Final573

Shuyang Wu

12/4/2016

Problem 1 (25 pts) In this problem we will use the infidelity data, 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 (Hollinghead 7-point classification with reverse numbering), and a numeric self-rating of their marriage (from 1=very unhappy to 5=very happy).

```
#load dataset
data("Affairs")
affairs <- Affairs

#Data exploration
head(affairs) #look at top 10 rows of the dataset
```

```
##    affairs gender age yearsmarried children religiousness education
## 4        0   male  37        10.00       no             3        18
## 5        0 female  27         4.00       no             4        14
## 11       0 female  32        15.00      yes             1        12
## 16       0   male  57        15.00      yes             5        18
## 23       0   male  22         0.75       no             2        17
## 29       0 female  32         1.50       no             2        17
##    occupation rating
## 4           7      4
## 5           6      4
## 11          1      4
## 16          6      5
## 23          6      3
## 29          5      5
```

```
dim(affairs)
```

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

```
summary(affairs) #see summary statistics of the dataset
```

```
##     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
```

```r
prop.table(table(affairs$gender)) #gender distribution
```

```
##
##    female      male
## 0.5241265 0.4758735
```

```r
prop.table(table(affairs$children)) #children status distribution
```

```
##
##        no       yes
## 0.2845258 0.7154742
```

```r
#create binary variable for affairs
affairs$A <- rep(FALSE, 601)
affairs$A[affairs$affairs != 0] <- TRUE

#logistic regression model predicting affair status
glm.a <- glm(A ~ gender+age+yearsmarried+children+religiousness+education+occupation+rating, family = "b
summary(glm.a)
```

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

```r
#find best glm model reference: ftp://cran.r-project.org/pub/R/web/packages/bestglm/bestglm.pdf
best <- bestglm(affairs[,-1], family = binomial, IC = "AIC", method = "exhaustive")
```

```
## Morgan-Tatar search since family is non-gaussian.
```

```r
best$BestModels #show top five models
```

```
##   gender  age yearsmarried children religiousness education occupation
## 1   TRUE TRUE         TRUE    FALSE          TRUE     FALSE      FALSE
## 2   TRUE TRUE         TRUE     TRUE          TRUE     FALSE      FALSE
## 3  FALSE TRUE         TRUE     TRUE          TRUE     FALSE       TRUE
## 4  FALSE TRUE         TRUE    FALSE          TRUE     FALSE      FALSE
## 5  FALSE TRUE         TRUE     TRUE          TRUE     FALSE      FALSE
##   rating Criterion
## 1   TRUE  621.8590
## 2   TRUE  622.1529
## 3   TRUE  623.2897
## 4   TRUE  623.3578
## 5   TRUE  623.4076
```

```r
print(best) #print best model and its parameter
```

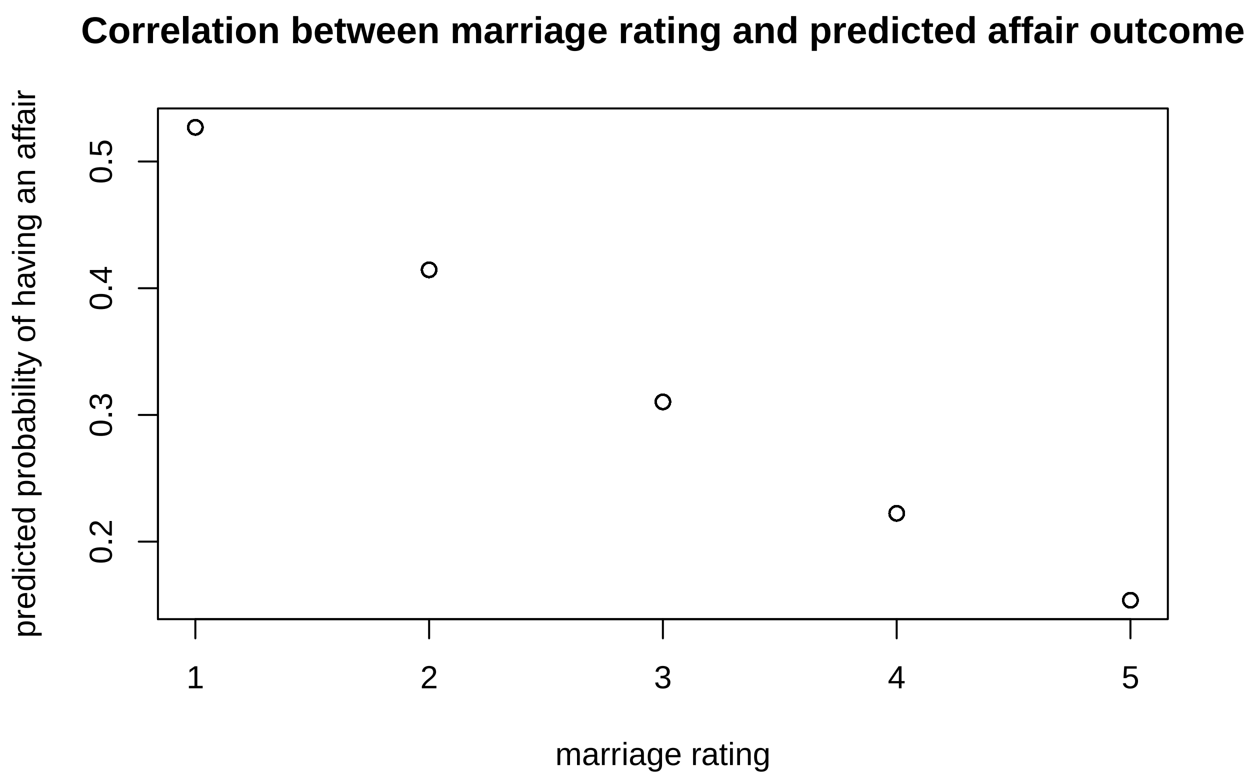
```
## AIC
## BICq equivalent for q in (0.809977862034941, 0.912637327155774)
## Best Model:
##                  Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)    1.94760307 0.61233521  3.180616 1.469624e-03
## gendermale     0.38612217 0.20702802  1.865072 6.217131e-02
## age           -0.04392545 0.01806068 -2.432104 1.501138e-02
## yearsmarried   0.11132715 0.02982799  3.732304 1.897360e-04
## religiousness -0.32714238 0.08947345 -3.656307 2.558752e-04
## rating        -0.46721157 0.08928317 -5.232919 1.668543e-07
```

```r
#create an artificial test dataset
yearsmarried <- rep(mean(affairs$yearsmarried), 601)
test <- data.frame(yearsmarried)
test$religiousness <- rep(mean(affairs$religiousnes), 601)
r <- 1:5
set.seed(1)
test$rating <- sample(r, 601, replace = TRUE)
test <- as.data.frame(test)
glm.best <- glm(A ~ yearsmarried+religiousness+rating, family = "binomial", data = affairs)

#predict the testset and visualize the relationship between predictor variable and the predicted outcom
pred.a <- predict(glm.best, test, type = "response")
pred.b <- rep(FALSE, 601)
pred.b[pred.a > 0.5] <- TRUE
prop.table(table(pred.b)) #proportion of people having affair
```

```
## pred.b
##     FALSE      TRUE
## 0.8069884 0.1930116
```

```r
test$pred.affair <- pred.a
plot(test$rating, test$pred.affair, main = "Correlation between marriage rating and predicted affair ou
```



(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?

The average number of affairs made was 1.456. The gender distribution was 52.4% women and 49.6% men. The average age of participants was 32.5, the average duration of the marriages was 8 years. 28.4% couple did not have children, 71.5% did.

(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.

(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.

Religiousness, rating about the marriage seem to be most predictive of having an affair, years married is less predictive than these two, age is also a weak predictor. Gender, whether have children or not, education and occupation are not predictive of 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.

The best fit model has yearsmarried, religiousness and rating as predictors. It did not include age which was a less significant predictor shown in the glm model in (c).

(e) Interpret the model parameters using the model from part (d).

The fit model can be interpreted as: whether of not having an affair = 0.055 x yearsmarried - 0.331 x religiousness - 0.453 x rating + 1.138. The first positive parameter means the more years married, the more likely one is to have an affair. The two negative parameters show that the less regilious and less satisfied/rating of the marriage the more likely one is to have an affair.

