# IMT 573 Final Exam

*Yash Manish Raichura*

*Due: December 10, 2019*

### Instructions

This is a take-home final examination. You may use your computer, books/articles, notes, course materials, etc., but all work must be your own! References must be appropriately cited. Please justify your answers and show all work; a complete argument must be presented to obtain full credit. Before beginning this exam, please ensure you have access to R and RStudio; this can be on your own personal computer or on the IMT 573 R Studio Server.

1. Download the `final_exam.rmd` file from Canvas or save a copy to your local directory on RStudio Server. Open `final_exam.rmd` in RStudio and supply your solutions to the exam by editing `final_exam.rmd`.
2. Replace the "Insert Your Name Here" text in the `author:` field with your own full name.
3. Be sure to include well-documented (e.g. commented) code chucks, figures, and clearly written text chunk explanations as necessary. Any figures should be clearly labeled and appropriately referenced within the text. Be sure that each visualization adds value to your written explanation; avoid redundancy – you do not need four different visualizations of the same pattern.
4. **Collaboration is not allowed on this exam.** You may only speak with the Instructor (Lavi Aulck) and the TA (Varun Panicker) about this material.
5. All materials and resources that you use (with the exception of lecture slides) must be appropriately referenced within your assignment.
6. Remember partial credit will be awarded for each question for which a serious attempt at finding an answer has been shown. Students are *strongly* encouraged to attempt each question and to document their reasoning process even if they cannot find the correct answer. If you would like to include R code to show this process, but it does not run without errors, you can do so with the `eval=FALSE` option. (Note: I am also using the `include=FALSE` option here to not include this code in the PDF, but you need to remove this or change it to `TRUE` if you want to include the code chunk.)
7. When you have completed the assignment and have **checked** that your code both runs in the Console and knits correctly when you click `Knit PDF`, rename the knitted PDF file to `YourLastName_YourFirstName.pdf`, and submit BOTH your RMarkdown and PDF files on Canvas.

### Statement of Compliance

You **must** include the a "signed" Statement of Compliance in your submission. The Compliance Statement is found on the next page of this exam. You must include this text, word-for-word, in your final exam submission. Adding your name indicates you have read the statement and agree to its terms. Failure to do so will result in your exam **not** being accepted.

**Statement of Compliance**

I affirm that I have had no conversation regarding this exam with any persons other than the instructor (Lavi Aulck) and TA (Varun Panicker). Further, I certify that the attached work represents my own thinking. Any information, concepts, or words that originate from other sources are cited in accordance with University of Washington guidelines as published in the Academic Code (available on the course website). I am aware of the serious consequences that result from improper discussions with others, sharing this exam, or from the improper citation of work that is not my own. The above also pertains to this exam after my enrollment in the course is completed.

(Yash Manish Raichura)

(12-09-2019)

**Setup**

In this exam you will need, at minimum, the following R packages.

```r
#install.packages('tidyverse')
#install.packages('mice')
#install.packages('AER')
#install.packages('bestglm')
#install.packages('ggpubr')
#install.packages('leaps')
#install.packages('ggcorrplot')
#install.packages('manip')
#install.packages('MASS')
#install.packages('caret')
#install.packages('caTools')
#install.packages('gbm')
#install.packages('randomForest')
#install.packages('ISLR')
#install.packages('AER')
#install.packages('rpart')
library(rpart)
library(AER)
library(ISLR)
library(caret)
library(randomForest)
library(gbm)
library(MASS)
library(caTools)
library(MASS, quietly = TRUE)
library(MASS)
library(ggplot2)
library(ggpubr)
library(tidyverse)
library(bestglm)
library(leaps)
library(ggcorrplot)
library(tidyverse)
```

## Problem 1

(15 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.

```
affairs <- data(Affairs)
affairs <- Affairs
#View(affairs)
```

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 (Hillingshead 7-point classification with reverse numbering), and a numeric self-rating of their marriage (from 1=very unhappy to 5=very happy).

```
?Affairs
```

```
## starting httpd help server ... done
```

```
head(affairs)
```

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

```
summary(affairs)
```

```
##     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
```

```
str(affairs)
```

```
## 'data.frame':    601 obs. of  9 variables:
```

```
##  $ affairs      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ gender       : Factor w/ 2 levels "female","male": 2 1 1 2 2 1 1 2 1 2 ...
##  $ age          : num  37 27 32 57 22 32 22 57 32 22 ...
##  $ yearsmarried : num  10 4 15 15 0.75 1.5 0.75 15 15 1.5 ...
##  $ children     : Factor w/ 2 levels "no","yes": 1 1 2 2 1 1 1 2 2 1 ...
##  $ religiousness: int  3 4 1 5 2 2 2 2 4 4 ...
##  $ education    : num  18 14 12 18 17 17 12 14 16 14 ...
##  $ occupation   : int  7 6 1 6 6 5 1 4 1 4 ...
##  $ rating       : int  4 4 4 5 3 5 3 4 2 5 ...
```

