exp
##
##              Df Sum of Sq    RSS    AIC
## - Income      1     0.236 128.27 61.105
## - HS.Grad     1     0.973 129.01 61.392
## <none>                    128.03 63.013
## - Area        1     7.514 135.55 63.865
## - Illiteracy  1     8.299 136.33 64.154
## - Frost       1     9.260 137.29 64.505
## - Population  1    25.719 153.75 70.166
## - Life.Exp    1   127.175 255.21 95.503
##
## Step:  AIC=61.11
## Murder ~ Illiteracy + HS.Grad + Population + Area + Frost + Life.Exp
##
##              Df Sum of Sq    RSS    AIC
## - HS.Grad     1     0.763 129.03 59.402
## <none>                    128.27 61.105
## - Area        1     7.310 135.58 61.877
## - Illiteracy  1     8.715 136.98 62.392
## - Frost       1     9.345 137.61 62.621
## - Population  1    27.142 155.41 68.702
## - Life.Exp    1   127.500 255.77 93.613
##
## Step:  AIC=59.4
## Murder ~ Illiteracy + Population + Area + Frost + Life.Exp
##
##              Df Sum of Sq    RSS    AIC
## <none>                    129.03 59.402
## - Illiteracy  1     8.723 137.75 60.672
## - Frost       1    11.030 140.06 61.503
## - Area        1    15.937 144.97 63.225
## - Population  1    26.415 155.45 66.714
## - Life.Exp    1   140.391 269.42 94.213
```

```
##
## Call:
## lm(formula = Murder ~ Illiteracy + Population + Area + Frost +
##     Life.Exp, data = state_final)
##
## Coefficients:
## (Intercept)   Illiteracy   Population         Area        Frost
##   1.202e+02    1.173e+00    1.780e-04    6.804e-06   -1.373e-02
##    Life.Exp
##  -1.608e+00
```

```
#Forward stepwise
step(lm(Murder~1,data = state_final),direction = "forward",
     scope = ~ Illiteracy + HS.Grad + Income + Population +Area + Frost+ Life.Exp)
```

```
## Start:  AIC=131.59
## Murder ~ 1
##
##              Df Sum of Sq    RSS     AIC
```

```
## + Life.Exp    1    407.14 260.61  86.550
## + Illiteracy  1    329.98 337.76  99.516
## + Frost       1    193.91 473.84 116.442
## + HS.Grad     1    159.00 508.75 119.996
## + Population  1     78.85 588.89 127.311
## + Income      1     35.35 632.40 130.875
## + Area        1     34.83 632.91 130.916
## <none>                    667.75 131.594
##
## Step:  AIC=86.55
## Murder ~ Life.Exp
##
##              Df Sum of Sq    RSS    AIC
## + Frost       1    80.104 180.50 70.187
## + Illiteracy  1    60.549 200.06 75.329
## + Population  1    56.615 203.99 76.303
## + Area        1    14.121 246.49 85.764
## <none>                    260.61 86.550
## + HS.Grad     1     1.124 259.48 88.334
## + Income      1     0.958 259.65 88.366
##
## Step:  AIC=70.19
## Murder ~ Life.Exp + Frost
##
##              Df Sum of Sq    RSS    AIC
## + Population  1   23.7098 156.79 65.146
## + Area        1   21.0840 159.42 65.976
## <none>                    180.50 70.187
## + Illiteracy  1    6.0663 174.44 70.477
## + Income      1    5.5598 174.94 70.622
## + HS.Grad     1    2.0679 178.44 71.610
##
## Step:  AIC=65.15
## Murder ~ Life.Exp + Frost + Population
##
##              Df Sum of Sq    RSS    AIC
## + Area        1   19.0402 137.75 60.672
## + Illiteracy  1   11.8262 144.97 63.225
## <none>                    156.79 65.146
## + HS.Grad     1    1.8215 154.97 66.561
## + Income      1    0.7393 156.06 66.909
##
## Step:  AIC=60.67
## Murder ~ Life.Exp + Frost + Population + Area
##
##              Df Sum of Sq    RSS    AIC
## + Illiteracy  1    8.7227 129.03 59.402
## <none>                    137.75 60.672
## + Income      1    1.2408 136.51 62.220
## + HS.Grad     1    0.7708 136.98 62.392
##
## Step:  AIC=59.4
## Murder ~ Life.Exp + Frost + Population + Area + Illiteracy
##
```

```
##           Df Sum of Sq    RSS    AIC
## <none>                 129.03 59.402
## + HS.Grad  1   0.76279 128.27 61.105
## + Income   1   0.02595 129.01 61.392
```