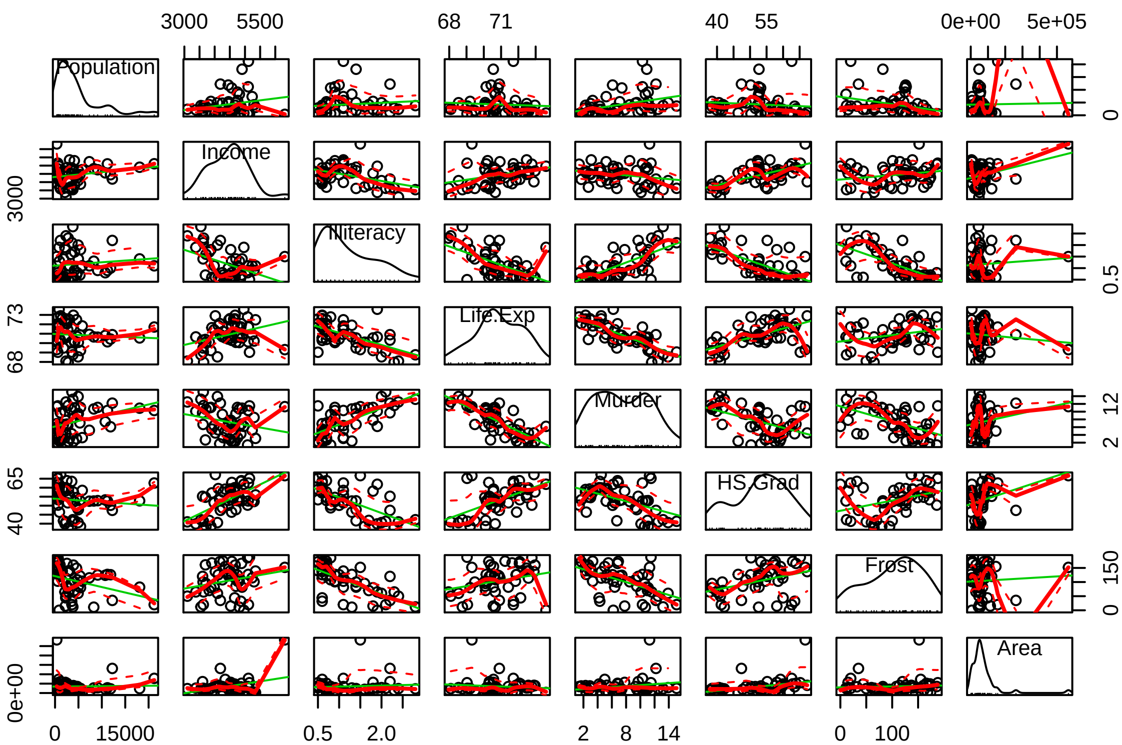
(f) Create an artificial test dataset where martial rating varies from 1 to 5 and all other variables are set to their means. Use this test dataset and the predict function to obtain predicted probabilities of having an affair for case in the test data. Interpret your results and use a visualization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

The overall predicted proportion of people having an affair is 0.193. My result also shows a negative correlation between the marriage rating and the predicted probability of having an affair, meaning the higher rating one gives regarding the marriage, the less likely one would have an affair.

Problem 2

Problem 2 (25 pts) 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.

```r
state <- data.frame(state.x77)
scatterplotMatrix(state)
```

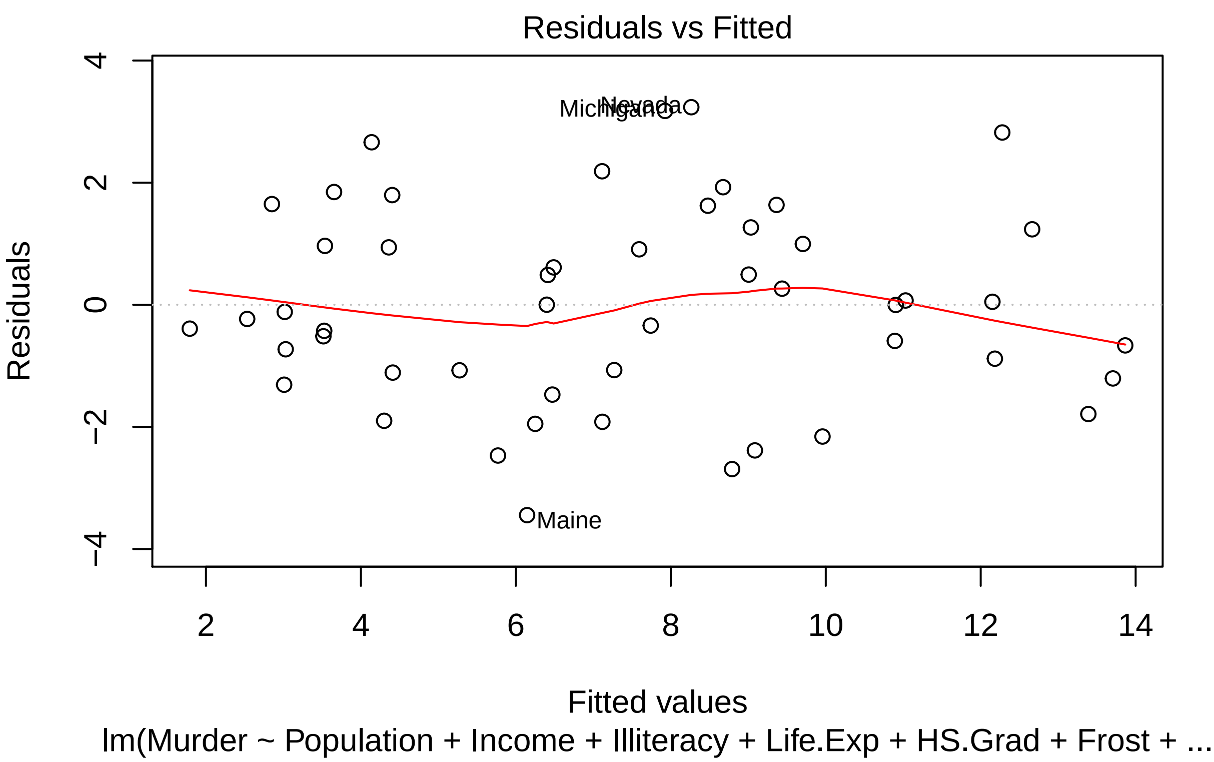


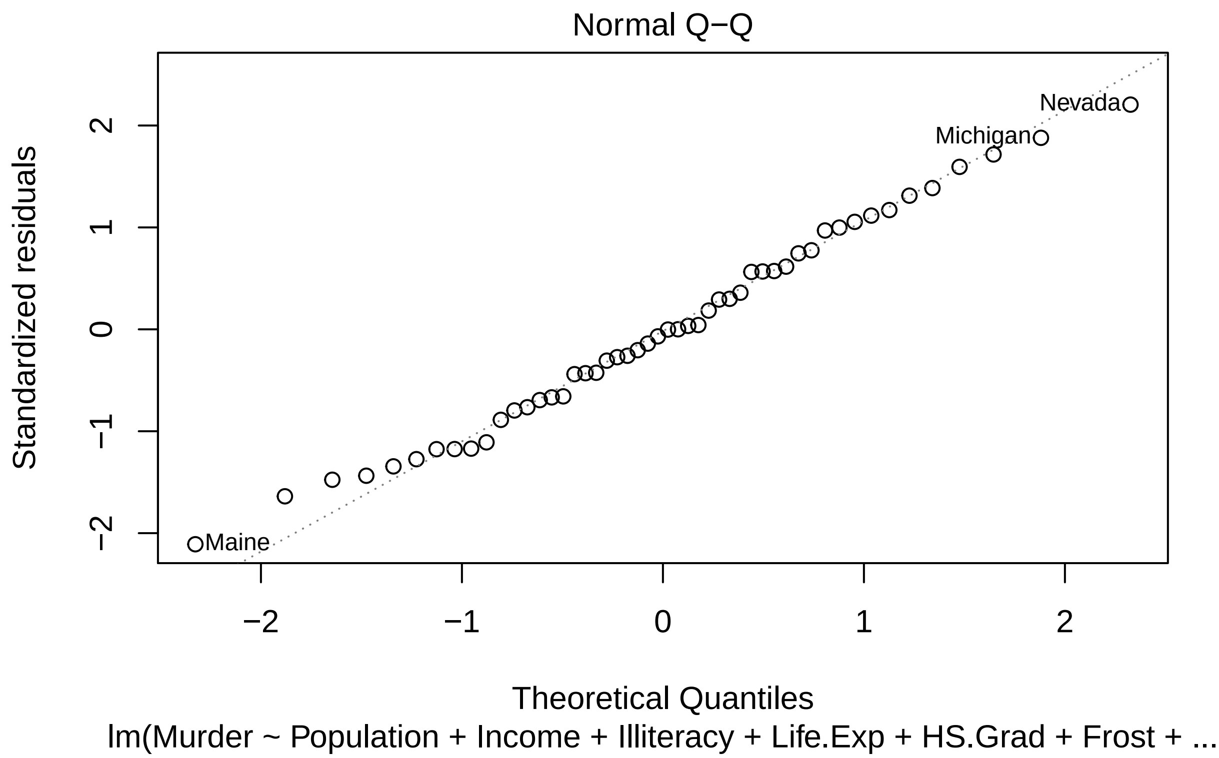

```r
#linear regression model
lm.s <- lm(Murder ~ Population+Income+Illiteracy+Life.Exp+HS.Grad+Frost+Area, data = state)
summary(lm.s)
```