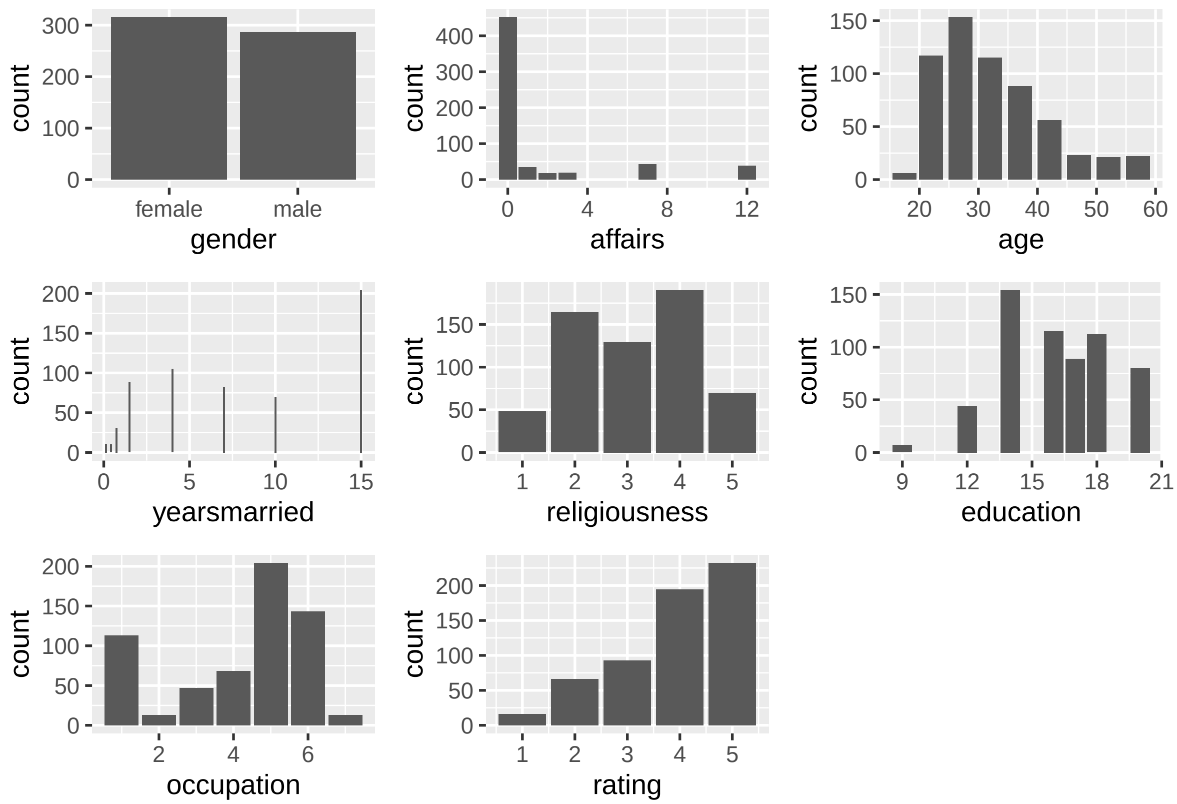
(a) Describe the participants. Use descriptive, summarization, and exploratory techniques to describe the participants in the study. For example, what proportion of respondents are female? What is the average age of respondents? In your response comment on any ethical and privacy concerns you have with this dataset.

The dataset is a survey conducted by Psychology Today in 1969. The results of the survey cannot be corroborated and we do not know how truthful the participants have been in answering the questions.

```r
#Checking for NA values in the dataset
sum(is.na(affairs))
```

```
## [1] 0
```

```r
a <- ggplot(data = affairs) + geom_bar(mapping = aes(x = gender))
b <- ggplot(data = affairs, mapping =aes(x = affairs)) + geom_bar()
c <- ggplot(data = affairs, mapping =aes(x = age)) + geom_bar()
d <- ggplot(data = affairs, mapping =aes(x = yearsmarried)) + geom_bar()
e <- ggplot(data = affairs, mapping =aes(x = religiousness)) + geom_bar()
f <- ggplot(data = affairs, mapping =aes(x = education)) + geom_bar()
g <- ggplot(data = affairs, mapping =aes(x = occupation)) + geom_bar()
h <- ggplot(data = affairs, mapping =aes(x = rating)) + geom_bar()
figure <- ggarrange(a,b,c,d,e,f,g,h)
figure
```




```r
#Number of affairs based on the gender
affairs %>% group_by(affairs,gender) %>% summarize(count=n())
```

```
## # A tibble: 12 x 3
## # Groups:   affairs [6]
##    affairs gender count
##      <dbl> <fct>  <int>
##  1       0 female   243
##  2       0 male     208
##  3       1 female    15
##  4       1 male      19
##  5       2 female     7
##  6       2 male      10
##  7       3 female     8
##  8       3 male      11
##  9       7 female    22
## 10       7 male      20
## 11      12 female    20
## 12      12 male      18
```

```r
#Proportion of males and females in the participant pool
summary(affairs$gender)
```

```
## female   male
##    315    286
```

```r
#Average age of the pariticipant pool
mean(affairs$age)
```

```
## [1] 32.48752
```

```r
#Average age and number of affairs per gender

#1. Average age and number of affairs of males
men_age<- filter(affairs, gender == 'male')
mean(men_age$age)
```

```
## [1] 34.34441
```

```r
mean(men_age$affairs)
```

```
## [1] 1.496503
```

```r
# 2. Average age and number of affairs of females
female_age<- filter(affairs, gender == 'female')
mean(female_age$age)
```