```
##
## Call:
## lm(formula = Murder ~ Life.Exp + Frost + Population + Area +
##     Illiteracy, data = state_final)
##
## Coefficients:
## (Intercept)     Life.Exp        Frost   Population         Area
##   1.202e+02   -1.608e+00   -1.373e-02    1.780e-04    6.804e-06
##  Illiteracy
##   1.173e+00
```

I have used backward and forward stepwise selection. Both the methods give exact same model. The model generated has only five predictor varaibles, namely, Population, Frost, Area, Life.Exp and Illiteracy. There are 3 more predictors than the predictors which were statistically significant when the full model was fitted. We see the difference in the models because in backward stepwise selection process, initially all predictor model is considered and then its AIC is compared by removing one of the predictors and if a reduction in AIC is observed the model is further truncated. Thus in iterative process the backward stepwise selection method reaches the best fit model which in this case seems to be of Population, Frost, Area, Life.Exp and Illiteracy for predicting Murder in States. In forward stepwise selection, we start with a Null model and go adding new Predictors to the model till we reach a best fit. Thus the difference between the full model and the best fit model using stepwise selection.

(e) Assess the model (from part (d)) generalizability. Perform a 10-fold cross validation to estimate model performance. Report the results.

Generalizability of a model is known from the R^2 and the MSE values. Thus fitting a logistic model as the best model obtained from subset selection method

```
best_fit_states <- lm( Murder ~ Life.Exp + Frost + Population + Area +
                           Illiteracy, data=state_final)
summary(best_fit_states)
```

```
##
## Call:
## lm(formula = Murder ~ Life.Exp + Frost + Population + Area +
##     Illiteracy, data = state_final)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2976 -1.0711 -0.1123  1.1092  3.4671
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.202e+02  1.718e+01   6.994 1.17e-08 ***
## Life.Exp    -1.608e+00  2.324e-01  -6.919 1.50e-08 ***
## Frost       -1.373e-02  7.080e-03  -1.939  0.05888 .
## Population   1.780e-04  5.930e-05   3.001  0.00442 **
## Area         6.804e-06  2.919e-06   2.331  0.02439 *
## Illiteracy   1.173e+00  6.801e-01   1.725  0.09161 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.712 on 44 degrees of freedom
## Multiple R-squared:  0.8068, Adjusted R-squared:  0.7848
## F-statistic: 36.74 on 5 and 44 DF,  p-value: 1.221e-14
```

```
#Calculating MSE
(MSE <- mean((best_fit_states$fitted.values - state_final$Murder)^2,na.rm = TRUE))
```

```
## [1] 2.580632
```

As the models R^2 has increase to 78.48% and MSE is only 2.58, we can say that the best fit model is much better than the initially fully fitted model and thus more generalizable.

Performing 10 fold Cross validation:

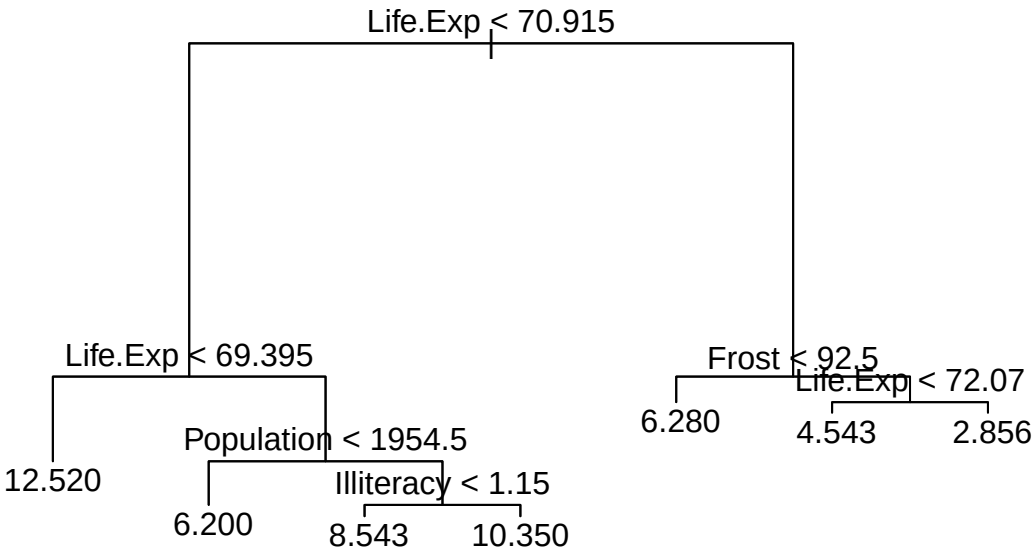
```
#Creating a vector to store the 10 fold cross validation errors
cv.error.10=rep(0 ,10)
#Iterating 10 times to fit different parts of the data as train and validation
#Also capturing the errors in the vector initiated above
for (i in 1:10){
glm.fit=glm(Murder ~ Life.Exp + Frost + Population + Area +
    Illiteracy,data=state_final)
cv.error.10[i]=cv.glm(state_final ,glm.fit ,K=10) $delta [1]
}
#MSE for the 10 fold cv
cv.error.10
```