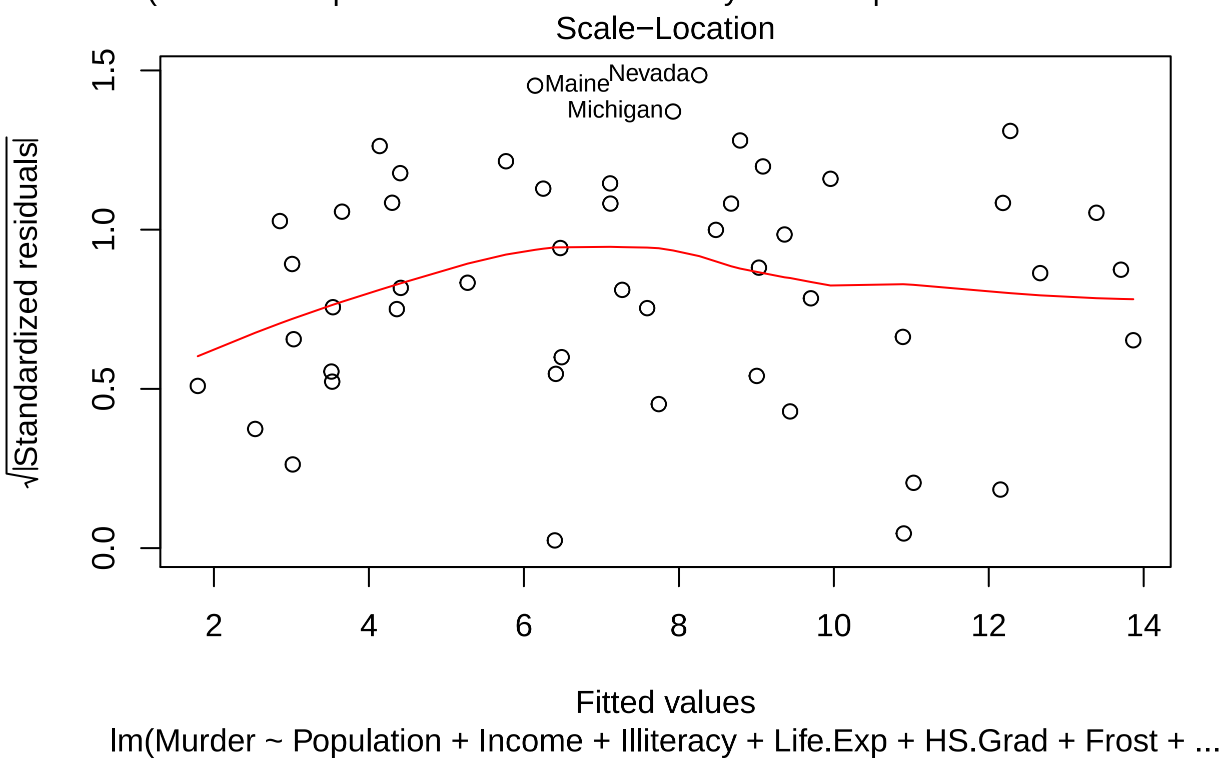
```
##
## Call:
## lm(formula = Murder ~ Population + Income + Illiteracy + Life.Exp +
##     HS.Grad + Frost + Area, data = state)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4452 -1.1016 -0.0598  1.1758  3.2355
```

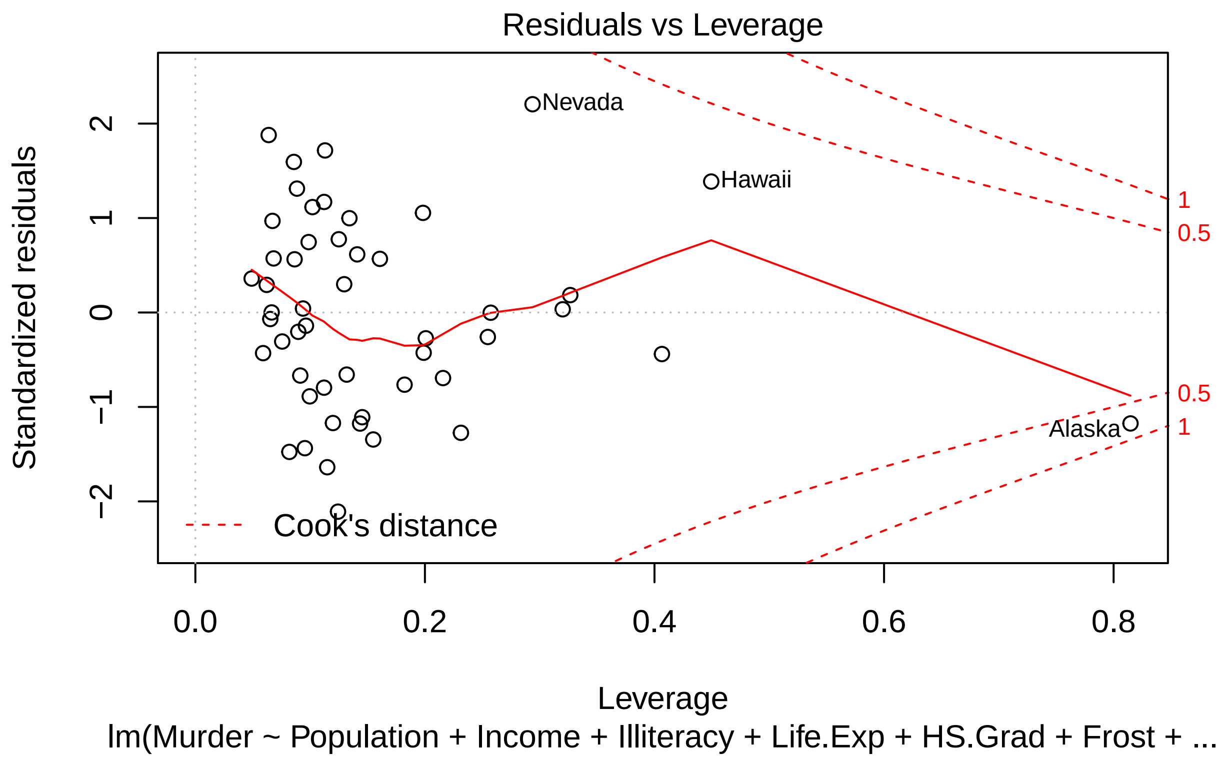
```
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.222e+02  1.789e+01   6.831 2.54e-08 ***
## Population    1.880e-04  6.474e-05   2.905  0.00584 **
## Income       -1.592e-04  5.725e-04  -0.278  0.78232
## Illiteracy    1.373e+00  8.322e-01   1.650  0.10641
## Life.Exp     -1.655e+00  2.562e-01  -6.459 8.68e-08 ***
## HS.Grad       3.234e-02  5.725e-02   0.565  0.57519
## Frost        -1.288e-02  7.392e-03  -1.743  0.08867 .
## Area          5.967e-06  3.801e-06   1.570  0.12391
## ---
## 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
```

```r
plot(lm.s)
```




Normal Q−Q
Standardized residuals
Nevada
Michigan
Maine
Theoretical Quantiles
lm(Murder ~ Population + Income + Illiteracy + Life.Exp + HS.Grad + Frost + ...