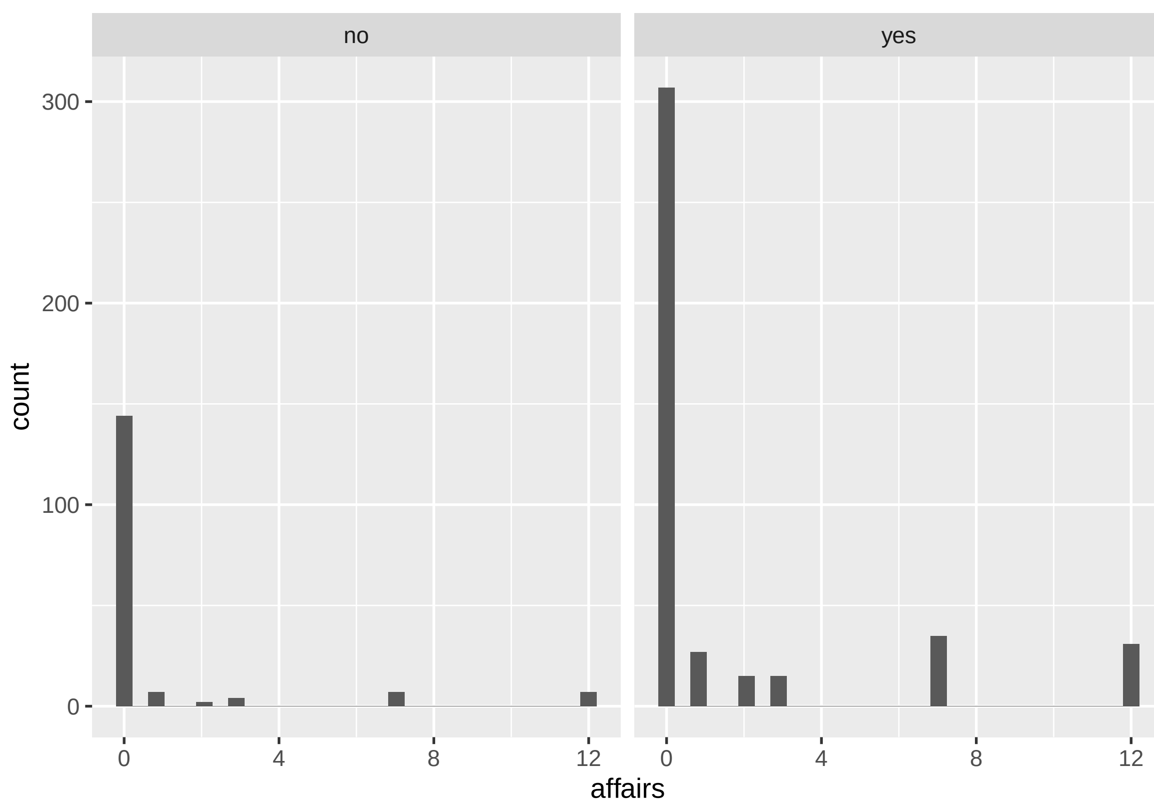
```
## [1] 30.80159
```

```r
mean(female_age$affairs)
```

```
## [1] 1.419048
```

```r
#Number of affiars based on whether the participants had children or not
ggplot(affairs) + geom_histogram(aes(x=affairs)) + facet_wrap(~ children)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




(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, consider the binary outcome - had an affair versus didn't have an affair. Create a new variable to capture this response variable of interest.

What might the advantages and disadvantages of this approach to modeling the data be in this context?

Binary column 'affair' created below based on number of affairs. The column value is 1 if the participant has had an affair, else it is set to 0. If only the binary variable is taken to model the data, the model prediction will only be limited to understanding the factors related to whether the participant will engage in an affair or not. No inferences would be made about the number of affairs.

```
colnames(affairs)[1] <- "number_of_affairs"
affairs$affair <- NA
affairs$affair[affairs$number_of_affairs > 0] <- 1
affairs$affair[affairs$number_of_affairs == 0] <- 0
sum(is.na(affairs$affair))
```

```
## [1] 0
```

(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.

Since we have a binary variable created of whether the participant has had an affair or not, we can use logistic regression.

```
#Fully fitted model

fit1 <- glm(affair ~ gender + age + yearsmarried + children + religiousness + education + occupatio
                data=affairs,family=binomial())

summary(fit1)
```

```
##
## Call:
## glm(formula = affair ~ 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
```

As we see above, p-value is quite large fro gender, age, children, education and occupation as compared to critical value (0.05 in this case). Hence, these variables are not as much statistically significant. However, it seems that rating, number of years married and religiousness have an impact on a person having an affair. Looking at Beta values for these 3 variables, we can say that as the number of years increases of the participant in the marriage, the liklier is the probability of having an affair. The same goes for rating. The higher the rating (i.e. happier participant in the marriage), the chances of having an affair decreases since the beta values are positive. Surprisingly, the same is valid for having children too. People having children have higher probabilities of being involved in an affair.

(d) Use an all subsets model selection procedure to obtain a "best" fit model. Note that an all subsets model selection is not the same as forward/backward selection. 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.

Xy dataframe was created consisting of the explanatory variables and the response variable at the end for best subset model selection using bestglm() function. Here we see that age and gender (male) have also been included in the model. The coefficient for age is negative which shows that people lower than the mean age have higher probability of being involved in an affair. Whereas, men are more likely to have an affair with Beta1 being approximately 0.06. If a different information criteria, BIC is used, the model explanatory variables are statistically the same as generated by the glm function.