```
##  [1] 4.099615 3.357563 3.425193 3.680787 3.937728 4.093208 3.535655
##  [8] 3.463679 3.622385 3.597155
```

Above I have used cv.glm function which produces a list of components. The two numbers in the delta vector depict the results of cross-validation. I have given K=10 for performing 10 fold cross validation. As different set of folds are used, the MSE ranges from 3.2 to 4.6, which estimates the performance of the model.

(f) Fit a regression tree using the same covariates in your "best" fit model from part (d). Use cross validation to select the "best" tree.

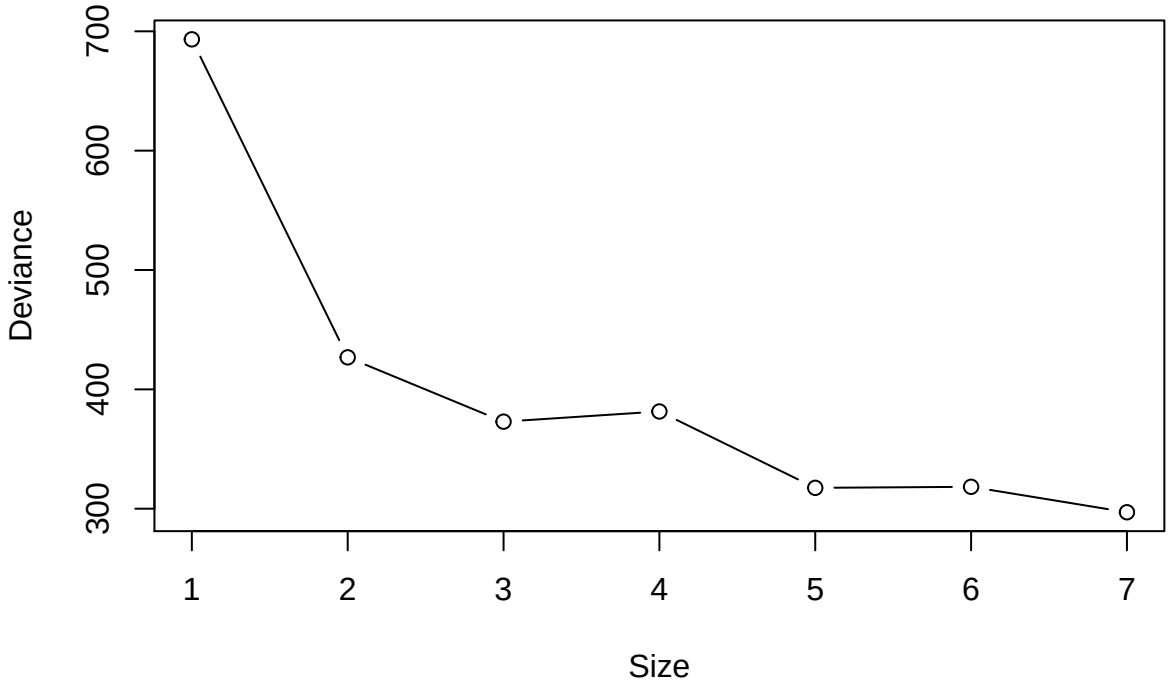
```
#Fitting Regression tree
tree_fit <- tree( Murder ~ Life.Exp + Frost + Population + Area + Illiteracy,
                  data = state_final)

#Plotting the fitted tree
plot(tree_fit)
text(tree_fit)
```



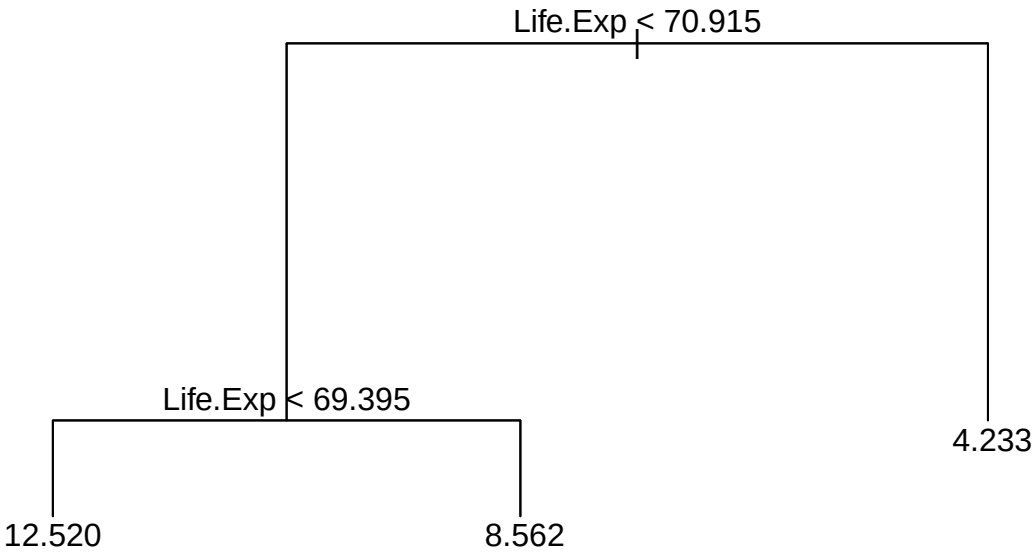