Scale−Location
√|Standardized residuals|
Maine
Nevada
Michigan
Fitted values
lm(Murder ~ Population + Income + Illiteracy + Life.Exp + HS.Grad + Frost + ...




```r
#stepwise model selection
steps <- stepAIC(lm.s, direction="both")
```

```
## Start:  AIC=63.01
## Murder ~ Population + Income + Illiteracy + Life.Exp + HS.Grad +
##     Frost + Area
##
##              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 ~ Population + Illiteracy + Life.Exp + HS.Grad + Frost +
##     Area
##
##              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
## + Income      1     0.236 128.03 63.013
## - Population  1    27.142 155.41 68.702
## - Life.Exp    1   127.500 255.77 93.613
```

```
##
## Step:  AIC=59.4
## Murder ~ Population + Illiteracy + Life.Exp + Frost + Area
##
##               Df Sum of Sq    RSS    AIC
## <none>                     129.03 59.402
## - Illiteracy  1     8.723 137.75 60.672
## + HS.Grad     1     0.763 128.27 61.105
## + Income      1     0.026 129.01 61.392
## - 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
```

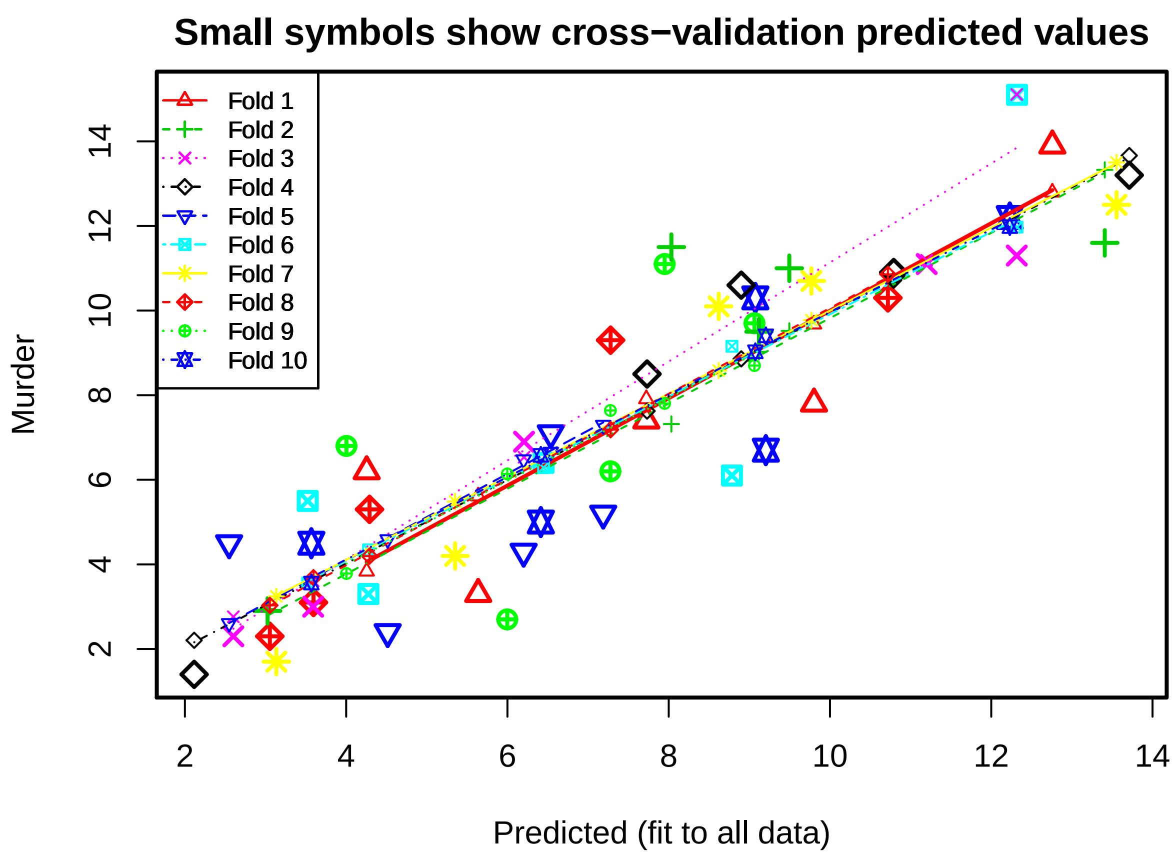
```
steps$anova #display final model
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## Murder ~ Population + Income + Illiteracy + Life.Exp + HS.Grad +
##     Frost + Area
##
## Final Model:
## Murder ~ Population + Illiteracy + Life.Exp + Frost + Area
##
##
##        Step Df  Deviance Resid. Df Resid. Dev      AIC
## 1                               42   128.0331 63.01329
## 2  - Income  1 0.2357225        43   128.2688 61.10526
## 3 - HS.Grad  1 0.7627900        44   129.0316 59.40172
```

```
#10 fold cross-validation
lm.f <- lm(Murder ~ Population + Illiteracy + Life.Exp + Frost + Area, state)
cv.l <- cv.lm(state, lm.f, m=10)
```

```
## Analysis of Variance Table
##
## Response: Murder
##            Df Sum Sq Mean Sq F value  Pr(>F)
## Population  1   78.9    78.9   26.89 5.2e-06 ***
## Illiteracy  1  299.6   299.6  102.18 4.8e-13 ***
## Life.Exp    1  136.8   136.8   46.63 2.0e-08 ***
## Frost       1    7.5     7.5    2.57   0.116
## Area        1   15.9    15.9    5.43   0.024 *
## Residuals  44  129.0     2.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Warning in cv.lm(state, lm.f, m = 10):
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```




```
##
## fold 1
## Observations in test set: 5
##              Arizona Georgia Hawaii Massachusetts  Ohio
## Predicted       9.80   12.76   4.25          5.64  7.72
## cvpred          9.68   12.79   3.84          5.61  7.91
## Murder          7.80   13.90   6.20          3.30  7.40
## CV residual    -1.88    1.11   2.36         -2.31 -0.51
##
## Sum of squares = 15.9    Mean square = 3.19    n = 5
##
## fold 2
## Observations in test set: 5
##              Nebraska Nevada South Carolina Tennessee Virginia
## Predicted      3.0256   8.03          13.41      9.50    9.119
## cvpred         2.9443   7.32          13.33      9.52    9.025
## Murder         2.9000  11.50          11.60     11.00    9.500
## CV residual   -0.0443   4.18          -1.73      1.48    0.475
##
## Sum of squares = 22.9    Mean square = 4.58    n = 5
##
## fold 3
## Observations in test set: 5
##              Alaska Minnesota North Carolina Wisconsin Wyoming
## Predicted     12.32     2.601         11.199     3.591   6.206
## cvpred        15.11     2.761         11.189     3.611   6.532
## Murder        11.30     2.300         11.100     3.000   6.900
```

```
## CV residual  -3.81    -0.461         -0.089    -0.611   0.368
##
## Sum of squares = 15.2    Mean square = 3.05    n = 5
##
## fold 4
## Observations in test set: 5
##             Kentucky Louisiana Maryland New York North Dakota
## Predicted       8.90    13.712    7.732   10.790        2.115
## cvpred          8.86    13.664    7.623   10.716        2.205
## Murder         10.60    13.200    8.500   10.900        1.400
## CV residual     1.74    -0.464    0.877    0.184       -0.805
##
## Sum of squares = 4.7    Mean square = 0.94    n = 5
##
## fold 5
## Observations in test set: 5
##             Indiana New Jersey Rhode Island Utah Washington
## Predicted     6.538       7.19         4.51 2.55       6.20
## cvpred        6.657       7.29         4.59 2.61       6.48
## Murder        7.100       5.20         2.40 4.50       4.30
## CV residual   0.443      -2.09        -2.19 1.89      -2.18
##
## Sum of squares = 17.7    Mean square = 3.54    n = 5
##
## fold 6
## Observations in test set: 5
##             Alabama New Hampshire Oklahoma Pennsylvania Vermont
## Predicted     12.32          4.28    6.448         8.78    3.52
## cvpred        11.97          4.34    6.287         9.16    3.56
## Murder        15.10          3.30    6.400         6.10    5.50
## CV residual    3.13         -1.04    0.113        -3.06    1.94
##
## Sum of squares = 24    Mean square = 4.8    n = 5
##
## fold 7
## Observations in test set: 5
##             Arkansas Florida Mississippi Oregon South Dakota
## Predicted       8.62   9.769        13.6   5.35         3.13
## cvpred          8.58   9.774        13.5   5.49         3.24
## Murder         10.10  10.700        12.5   4.20         1.70
## CV residual     1.52   0.926        -1.0  -1.29        -1.54
##
## Sum of squares = 8.2    Mean square = 1.64    n = 5
##
## fold 8
## Observations in test set: 5
##             California Connecticut Idaho   Iowa Missouri
## Predicted       10.718       3.595  4.29  3.053     7.28
## cvpred          10.856       3.685  4.20  3.031     7.18
## Murder          10.300       3.100  5.30  2.300     9.30
## CV residual     -0.556      -0.585  1.10 -0.731     2.12
##
## Sum of squares = 6.88    Mean square = 1.38    n = 5
##
```