```
Xy <- affairs[,2:10]

fit2 <- bestglm(Xy, IC = 'AIC')
```

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

```
fit2
```

```
## AIC
## BICq equivalent for q in (0.821508156450582, 0.932574998701236)
## Best Model:
##                  Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept)    0.82158177 0.103299583  7.953389 9.179310e-15
## gendermale     0.06360652 0.034902126  1.822425 6.889227e-02
## age           -0.00739692 0.002988183 -2.475390 1.358642e-02
## yearsmarried   0.01859607 0.004970389  3.741371 2.007405e-04
## religiousness -0.05442460 0.014815335 -3.673531 2.607919e-04
## rating        -0.08759874 0.015763910 -5.556917 4.146818e-08
```

```
bestglm(Xy, IC = 'BIC')
```

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

```
## BIC
## BICq equivalent for q in (0.272922984053981, 0.743709326574774)
## Best Model:
##                  Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept)    0.68930455 0.082442762  8.361007 4.390557e-16
## yearsmarried   0.00928703 0.003211861  2.891479 3.973887e-03
## religiousness -0.05594171 0.014871389 -3.761701 1.853946e-04
## rating        -0.08681213 0.015834219 -5.482564 6.193727e-08
```

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

The logistic regression coefficients give a change in the log odds of the outcome for a unit increase in the predictor variable. The intercept value is 0.82 As shown in the output of bestglm() function, for every unit change in age, log odds of having an affair decreases by 0.07. For every unit increase in yearsmarried, the log odds of having an affair increases by 0.018. For every unit increase in religiousness, the log odds of having an affair decreases by 0.05. For every unit increase in rating, the log odds of having an affair decreases by 0.08. The variable gendermale is 0 when the participant is female and 1 when the participant is male. Thus, if we are predicting females involved in a n affair, we get: affairs = 0.82158177 + 0.06360652 x 0 For gender = male, the equation changes to affairs = 0.82158177 + 0.06360652 x 1

The p-value of all these parameters is less than 0.05 which make the results statistically significant.

(f) Create an artificial test dataset where marital 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.

Since gender and children are factors, we are not considering a part of the model to be validated on the test dataset. If we convert these factor variables to 1's and 0's i.e 1 for males and 0 for females, and 1 for yes and 0 for no in children, the mean of the gender variable comes up to be 1.47 which is a incorrect, making the model output biased. Hence, these 2 variables have not been considered. Also, these variables were not statistically significant and in turn will not affect the model.

As we can see, by taking means of all the other predictor variables, the probability of having an affair decreases with how happy the participant is with his/her marriage. Hence, the model is prediciting correctly.

```r
#affairs$gender <- as.character(affairs$gender)
#affairs$gender[affairs$gender == 'male'] <- '1'
#affairs$gender[affairs$gender == 'female'] <- '0'
#affairs$gender <- as.numeric(affairs$gender)
#affairs$gender <- as.factor(affairs$gender)
#mean(affairs$gender)
#str(affairs)

fit3 <- glm(affair ~ age + yearsmarried + religiousness + education + occupation +rating,
                data=affairs,family=binomial())

summary(fit3)
```

```
##
## Call:
## glm(formula = affair ~ age + yearsmarried + religiousness + education +
##     occupation + rating, family = binomial(), data = affairs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6302  -0.7546  -0.5727  -0.2787   2.4410
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.32620    0.85389   1.553 0.120393
## age           -0.04105    0.01794  -2.288 0.022138 *
## yearsmarried   0.10617    0.02949   3.600 0.000318 ***
## religiousness -0.32024    0.08958  -3.575 0.000351 ***
## education      0.03615    0.04977   0.726 0.467571
## occupation     0.04689    0.06659   0.704 0.481292
```

```
## rating        -0.47870    0.09050  -5.289 1.23e-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: 613.17  on 594  degrees of freedom
## AIC: 627.17
##
## Number of Fisher Scoring iterations: 4
```

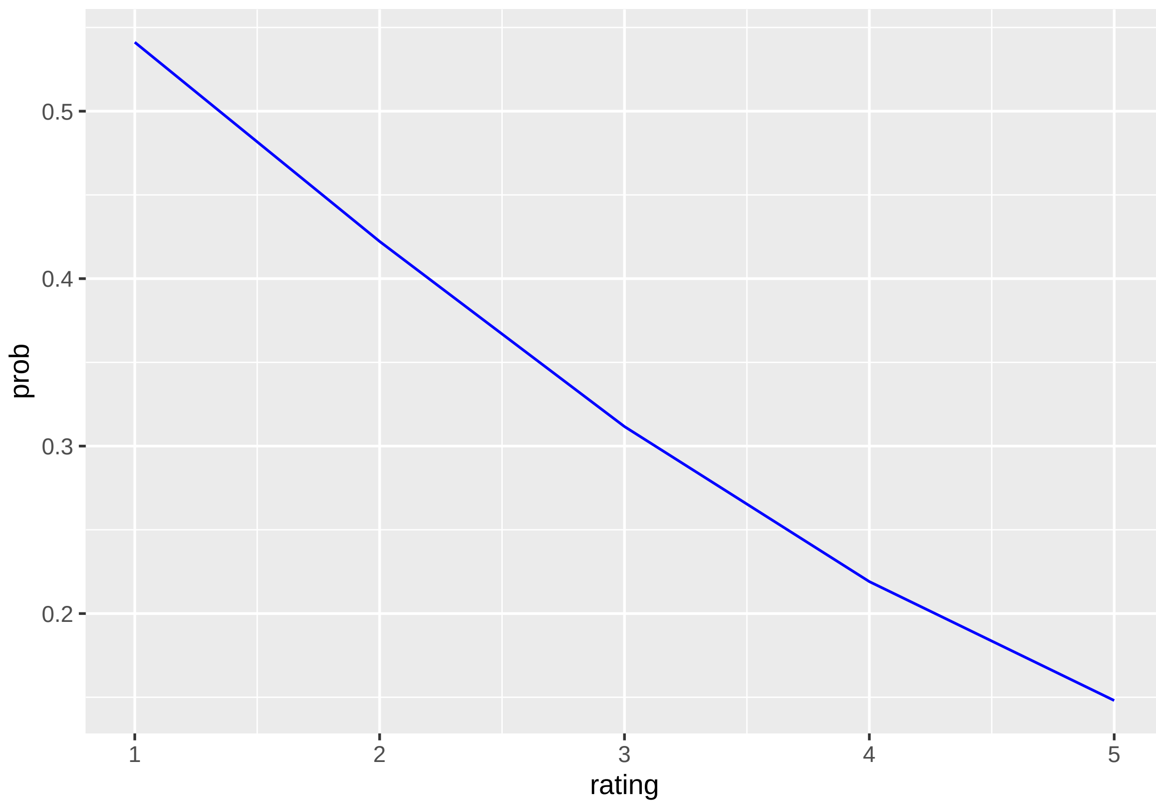
```r
test <- data.frame(rating = c(1, 2, 3, 4, 5), age = mean(affairs$age),yearsmarried =
                     mean(affairs$yearsmarried),
religiousness = mean(affairs$religiousness), education = mean(affairs$education),
education = mean(affairs$education), occupation = mean(affairs$occupation))