```
summary(tree_fit)
```

```
##
## Regression tree:
## tree(formula = Murder ~ Life.Exp + Frost + Population + Area +
##     Illiteracy, data = state_final)
## Variables actually used in tree construction:
## [1] "Life.Exp"   "Population" "Illiteracy" "Frost"
## Number of terminal nodes:  7
## Residual mean deviance:  2.813 = 121 / 43
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.50000 -1.18900  0.02222  0.00000  0.74290  4.02000
```

```
#Performing Cross Validation
tree.cv <- cv.tree(tree_fit)
plot(tree.cv$size ,tree.cv$dev ,type='b', xlab = "Size", ylab = "Deviance")
```




```
#Plotting Pruned Best Regression tree
prune.fit <- prune.tree(tree_fit, best = 3)
plot(prune.fit)
text(prune.fit ,pretty=0)
```




```
summary(prune.fit)
```

```
##
## Regression tree:
## snip.tree(tree = tree_fit, nodes = c(3L, 5L))
## Variables actually used in tree construction:
## [1] "Life.Exp"
## Number of terminal nodes:  3
## Residual mean deviance:  4.653 = 218.7 / 47
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -5.86200 -1.40200 -0.02917  0.00000  1.49700  6.06700
```

On performing cross validation and plotting the Size vs. Deviance plot, we see that though the deviance changes by small amount for size 3 tree to size 7 tree. Hence we can consider to reduce the size of tree to 3. Thus by pruning the tree to size 3, we have obtained the regression tree model for predicting Murder, but the Residual Mean deviance has increased from 2.8 to 4.65. Thus the regression tree model generated with size 7 is the best tree model.

(g) Compare the models from part (d) and (f) based on their performance. Which do you prefer? Be sure to justify your preference.

```
mean(cv.error.10)
```

```
## [1] 3.681297
```

```
summary(tree_fit)
```

```
##
## Regression tree:
## tree(formula = Murder ~ Life.Exp + Frost + Population + Area +
##     Illiteracy, data = state_final)
## Variables actually used in tree construction:
## [1] "Life.Exp"   "Population" "Illiteracy" "Frost"
## Number of terminal nodes:  7
## Residual mean deviance:  2.813 = 121 / 43
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.50000 -1.18900  0.02222  0.00000  0.74290  4.02000
```

On applying 10 fold cross validation to the the model generated in ques d, we get the MSE errors for each of the folds. Thus taking the mean of those errors, we get a mean MSE of 3.58. Now after pruning the tree the Residual mean deviance is 2.8, which indicates the MSE for this tree model. As the MSE for tree model is less than the MSE for regression model, we can say that performance of Tree model is better than Logistic model.

3. The Wisconsin Breast Cancer dataset is available as a comma-delimited text file on the UCI Machine Learning Repository http://archive.ics.uci.edu/ml. Our goal in this problem will be to predict whether observations (i.e. tumors) are malignant or benign.

(a) Obtain the data, and load it into R by pulling it directly from the web. (Do not download it and import it from a CSV file.) Give a brief description of the data.

```
#Reading data from the online repository
data <- fread('http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-
#Exploring dataset
dim(data)
```

```
## [1] 699  11
```

```
sapply (data, class)
```

```
##          V1          V2          V3          V4          V5          V6
##   "integer"   "integer"   "integer"   "integer"   "integer"   "integer"
##          V7          V8          V9         V10         V11
## "character"   "integer"   "integer"   "integer"   "integer"
```