```
## fold 9
## Observations in test set: 5
##              Colorado Delaware Maine Michigan New Mexico
## Predicted        4.00     7.28  6.00     7.95       9.06
## cvpred           3.78     7.64  6.14     7.80       8.70
## Murder           6.80     6.20  2.70    11.10       9.70
## CV residual      3.02    -1.44 -3.44     3.30       1.00
##
## Sum of squares = 34.9    Mean square = 6.97    n = 5
##
## fold 10
## Observations in test set: 5
##              Illinois Kansas Montana  Texas West Virginia
## Predicted        9.07  3.568    6.41 12.230           9.2
## cvpred           9.03  3.558    6.58 11.985           9.4
## Murder          10.30  4.500    5.00 12.200           6.7
## CV residual      1.27  0.942   -1.58  0.215          -2.7
##
## Sum of squares = 12.3    Mean square = 2.47    n = 5
##
## Overall (Sum over all 5 folds)
##   ms
## 3.25
```

```r
#cross-validated standard error of estimate =
sqrt((3.19 + 4.58 + 3.05 + 0.94 + 3.54 + 4.8 + 1.64 + 1.38 + 6.97 + 2.47)/50)
```

```
## [1] 0.807
```

(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.

Income and high school graduate rate, life expectancy are positively associated. Income and Murder rate are negatively associated. Illiteracy and life expectancy, high school graduate rate, Frost (mean number of days with minimum temperature below freezing) are negatively associated. Illiteracy and Murder rate are possitively associated. Life expectancy and murder rate are negatively associated. High school graduate rate and life expectance are possitively associated. Frost and murder rate are nagetively associated.

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

Predictor variables were able to explain 77.6% of the variance in the murder rate based on this linear regression model.

(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.

There are four assumptions for the regression analysis: linear and additive relationship between predictors and the outcome; no correlation between errors; constant variance of the errors; and normality of the error distribution. Because the residual versus fitted plot shows symmetrically distributed points around the horizonal line with a relatively constant variance, which means the errors variance is constant. Also because the fitted line is reasonably linear, the predictors and the outcome have a linear relationship. Correlation between errors is often associated with time series data, thus does not apply here. The normal Q-Q plot shows a close linear fitted line as well, meaning that error distribution is normal. All in all, all assumptions for regression analysis is satisfied. I do not have any concerns about my model

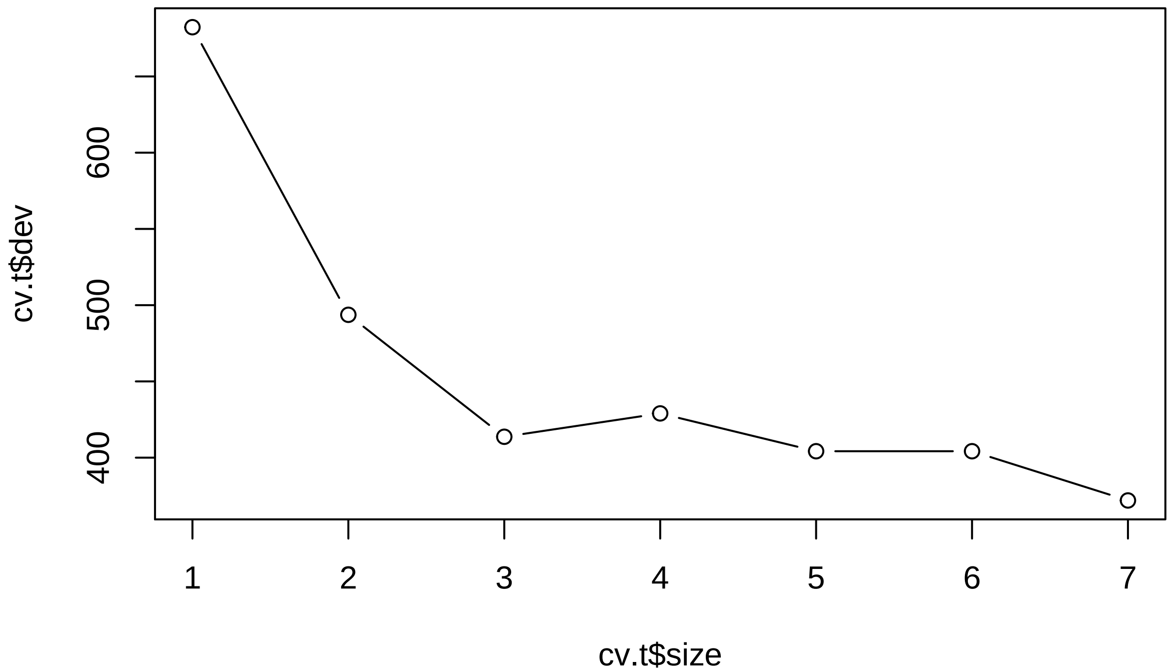
(d) Use a stepwise model selection procedure of your choice to obtain a best fit model. Is the model different from the full model you fit in part (b)? If yes, how so?

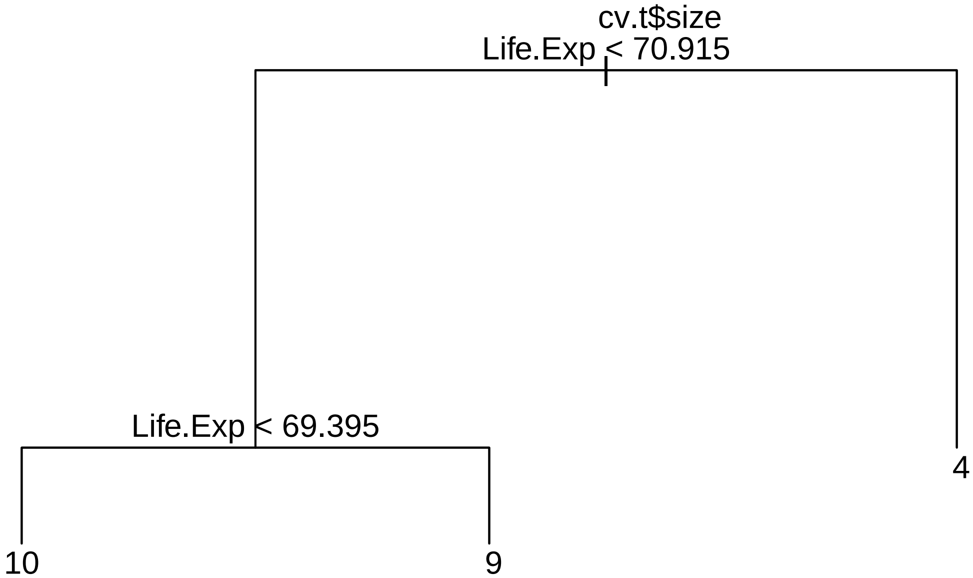
The best fit model includes only five predictors: Population + Illiteracy + Life.Exp + Frost + Area, comparing to the model I fit in part(b) which contains all possible predictors from the dataset. (Because lower AIC indicates better fit and the AIC stopped decreasing at the model described above which means that dropping any more variables would not result in a lower AIC, thus this is the best fit model.)

(e) Assess the model (from part (d)) generalizability. Perform a 10-fold cross validation to estimate model performance. Report the results. Mean square errors of the ten fold cross-validation were all under 7, which I think it's not bad. The estimated standard error of estimate is 0.807, less than 1. Thus I think the model predicts outside datasets pretty well meaning it has a high generalizability. Reference: statmethods.net/stats/regression.html

```
##
## Regression tree:
## tree(formula = Murder ~ Population + Illiteracy + Life.Exp +
##     Frost + Area, data = state)
## Variables actually used in tree construction:
## [1] "Life.Exp"   "Population" "Illiteracy" "Frost"
## Number of terminal nodes:  7
## Residual mean deviance:  2.81 = 121 / 43
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -3.50   -1.19    0.02    0.00    0.74    4.02
```

```
## $size
## [1] 7 6 5 4 3 2 1
##
## $dev
## [1] 372 404 404 429 414 494 682
##
## $k
## [1]  -Inf  11.2  12.2  27.5  46.9  91.0 358.0
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```





(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. Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot. 3 node = best tree (lowest error rate)

(g) Compare the models from part (d) and (f) based on their performance. Which do you prefer? Be sure to justify your preference. Good model performence is identified with low variance and low squared bias. The tree model mean squared error is 2.81 whereas the regression model mean squared error is 0.651. Thus I prefer the regression model over the tree model.