View(test)
head(test)
```

```
##   rating      age yearsmarried religiousness education education.1 occupation
## 1      1 32.48752     8.177696      3.116473  16.16639    16.16639   4.194676
## 2      2 32.48752     8.177696      3.116473  16.16639    16.16639   4.194676
## 3      3 32.48752     8.177696      3.116473  16.16639    16.16639   4.194676
## 4      4 32.48752     8.177696      3.116473  16.16639    16.16639   4.194676
## 5      5 32.48752     8.177696      3.116473  16.16639    16.16639   4.194676
```

```r
#Probability outcomes of having an affair
test$prob <- predict(fit3, test, type="response")
test$prob
```

```
## [1] 0.5411718 0.4222255 0.3116645 0.2190769 0.1480776
```

```r
#Ranking vs Probability of having an affair
ggplot(data = test, mapping = aes(x=rating, y = prob)) + geom_line(color='blue')
```




(g) Reflect on your analysis in this problem. After completing all the parts of this analysis what remaining and additional ethical and privacy concerns do you have?

The entire survey depends on individual perception. For some, the rating 4 might mean they are very happy with the marriage and for some that might not be the case. Also, inclusion of religion into the dataset might raise some ethical concerns. Ideally, it should have no impact on whether an individual is involved in an affair or not.

**Problem 2** (10 pts)

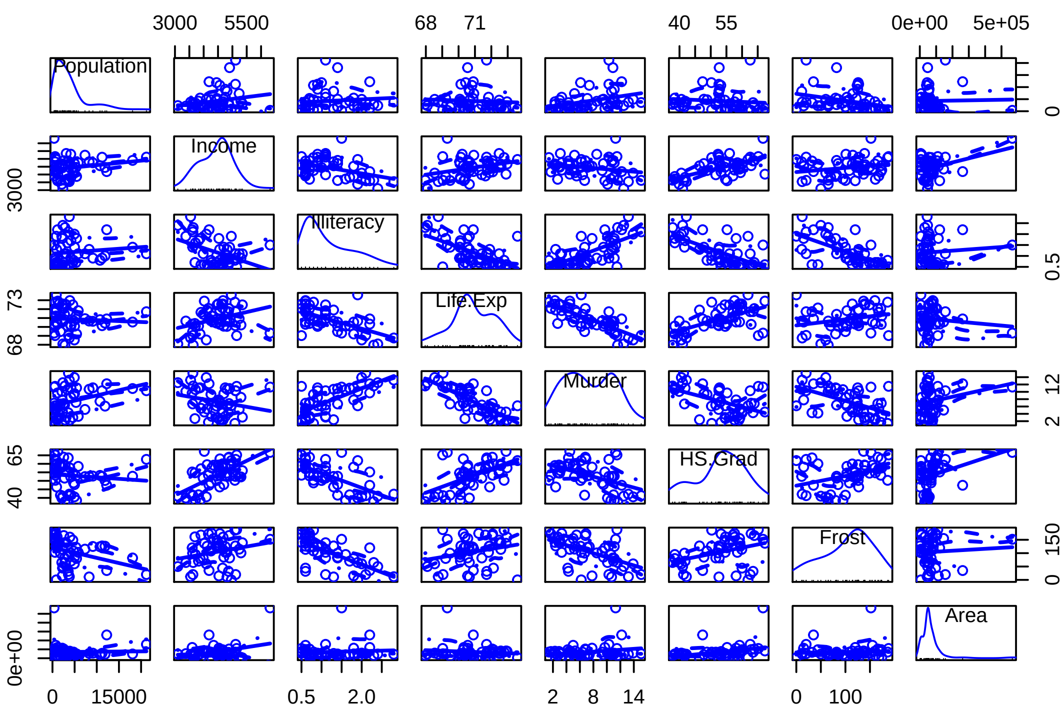
In this problem we will revisit the `state` dataset. This data, available as part of the base **R** package, contains various data related to the 50 states of the United States of America.

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.

(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.

By calculating correlation coefficient, we see that linear relationships cannot be assumed for all covariate relationships. For example, in the plot of Income v/s Murder, correlation is equal to -0.23 which suggests that the bivariate relationship is not completely linear. Although, on the other end, the correlation between Illiteracy and Murder is 0.70 which suggests that the two variables have a strong positive linear relationship i.e. more the illiteracy in the state, more is the count for murder. Also, in the correlation matrix, we can plot the correlation between all the numeric variables which give us a glimpse of the relationship the bivariate variables share.

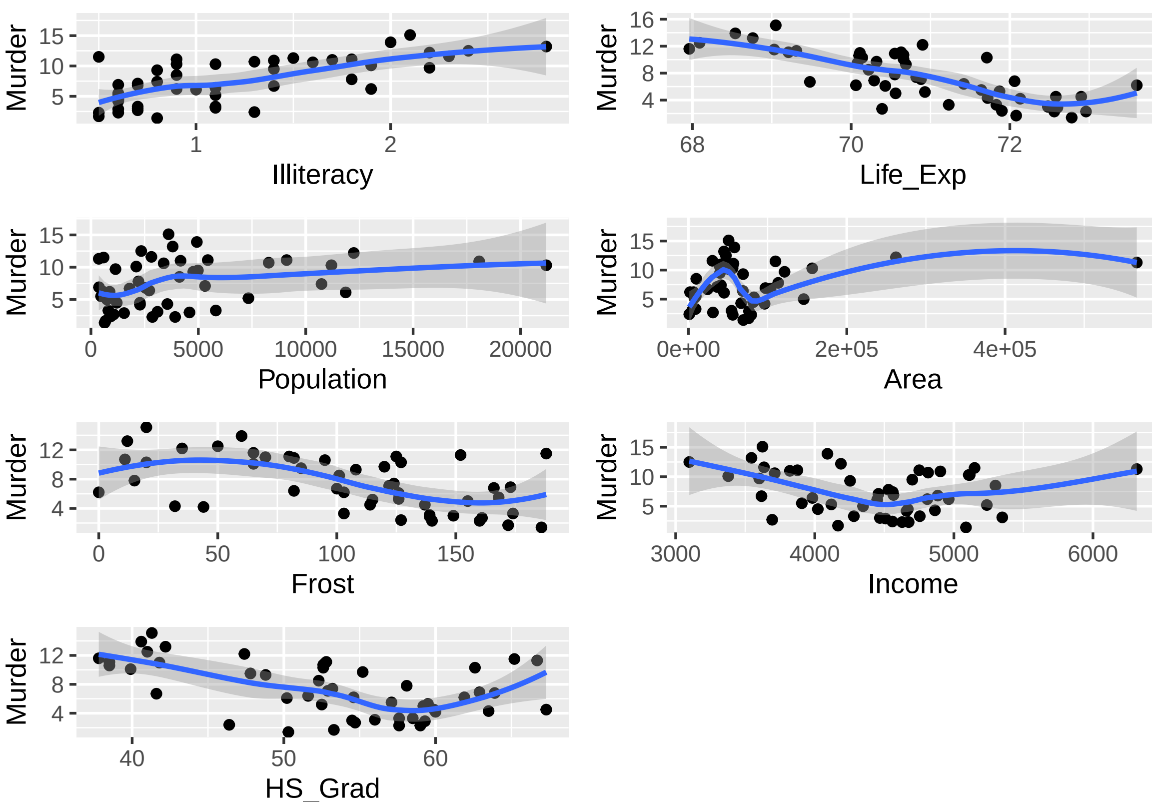
```
#data(state)
#View(state.x77)
states <- cbind(state.x77, state.area, state.name)
#View(states)
states <- tbl_df(states)