```
summary(data)
```

```
##        V1                  V2               V3               V4
##  Min.   :   61634   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:  870688   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 1171710   Median : 4.000   Median : 1.000   Median : 1.000
##  Mean   : 1071704   Mean   : 4.418   Mean   : 3.134   Mean   : 3.207
##  3rd Qu.: 1238298   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 5.000
##  Max.   :13454352   Max.   :10.000   Max.   :10.000   Max.   :10.000
##        V5               V6              V7                 V8
##  Min.   : 1.000   Min.   : 1.000   Length:699         Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 2.000   Class :character   1st Qu.: 2.000
##  Median : 1.000   Median : 2.000   Mode  :character   Median : 3.000
##  Mean   : 2.807   Mean   : 3.216                      Mean   : 3.438
##  3rd Qu.: 4.000   3rd Qu.: 4.000                      3rd Qu.: 5.000
##  Max.   :10.000   Max.   :10.000                      Max.   :10.000
##        V9              V10              V11
##  Min.   : 1.000   Min.   : 1.000   Min.   :2.00
```

```
##  1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.:2.00
##  Median : 1.000   Median : 1.000   Median :2.00
##  Mean   : 2.867   Mean   : 1.589   Mean   :2.69
##  3rd Qu.: 4.000   3rd Qu.: 1.000   3rd Qu.:4.00
##  Max.   :10.000   Max.   :10.000   Max.   :4.00
```

The data has been taken from Diagnostic Wisconsin Breast Cancer Database. The Features have been computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. Ten real-valued features were computed for each cell nucleus:

a) radius (mean of distances from center to points on the perimeter)
b) texture (standard deviation of gray-scale values)
c) perimeter
d) area
e) smoothness (local variation in radius lengths)
f) compactness (perimeter^2 / area - 1.0)
g) concavity (severity of concave portions of the contour)
h) concave points (number of concave portions of the contour)
i) symmetry
j) fractal dimension ("coastline approximation" - 1)

The results of this data collection and further analysis helped in predicting field 2, diagnosis: B = benign, M = malignant, sets are linearly separable using all 30 input features and best predictive accuracy obtained using one separating plane in the 3-D space of Worst Area, Worst Smoothness and Mean Texture. Estimated accuracy 97.5% using repeated 10-fold crossvalidations. Classifier has correctly diagnosed 176 consecutive new patients as of November 1995. Reference for this answer - http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.names

The dataset in all consists of 699 rows and 11 columns. There are 16 rows which have an unknown data for one of the features. The datatype for "Class" feature is given as integer, as there are only two levels of values for this features, we should converts its datatype to factor.

(b) Tidy the data, ensuring that each variable is properly named and cast as the correct data type. Discuss any missing data.

```
#Adding column headers to the dataset
colnames(data) <- c("Sample_code_number","Clump_Thickness","Uniformity_of_Cell_Size", "Uniformity_of_Cel
                    "Marginal_Adhesion", "Single_Epithelial_Cell_Size","Bare_Nuclei", "Bland_Chromatin"
                    "Normal_Nucleoli", "Mitoses", "Class")

#Converting datatype to factor
data$Class <-as.factor(data$Class)

#Cleaning dataset
clean_data <- subset(data, data$Bare_Nuclei != "?")

#Converting datatype to integer
clean_data$Bare_Nuclei <-as.integer(clean_data$Bare_Nuclei)
```

As discussed above, there are 16 rows with unknown data in the one of the features. On further inspection we see that feature "Bare Nuclei" has 16 rows as "?" which is the represents unknown data and can be removed from the dataset. Also converting datatype of "Bare Nuclei" from character to integer.

(c) Split the data into a training and validation set such that a random 70% of the observations are in the training set.

```
#Segregating data in train and validation
set.seed(11)
train <- sample(nrow(clean_data), nrow(clean_data)*0.7)
data.train <- clean_data[train, ]
data.validation<- clean_data[-train, ]
```

(d) Fit a regression model to predict whether tissue samples are malignant or benign. Classify cases in the validation set. Compute and discuss the resulting confusion matrix.

```
#Fitting a logistic regression
glm_data <- glm(Class~Clump_Thickness+Uniformity_of_Cell_Size+Uniformity_of_Cell_Shape+
                  Marginal_Adhesion+Single_Epithelial_Cell_Size+Bare_Nuclei+Bland_Chromatin
                +Normal_Nucleoli+Mitoses, data = data.train, family="binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
#Calculating predictions on validation set
yhat = predict(glm_data, newdata=data.validation, type="response")

#If probabilty > 0.5 segregating as malign tumour or else benign
yhat <- ifelse(yhat > 0.5,4,2)
testnew <- cbind(data.validation, yhat)

#Calculating falsepositives using confusion matrix
confmatrix_class<-table(testnew$Class,testnew$yhat)
confmatrix_class
```

```
##
##       2   4
##   2 132   7
##   4   6  60
```

```
#Accuracy of the model
sum(diag(confmatrix_class))/sum(confmatrix_class)
```

```
## [1] 0.9365854
```

From the resulting confusion matrix we can see that there are 7 falsepositives and 6 truenegatives. The accuracy of this logistic regression model is 93.65%. Looking at the accuracy of the model and the confusion matrix we can say that generated logistic regression model is fairly good model.

(e) Fit a random forest model to predict whether tissue samples are malignant or benign. Classify cases in the validation set. Compute and discuss the resulting confusion matrix.