Problem 3

Problem 3 (25 pts) 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.

```r
#load the dataset
wdbc <- read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdb
colnames(wdbc) <- c("id","diagnosis","radius_mean","texture_mean","perimeter_mean","area_mean","smoothne
summary(wdbc)
```

```
##        id           diagnosis  radius_mean     texture_mean
##  Min.   :8.67e+03   B:357     Min.   : 6.98   Min.   : 9.7
##  1st Qu.:8.69e+05   M:212     1st Qu.:11.70   1st Qu.:16.2
##  Median :9.06e+05             Median :13.37   Median :18.8
##  Mean   :3.04e+07             Mean   :14.13   Mean   :19.3
##  3rd Qu.:8.81e+06             3rd Qu.:15.78   3rd Qu.:21.8
##  Max.   :9.11e+08             Max.   :28.11   Max.   :39.3
##  perimeter_mean    area_mean    smoothness_mean  compactness_mean
##  Min.   : 43.8   Min.   : 144   Min.   :0.0526   Min.   :0.019
##  1st Qu.: 75.2   1st Qu.: 420   1st Qu.:0.0864   1st Qu.:0.065
##  Median : 86.2   Median : 551   Median :0.0959   Median :0.093
##  Mean   : 92.0   Mean   : 655   Mean   :0.0964   Mean   :0.104
##  3rd Qu.:104.1   3rd Qu.: 783   3rd Qu.:0.1053   3rd Qu.:0.130
##  Max.   :188.5   Max.   :2501   Max.   :0.1634   Max.   :0.345
##  concavity_mean  concave.points_mean symmetry_mean
##  Min.   :0.000   Min.   :0.0000      Min.   :0.106
##  1st Qu.:0.030   1st Qu.:0.0203      1st Qu.:0.162
##  Median :0.062   Median :0.0335      Median :0.179
##  Mean   :0.089   Mean   :0.0489      Mean   :0.181
##  3rd Qu.:0.131   3rd Qu.:0.0740      3rd Qu.:0.196
##  Max.   :0.427   Max.   :0.2012      Max.   :0.304
##  fractal_dimension_mean   radius_se       texture_se    perimeter_se
##  Min.   :0.0500         Min.   :0.112   Min.   :0.36   Min.   : 0.76
##  1st Qu.:0.0577         1st Qu.:0.232   1st Qu.:0.83   1st Qu.: 1.61
##  Median :0.0615         Median :0.324   Median :1.11   Median : 2.29
##  Mean   :0.0628         Mean   :0.405   Mean   :1.22   Mean   : 2.87
##  3rd Qu.:0.0661         3rd Qu.:0.479   3rd Qu.:1.47   3rd Qu.: 3.36
##  Max.   :0.0974         Max.   :2.873   Max.   :4.88   Max.   :21.98
##     area_se    smoothness_se     compactness_se    concavity_se
##  Min.   :  7   Min.   :0.00171   Min.   :0.0023   Min.   :0.000
##  1st Qu.: 18   1st Qu.:0.00517   1st Qu.:0.0131   1st Qu.:0.015
##  Median : 25   Median :0.00638   Median :0.0204   Median :0.026
##  Mean   : 40   Mean   :0.00704   Mean   :0.0255   Mean   :0.032
##  3rd Qu.: 45   3rd Qu.:0.00815   3rd Qu.:0.0324   3rd Qu.:0.042
##  Max.   :542   Max.   :0.03113   Max.   :0.1354   Max.   :0.396
##  concave.points_se  symmetry_se     fractal_dimension_se  radius_worst
##  Min.   :0.0000    Min.   :0.0079   Min.   :0.00089      Min.   : 7.9
##  1st Qu.:0.0076    1st Qu.:0.0152   1st Qu.:0.00225      1st Qu.:13.0
##  Median :0.0109    Median :0.0187   Median :0.00319      Median :15.0
##  Mean   :0.0118    Mean   :0.0205   Mean   :0.00379      Mean   :16.3
##  3rd Qu.:0.0147    3rd Qu.:0.0235   3rd Qu.:0.00456      3rd Qu.:18.8
##  Max.   :0.0528    Max.   :0.0790   Max.   :0.02984      Max.   :36.0
##  texture_worst  perimeter_worst   area_worst    smoothness_worst
##  Min.   :12.0   Min.   : 50.4   Min.   : 185   Min.   :0.0712
##  1st Qu.:21.1   1st Qu.: 84.1   1st Qu.: 515   1st Qu.:0.1166
##  Median :25.4   Median : 97.7   Median : 686   Median :0.1313
##  Mean   :25.7   Mean   :107.3   Mean   : 881   Mean   :0.1324
##  3rd Qu.:29.7   3rd Qu.:125.4   3rd Qu.:1084   3rd Qu.:0.1460
##  Max.   :49.5   Max.   :251.2   Max.   :4254   Max.   :0.2226
```

```
##  compactness_worst concavity_worst concave.points_worst symmetry_worst
##  Min.   :0.027     Min.   :0.000   Min.   :0.0000       Min.   :0.156
##  1st Qu.:0.147     1st Qu.:0.114   1st Qu.:0.0649       1st Qu.:0.250
##  Median :0.212     Median :0.227   Median :0.0999       Median :0.282
##  Mean   :0.254     Mean   :0.272   Mean   :0.1146       Mean   :0.290
##  3rd Qu.:0.339     3rd Qu.:0.383   3rd Qu.:0.1614       3rd Qu.:0.318
##  Max.   :1.058     Max.   :1.252   Max.   :0.2910       Max.   :0.664
##  fractal_dimension_worst
##  Min.   :0.0550
##  1st Qu.:0.0715
##  Median :0.0800
##  Mean   :0.0839
##  3rd Qu.:0.0921
##  Max.   :0.2075
```

```r
str(wdbc)
```

```
## 'data.frame':    569 obs. of  32 variables:
##  $ id                     : int  842302 842517 84300903 84348301 84358402 843786 844359 84458202 844
##  $ diagnosis              : Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ radius_mean            : num  18 20.6 19.7 11.4 20.3 ...
##  $ texture_mean           : num  10.4 17.8 21.2 20.4 14.3 ...
##  $ perimeter_mean         : num  122.8 132.9 130 77.6 135.1 ...
##  $ area_mean              : num  1001 1326 1203 386 1297 ...
##  $ smoothness_mean        : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
##  $ compactness_mean       : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
##  $ concavity_mean         : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
##  $ concave.points_mean    : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
##  $ symmetry_mean          : num  0.242 0.181 0.207 0.26 0.181 ...
##  $ fractal_dimension_mean : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
##  $ radius_se              : num  1.095 0.543 0.746 0.496 0.757 ...
##  $ texture_se             : num  0.905 0.734 0.787 1.156 0.781 ...
##  $ perimeter_se           : num  8.59 3.4 4.58 3.44 5.44 ...
##  $ area_se                : num  153.4 74.1 94 27.2 94.4 ...
##  $ smoothness_se          : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
##  $ compactness_se         : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
##  $ concavity_se           : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
##  $ concave.points_se      : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
##  $ symmetry_se            : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
##  $ fractal_dimension_se   : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
##  $ radius_worst           : num  25.4 25 23.6 14.9 22.5 ...
##  $ texture_worst          : num  17.3 23.4 25.5 26.5 16.7 ...
##  $ perimeter_worst        : num  184.6 158.8 152.5 98.9 152.2 ...
##  $ area_worst             : num  2019 1956 1709 568 1575 ...
##  $ smoothness_worst       : num  0.162 0.124 0.144 0.21 0.137 ...
##  $ compactness_worst      : num  0.666 0.187 0.424 0.866 0.205 ...
##  $ concavity_worst        : num  0.712 0.242 0.45 0.687 0.4 ...
##  $ concave.points_worst   : num  0.265 0.186 0.243 0.258 0.163 ...
##  $ symmetry_worst         : num  0.46 0.275 0.361 0.664 0.236 ...
##  $ fractal_dimension_worst: num  0.1189 0.089 0.0876 0.173 0.0768 ...
```

```r
wdbc$id<-factor(wdbc$id)
wdbc$diagnosis <- as.character(wdbc$diagnosis)
wdbc$cancerous[wdbc$diagnosis == "M" ] <- TRUE
wdbc$cancerous[wdbc$diagnosis == "B"] <- FALSE
```

```r
#split dataset into training and testset
smp_size <- floor(0.70 * nrow(wdbc))
set.seed(12)
train_ind <- sample(seq_len(nrow(wdbc)), size = smp_size)
wdbc.train <- wdbc[train_ind, ]
wdbc.test <- wdbc[-train_ind, ]

#logistic regression
glm.bc <- glm(cancerous ~ radius_mean+texture_mean+perimeter_mean+area_mean+smoothness_mean+compactness_
```