#scatterplotMatrix
scatterplotMatrix(state.x77)
```



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

#Changing the column names to remove spaces
colnames(state.x77)[colnames(state.x77)=="Life Exp"] <- "Life_Exp"
colnames(state.x77)[colnames(state.x77)=="HS Grad"] <- "HS_Grad"

#Plotting bivariate relationships
a1 <- ggplot(state.x77, aes(x=Illiteracy, y=Murder)) + geom_point()+geom_smooth()
b1 <- ggplot(state.x77, aes(x=Life_Exp , y=Murder)) + geom_point() +geom_smooth()
c1 <- ggplot(state.x77, aes(x = Population, y = Murder)) + geom_point() + geom_smooth()
d1 <- ggplot(state.x77, aes(x = Area, y = Murder)) + geom_point() + geom_smooth()
e1 <- ggplot(state.x77, aes(x = Frost, y = Murder)) + geom_point() + geom_smooth()
f1 <- ggplot(state.x77, aes(x = Income, y = Murder)) + geom_point() + geom_smooth()
g1 <- ggplot(state.x77, aes(x = HS_Grad, y = Murder)) + geom_point() + geom_smooth()
figure <- ggarrange(a1,b1,c1,d1,e1,f1,g1, ncol=2, nrow=4)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```r
figure
```