```
#Fitting random forest
rf_data <- randomForest(Class~Clump_Thickness+Uniformity_of_Cell_Size+Uniformity_of_Cell_Shape
                        +Marginal_Adhesion+Single_Epithelial_Cell_Size+Bare_Nuclei+Bland_Chromatin
                        +Normal_Nucleoli+Mitoses, data = data.train, importance=TRUE)


#Predicting Class using the above random forest model
yhat2 <- predict(rf_data, newdata = data.validation)

#Checking accuracy of Random forest model
confmatrix_rf<-table(yhat2,data.validation$Class)
confmatrix_rf
```

```
##
## yhat2   2   4
##     2 133   1
##     4   6  65
```

```
sum(diag(confmatrix_rf))/sum(confmatrix_rf)
```

```
## [1] 0.9658537
```

From the resulting confusion matrix we can see that there is only 1 falsepositive and 6 truenegatives. The accuracy of this random forest model has increased to 96.58%.

(f) Compare the models from part (d) and (e) using ROC curves. Which do you prefer? Be sure to justify your preference.

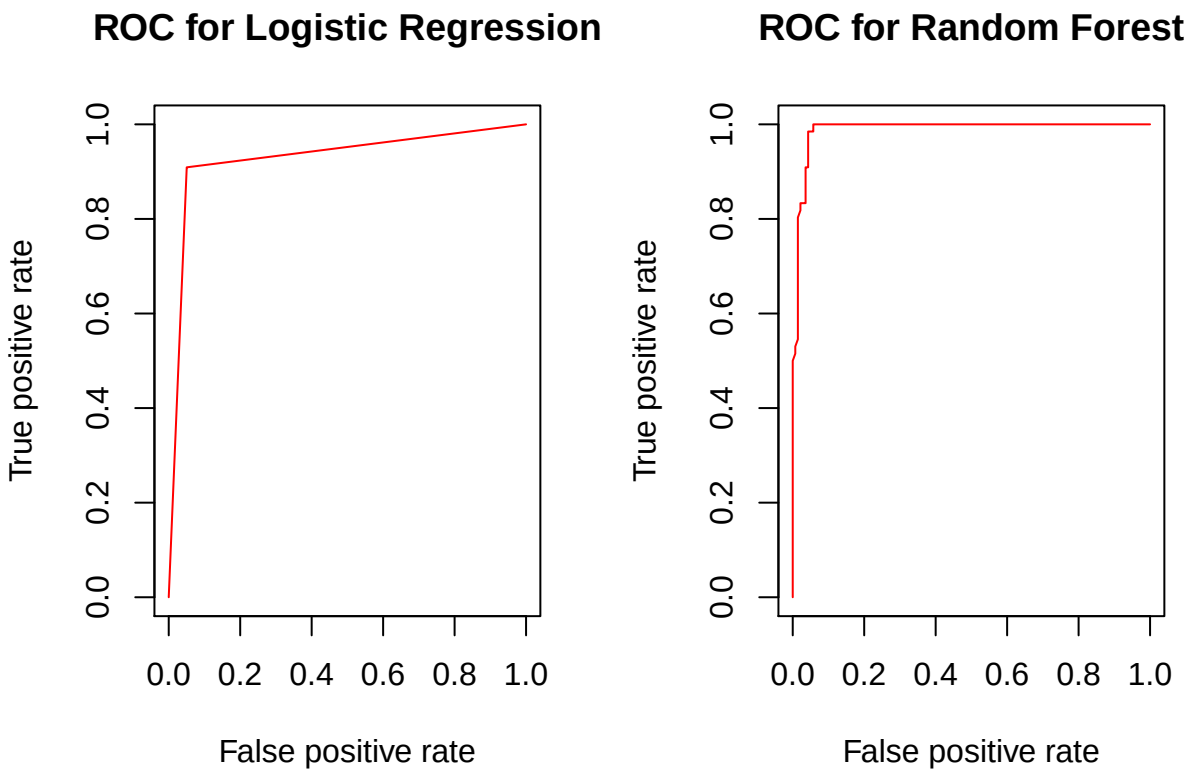
```
#ROC for Logistic
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
pr <- prediction(yhat, data.validation$Class)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")

#ROC for Random Forest with pclass and Title as predictors
probRF <- predict(rf_data, newdata = data.validation, type='prob')
predRF <- prediction(probRF[,2],data.validation$Class)
perfRF <- performance(predRF, measure = "tpr", x.measure = "fpr")