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

```r
#+radius_se+texture_se+perimeter_se+area_se+smoothness_se+compactness_se+concavity_se+concave.points_se
summary(glm.bc)
```

```
##
## Call:
## glm(formula = cancerous ~ radius_mean + texture_mean + perimeter_mean +
##     area_mean + smoothness_mean + compactness_mean + concavity_mean +
##     concave.points_mean + symmetry_mean + fractal_dimension_mean,
##     family = "binomial", data = wdbc.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8037  -0.1436  -0.0355   0.0036   2.9671
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -10.2941    14.9247   -0.69    0.490
## radius_mean             -5.0509     4.4507   -1.13    0.256
## texture_mean             0.4290     0.0826    5.19  2.1e-07 ***
## perimeter_mean           0.3305     0.6174    0.54    0.592
## area_mean                0.0465     0.0204    2.28    0.023 *
## smoothness_mean         87.0870    40.7470    2.14    0.033 *
## compactness_mean       -11.6711    27.0469   -0.43    0.666
## concavity_mean           2.8317     9.9083    0.29    0.775
## concave.points_mean     58.3508    34.3277    1.70    0.089 .
## symmetry_mean           20.2086    12.7647    1.58    0.113
## fractal_dimension_mean -10.8241   102.3592   -0.11    0.916
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 527.365  on 397  degrees of freedom
## Residual deviance:  97.003  on 387  degrees of freedom
## AIC: 119
##
## Number of Fisher Scoring iterations: 9
```

```r
#predict the test set
pred.bc <- predict(glm.bc, newdata = wdbc.test, type = "response")
pred.cancerous <- rep(TRUE, 171)
pred.cancerous[pred.bc < 0.5] <- FALSE
table(wdbc.test$cancerous, pred.cancerous) #confusion matrix
```

```
##        pred.cancerous
##         FALSE TRUE
##   FALSE   105    4
##   TRUE      6   56
```

```r
mean(wdbc.test$cancerous==pred.cancerous) #prediction accuracy
```

```
## [1] 0.942
```

```r
#random forest
wdbc$cancerous <- as.factor(wdbc$cancerous) #ensure type of random forest is "classification"
rf.bc <- randomForest(cancerous ~ radius_mean+texture_mean+perimeter_mean+area_mean+smoothness_mean+comp
```

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

```r
print(rf.bc)
```

```
##
## Call:
##  randomForest(formula = cancerous ~ radius_mean + texture_mean +      perimeter_mean + area_mean + sm
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 0.0457
##                     % Var explained: 80.5
```

```r
#predict the test set
pred.rf.bc <- predict(rf.bc, wdbc.test)
pred.rf.cancerous <- rep(TRUE, 171)
pred.rf.cancerous[pred.rf.bc < 0.5] <- FALSE
table(wdbc.test$cancerous, pred.rf.cancerous) #confusion matrix
```

```
##        pred.rf.cancerous
##         FALSE TRUE
##   FALSE   104    5
##   TRUE      5   57
```

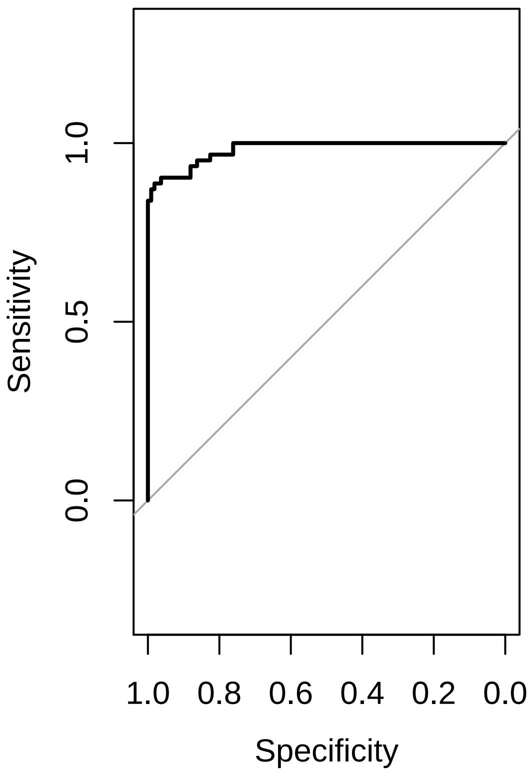
```r
mean(wdbc.test$cancerous==pred.rf.cancerous) #prediction accuracy
```

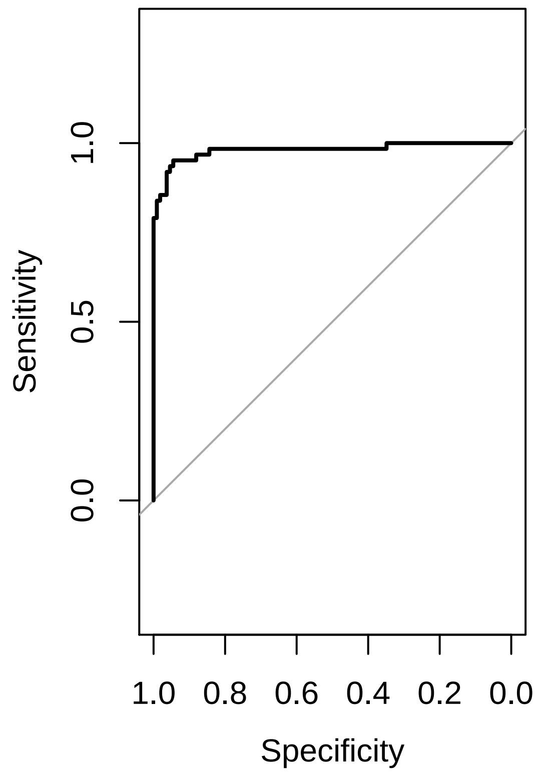
```
## [1] 0.942
```

```r
#roc curves
roc.glm <- roc(wdbc.test$cancerous, pred.bc)
roc.rf <- roc(wdbc.test$cancerous, pred.rf.bc)
par(mfrow = c(1,2))
plot.roc(roc.glm)
```

```
##
## Call:
## roc.default(response = wdbc.test$cancerous, predictor = pred.bc)
##
## Data: pred.bc in 109 controls (wdbc.test$cancerous FALSE) < 62 cases (wdbc.test$cancerous TRUE).
## Area under the curve: 0.982
```

```r
plot.roc(roc.rf)
```






```
##
## Call:
## roc.default(response = wdbc.test$cancerous, predictor = pred.rf.bc)
##
## Data: pred.rf.bc in 109 controls (wdbc.test$cancerous FALSE) < 62 cases (wdbc.test$cancerous TRUE).
## Area under the curve: 0.98
```

(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.

This dataset contains patient ID, diagnosis(benign or malignant), mean, standard error, and "worst" or largest (mean of the three largest values) of radius (mean of distances from center to points on the perimeter), texture (standard deviation of gray-scale values), perimeter, area, smoothness (local variation in radius lengths), compactness (perimeter^2 / area - 1.0), concavity (severity of concave portions of the contour), concave points (number of concave portions of the contour), symmetry, fractal dimension ("coastline approximation" - 1) of the tumor.