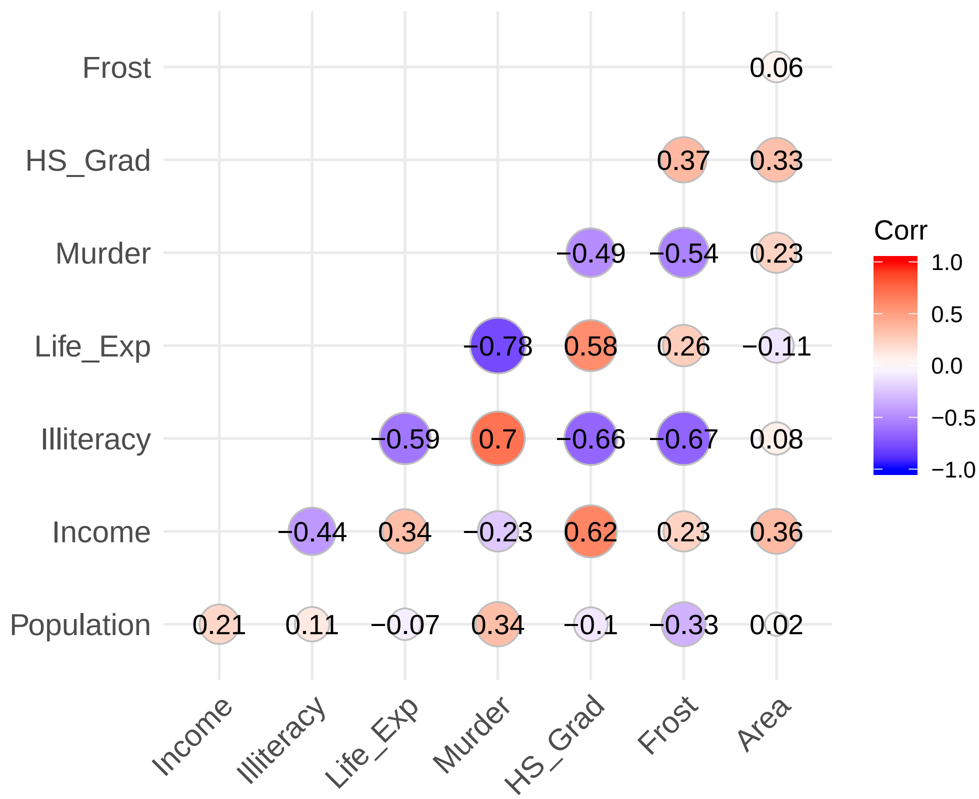



```r
#ggpairs(state.x77)
```

```r
s <- state.x77 %>% select_if(is.numeric)
corr <- cor(s)
corr
```

```
##              Population     Income  Illiteracy    Life_Exp     Murder
## Population  1.00000000  0.2082276  0.10762237 -0.06805195  0.3436428
## Income      0.20822756  1.0000000 -0.43707519  0.34025534 -0.2300776
## Illiteracy  0.10762237 -0.4370752  1.00000000 -0.58847793  0.7029752
## Life_Exp   -0.06805195  0.3402553 -0.58847793  1.00000000 -0.7808458
## Murder      0.34364275 -0.2300776  0.70297520 -0.78084575  1.0000000
## HS_Grad    -0.09848975  0.6199323 -0.65718861  0.58221620 -0.4879710
## Frost      -0.33215245  0.2262822 -0.67194697  0.26206801 -0.5388834
## Area        0.02254384  0.3633154  0.07726113 -0.10733194  0.2283902
##                 HS_Grad      Frost        Area
## Population -0.09848975 -0.3321525  0.02254384
## Income      0.61993232  0.2262822  0.36331544
## Illiteracy -0.65718861 -0.6719470  0.07726113
## Life_Exp    0.58221620  0.2620680 -0.10733194
## Murder     -0.48797102 -0.5388834  0.22839021
## HS_Grad     1.00000000  0.3667797  0.33354187
## Frost       0.36677970  1.0000000  0.05922910
## Area        0.33354187  0.0592291  1.00000000
```

```r
ggcorrplot(corr, lab = TRUE, type = "lower", method="circle")
```




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

In the below fitted linear regression model, we see that the variables - Population, Life_Exp and Income are statistically significant and impact murder across the states. The adjusted R square value is 0.7763. With every unit increase in population, murder increases by 0.000188. For every unit increase in income, there is a decrease in the murder rate by 0.000159. For every unit increase in Life_Exp, there is a decrease in the murder rate by 1.65486983.

```r
options(scipen=4)

linear_model <- lm(Murder ~ ., data = state.x77)
summary(linear_model)
```

```
##
## Call:
## lm(formula = Murder ~ ., data = state.x77)
##
## 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) 122.180392646  17.886225407   6.831 0.0000000254 ***
## Population    0.000188036   0.000064737   2.905      0.00584 **
## Income       -0.000159207   0.000572530  -0.278      0.78232
## Illiteracy    1.373109504   0.832202602   1.650      0.10641
## Life_Exp     -1.654869830   0.256211567  -6.459 0.0000000868 ***
## HS_Grad       0.032338308   0.057252663   0.565      0.57519
## Frost        -0.012884070   0.007392415  -1.743      0.08867 .
## Area          0.000005967   0.000003801   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
```

(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.

The multiple regression model is not fitted that well since we can see that the mean distance of the residuals is present and it needs to be minimum. The normal Q-Q plot shows that the data is normally distributed to an extent. The statistical assumptions made are - 1. Multi colinearity is followed by all the variables. 2. We also assume that the variables follow a normal distribution. 3. The response variable is a dependent variable and all the other predictor variables are independent variables.