#Plotting ROC for Logistic and Random Forest models
par(mfrow=c(1,2))
plot(prf, col=rainbow(5),main = "ROC for Logistic Regression" )
plot(perfRF, col=rainbow(5),main = "ROC for Random Forest" )
```



We know that, greater the AUR better is the model. Thus from above we see that AUC for Random Forest model is greater than AUC for logistic regression model, hence I would prefer Random Forest model to predict the class of the Tumour. We can also graphically see from the ROC curve that ROC for Random Forest model is more near the left to p corner than the ROC for Logistic Regression model which also is an indication for Random Forest being better model.

4. Please answer the questions below by writing a short response.

(a) Describe three real-life applications in which classification might be useful. Describe the response, as well as the predictors. Is the goal in each application inference or predictions? Explain your answer.

Classifications is used for when the response variables are qualitative. Below are three real life examples:

1. One of the major application of classification is seen in the banking industry where a person would default or not on the credit card bills is predicted. Hence considering this example, the response variable would be credit card defaults, yes or no. The predictor variables could vary depending from bank to bank, but some of them would be income of the person, occupation of the person (student or professional) and previous credit history (good or bad), etc. The aim in this scenario is prediction as we are trying to predict if a person would default or no on the credit card bills.

2. Another example for classification is predicting if the IPO (launch of a share in market) would be a success or a failure. For example, when the IPO of Facebook was launched many of the finance pundits, predicted its success or failure which is again an qualitative variable and thus the response varible here would be the performance of IPO on launch (success or failure). The predictors for this classification could be performance of similar stocks in previous IPO's, Market image of the company (good or bad), Value of stock at IPO launch (high or low), etc. Here also the aim of this classfication is prediction.

3. Finally, another major classification problem in real life scenario is classification of emails as spam.

Here the response varibale is the type of email (spam or not spam). The predictor variables could be email address of the sender, content in the email, timing of the email, etc. The goal again in this application is predictions.

(b) Describe three real-life applications in which regression might be useful. Describe the response, as well as the predictors. Is the goal in each application inference or predictions? Explain your answer.

Regression is used for when the response variables are quantitative. Below are three real life examples:

1. Understanding the factors affecting the income of high level executive in a fortune 500 company is an regression problem. The salary of the high level executive becomes the response variable. Some of the factors affecting high level executives salary would be industry sector, size, current market valuations of the company, etc. These would be the predictors. As we are trying to understand all the factors affecting the income, we can infer that some variables strongly affect and some dont have any affect on the income of high level executives, thus making this problem an Inference problem.

2. Another example is prediction of change in oil prices week over week. As we are predicting an quantitative varibale it is an regression problem. The response varible in this example is the percent change in oil prices week over week. The predictor varibles in this example could be change in oil prices in the previous weeks, change in dollar value in last week, performance of oil companies, innovation in oil generation industry, etc. This is an example of prediction.

3. Linear regression can be applied to pricing of car models. Here the price of the car model would be the response variable. The brand, number of engines, miles per gallon, horsepower, etc can be used as the predictor variables. As we are predicting the price of a car model, this is an example of prediction.

(c) What are the advantages and disadvantages of a very flexible (versus a less flexible) approach for regression or classification? Under what circumstances might a more flexible approach be preferred to a less flexible approach? When might a less flexible approach be preferred?

The advantages of very flexible model over less flexible model is that, flexible models make better predictions. They are more closer to the real values. The bias is also very less for flexible models.
The disadvantages for flexible models are: They are very complex and hence difficult to interpret. Also there may be case of over fitting.
We prefer more flexible approachg over less flexible wheneve the goal of the model is to make predictions. The less flexibe model is preferred whenever the goal of the model is interpretation of the results.

5. Suppose we have a dataset with five predictors, X1 = GPA, X2 = IQ, X3 = Gender (1 for Female, and 0 for Male), X4 = Interaction between GPA and IQ, and X5 = Interaction between GPA and Gender. The response is starting salary after graduation (in thousands of dollars). Suppose we use least squares to fit the model and get $\beta_0 = 50$; $\beta_1 = 20$; $\beta_2 = 0{:}07$ ^ $\beta_3 = 35$; $\beta_4 = 0{:}01$, and $\beta_5 = -10$.

(a) Which answer is correct and why?

i. For a fixed value of IQ and GPA, males earn more on average than females.

ii. For a fixed value of IQ and GPA, females earn more on average than males.

iii. For a fixed value of IQ and GPA, males earn more on average than females provided that the GPA is high enough.

iv. For a fixed value of IQ and GPA, females earn more on average than males provided that the GPA is high enough.

When the IQ and GPA are fixed, we cancompare the earnings of males and females which would be reprsented as below: salary(male) = β_0 + β_1 * GPA + $\beta_2 IQ$ + β_4GPAIQ
salary(female) = β_0 + β_1GPA + $\beta_2 IQ$ + β_3 + β_4GPAIQ + β_5GPA

The difference between male and female salary is given by salary(male) - salary(female) = -β_3-$\beta_5 GPA$ = *-35+10*GPA

Thus if the GPA is high enough we can say that MAles earn more than females, hence option iii.

(b) Predict the salary of a female with IQ of 110 and a GPA of 4.0.
From the above questions, the salary of female is given by :
salary(female) = β_0 + $\beta_1 GPA$ + β_2IQ + β_3 + $\beta_4 GPA$IQ + β_5*GPA

Given ID = 100 and GPA = 4.0, salary(female) = 50 + 20*4 + 0.07*110 + 35*1 + 0.014110 - 10*4 = 50 + 80 + 7.7 + 35 + 4.4 - 40 = 137.1

(c) True or false: Since the coeficient for the GPA/IQ interaction term is very small, there is little evidence of an interaction effect. Justify your answer.

False. The significance of any term is not dependent on its coefficient value. The significance is known by the difference in the value of coefficeint from the distribution of the null hypothesis. Thus by looking at the magnitude of the distance of the coefficient from zero we can ignore or keep the GPA/IQ interaction term and not depending on the magnitude of the interaction term. If the p value for interaction term is less than 0.05 then it is significant or else it is insignificant.

6. Apply boosting, bagging and random forests to a dataset of your choice that we have used in class. Be sure to fit the models on a training set and evaluate their performance on a test set.