Reference:archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.names

(b) Tidy the data, ensuring that each variable is properly named and cast as the correct data type. Discuss any missing data. There is no missing data.

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

(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.

The prediction accuracy is 0.94. False positive rate is 0.035. False negative rate is 0.023. Thus I think the model performance is pretty great.

(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.

The prediction accuract is 0.94. False positive rate is 0.029. False negative rate is 0.029. Thus the model performance is also pretty great.

(f) Compare the models from part (d) and (e) using ROC curves. Which do you prefer? Be sure to justify your preference. Regression model gives an AUC of 0.9822, random forest gives an AUC of 0.9805. I think both are good.

Problem 4

Problem 4 (15 pts) 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.

1. Metastasis of cancer can be predicted based on the size, location of the tumor, genetic profile of the patients and other risk factors. This is helpful because doctors can make more evidence based decisions regarding treatments for cancer patients. The goal is prediction.
2. Whether a candidate will win the election, for example presidencial election, can be predicted using current voting statistics, candicate's likability from the public pulled from social media, candidate's characteristics that were shown to be predictive from past trend, etc. The goal is prediction.
3. Predicting whether a student can past a certain test can be predicted based on the time he or she invested in studying, how long before the test did he or she start studying, GPA, past scores of midterms or similar tests in another subject, etc. The goal is prediction.

(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.

1. When a new product is about to be introduced to the market, researchers could predict how much sale can be expected given the current economy, competitor products sales, advertisement, location of the retail stores etc. Company can prepare the products based on the predicted amounts that's likely to be sold. The goal is prediction
2. Salary can be predicted when someone is looking for another job, based on the location of the new job, mean salary of the particular position, the job seeker's education and experience level etc. This can be helpful for people to decide whether they should relocate, seek further education for higher pay, also to learn what can be expected when they make certain career decision. The goal is prediction.
3. Flights, trains and buses dispatching, especially during the holiday seasons, should be predicted to minimize the trouble people might face when there isn't enough transpotation services available. Also, extra dispatching should be minimized so that there isn't too many staff on the jobs but don't have to be. The goal is 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?

Flexible approach can find non-linear relationships, produces less error/bias from fitting the data. But it's also more likely to cause higher variance when use different training sets. When there are a lot of data points for training and a small variance of error when a linear model is fitted, a more flexible approach is preferred. Although flexible, the large training set should be able to produce models with relatively high generalizibility, using a more flexible approach will more likely to increase model performance without causing overfitting. When there is not enough data for training and/or there are a large number of predictors, it's better to use a less flexible approach to account for the potential large variance and avoid overfitting.

Problem 5 (10 pts) 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).

(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.

Answer iii is correcr. Starting salary after graduation = 50 + 20 x GPA + 0.07 x IQ + 35 x Gender + 0.01 x GPA x IQ - 10 x GPA x Gender. So when the IQ and GPA are fixed, only 35 x Gender and -10 x GPA x Gender determines the model outcome. Because female is 1, male is 0, both items would be zeros for male. As for female, when GPA is equal to 3.5, the sum of two items equals zero; when GPA is larger than 3.5, the sum of two items is a negative number which is smaller than that for male (zero). Thus, males earn more on average than males when they have the same IQ and GPA.

(b) Predict the salary of a female with IQ of 110 and a GPA of 4.0. 50 + 20 x 4.0 + 0.07 x 110 + 35 x 1 + 0.01 x 4.0 x 110 - 10 x 4.0 x 1 = 137 The salary of a female with IQ of 110 and a GPA of 4.0 is approximately 137,000.

(c) True or false: Since the coefficient for the GPA/IQ interaction term is very small, there is little evidence of an interaction effect. Justify your answer. False. We don't have any information on the p-value from the FF-test to determine this variable's significance in predicting the outcome, thus it's inconclusive whether there is an interaction between GPA and IQ that affects the salary.

Extra Credit Problem 6

Problem 6

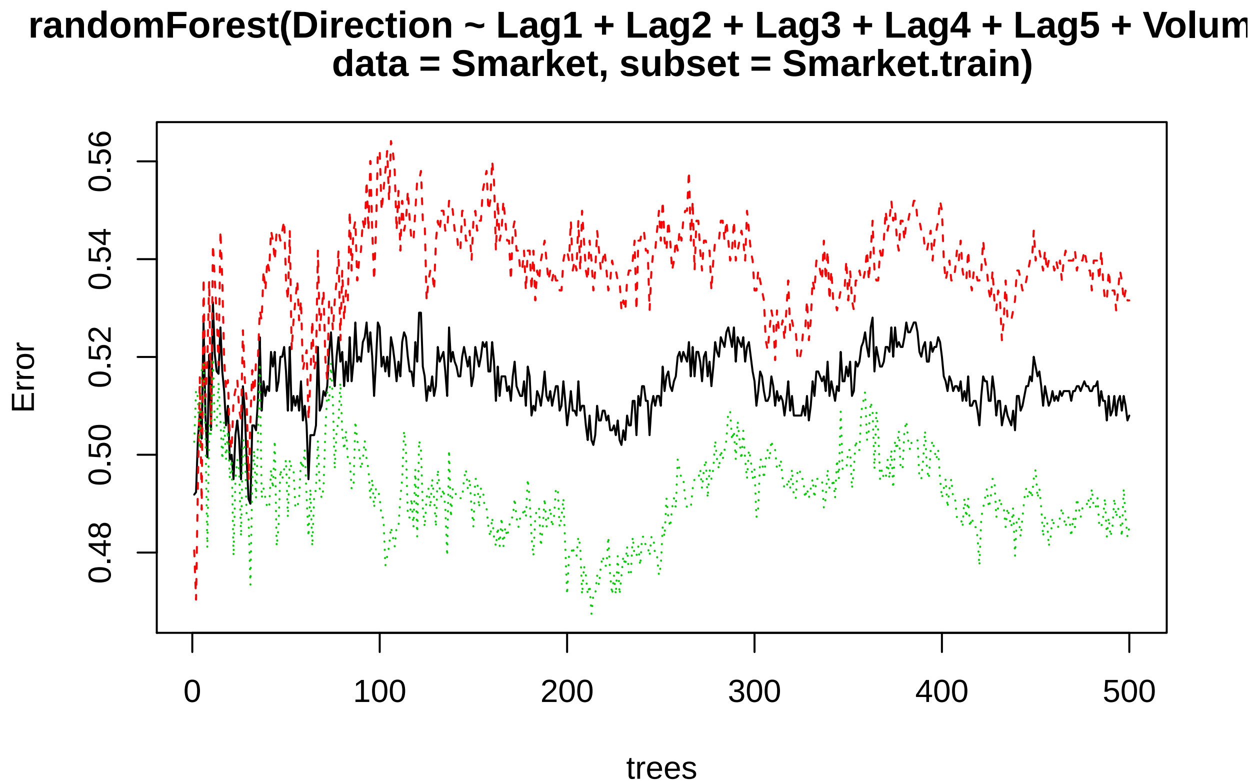
```
#Split dataset Smarket into train and testset
attach(Smarket)
Smarket.train <- (Year<2005) #extract Smarket's data before 2005
Smarket.2005 <- Smarket[!Smarket.train,] #extract Smarket's data after 2005
dim(Smarket.2005) #see how many data points are included
```

```
## [1] 252   9
```

```
#Random Forest Model(Comparing with week8b lab which used a logistic regression model)
rf.sm <- randomForest(Direction ~ Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=Smarket, subset = Smarket.train)
rf.sm
```

```
##
## Call:
##  randomForest(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 +      Lag5 + Volume, data = Smarket, s
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 51.7%
## Confusion matrix:
##      Down  Up class.error
## Down  218 273       0.556
## Up    243 264       0.479
```

```
plot(randomForest(Direction ~ Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=Smarket, subset = Smarket.train))
```




```r
pred.sm <- predict(rf.sm, newdata = Smarket.2005)
#check model accuracy
table(pred.sm, Smarket.2005$Direction)
```

```
##
## pred.sm Down Up
##    Down   55 72
##    Up     56 69
```

```r
mean(pred.sm==Smarket.2005$Direction)
```

```
## [1] 0.492
```

(a) How accurate are the results compared to simple methods like linear or logistic regression? The prediction accuracy from random forest model is 0.52, whereas the logistic regression model trained with the same set of predictors had a prediction accuracy of 0.48.