```
#Considering Stock Market dataset from ISLR
attach (Smarket)
smarket =data.frame(Smarket)

#Segregating in Training and Test
set.seed(1)
train <- sample(nrow(smarket), nrow(smarket)*0.7)
smarket$Direction <- ifelse(smarket$Direction == "Up", 1, 0)
smarket.train <- smarket[train, ]
smarket.test <- smarket[-train, ]

#Logistic Regression
logit.fit <- glm ( Direction~Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume ,
                   data = smarket.train , family = "binomial" )
logit.probs <- predict(logit.fit, newdata = smarket.test, type = "response")
logit.pred <- ifelse(logit.probs > 0.5, 1, 0)
table(smarket.test$Direction, logit.pred)
```

```
##    logit.pred
##       0   1
##   0  90  78
##   1 103 104
```

```
#Boosting
boost.fit <- gbm(Direction~Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume ,
                 data = smarket.train , distribution = "bernoulli", n.trees = 5000)
boost.probs <- predict(boost.fit, newdata = smarket.test, n.trees = 5000)
boost.pred <- ifelse(boost.probs > 0.5, 1, 0)
table(smarket.test$Direction, boost.pred)
```

```
##    boost.pred
##       0   1
##   0 166   2
##   1 205   2
```

```
#Bagging
bag.fit <- randomForest(Direction~Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume ,
                        data = smarket.train , mtry = 2)
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
bag.probs <- predict(bag.fit, newdata = smarket.test)
bag.pred <- ifelse(bag.probs > 0.5, 1, 0)
table(smarket.test$Direction, bag.pred)
```

```
##    bag.pred
##       0   1
##   0  92  76
##   1  87 120
```

```
#Random Forest
rf.fit <- randomForest(Direction~Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume ,
                       data = smarket.train )
```

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
rf.probs <- predict(rf.fit, newdata = smarket.test)
rf.pred <- ifelse(rf.probs > 0.5, 1, 0)
table(smarket.test$Direction, rf.pred)
```

```
##    rf.pred
##       0   1
##   0  89  79
##   1  88 119
```

```
#Comparing between all the above methods
dfpred = data.frame(logistic=logit.pred, boosting = boost.pred, bagging=bag.pred, randomforest=rf.pred)

y = smarket.test$Direction
sapply(dfpred, function(dfpred) mean((dfpred - y)^2))
```

```
##     logistic     boosting      bagging randomforest
##    0.4826667    0.5520000    0.4346667    0.4453333
```

(a) How accurate are the results compared to simple methods like linear or logistic regression?
I have applied boosting, bagging, and random forests to the Stock Market data set from the ISLR package. In this dataset I have tried to predict Direction being Up or Down. I have predicted Direction using Lag1, Lag2, Lag3, Lag4, Lag5 and Volume as the predictor variables. With respect to the accuracy of results, bagging and Random Forest perform better in terms of the Mean squared Error than Logistic regression. At the same time Boosting performs worse than Logistic regression.

(b) Which of the approaches yields the best performance?
From the Mean squared error we can saty that Bagging performs the best for this data set.