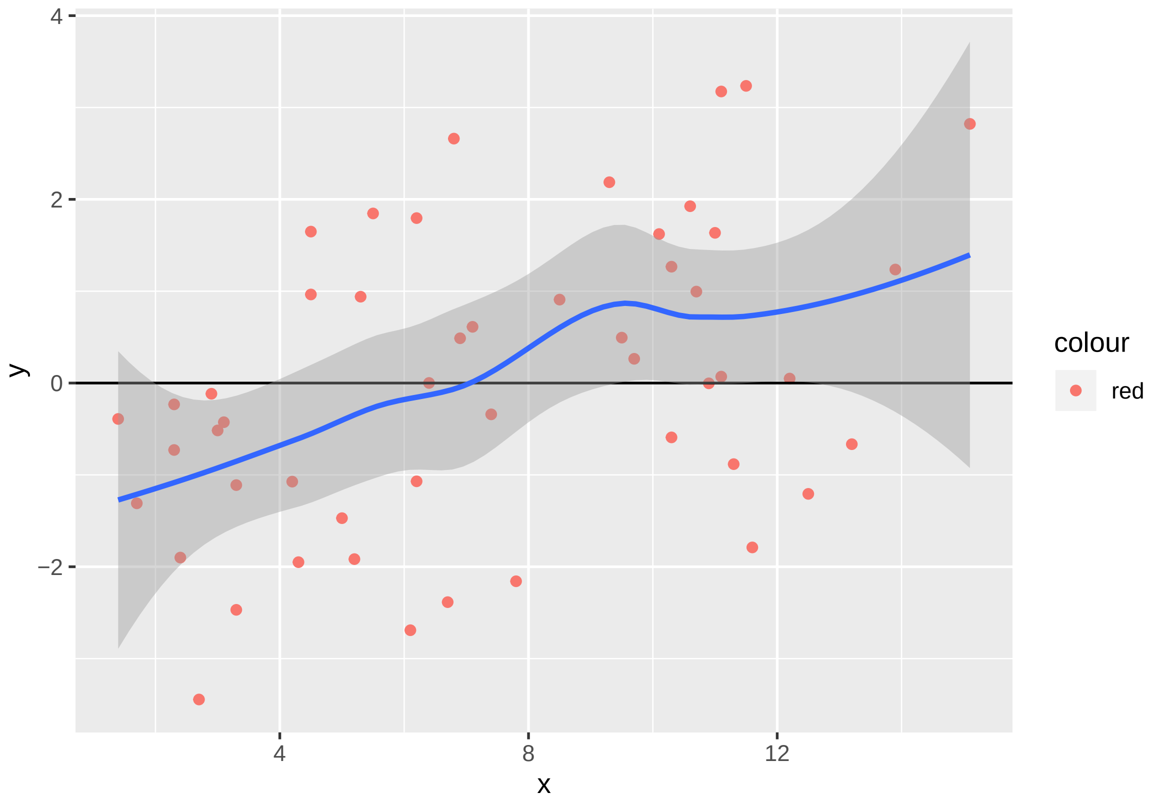
```r
residuals <- resid(linear_model)
residuals
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##   2.8215735739  -0.8829375254  -2.1580413274   1.6220404390  -0.5916537298
##       Colorado    Connecticut       Delaware        Florida        Georgia
##   2.6612510595  -0.4263928760  -1.0696967125   0.9958321590   1.2358454794
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##   1.7959667435   0.9400402071   1.2666252067   0.6120714669  -0.7287616677
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##   0.9644795295   1.9254632602  -0.6657054317  -3.4451929532   0.9082966366
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
```

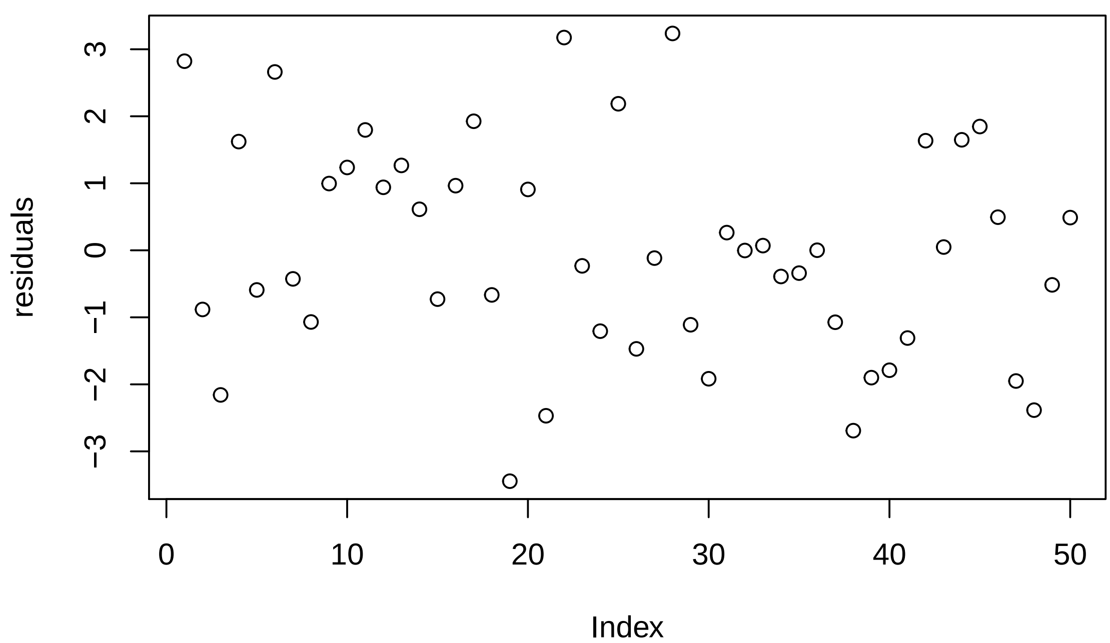
```
##   -2.4691573031   3.1744620358  -0.2323301253  -1.2066351948   2.1864218539
##         Montana       Nebraska         Nevada  New Hampshire     New Jersey
##   -1.4710062626  -0.1165054654   3.2355374028  -1.1110365244  -1.9169703814
##      New Mexico       New York North Carolina   North Dakota           Ohio
##    0.2639137224  -0.0031848118   0.0698818191  -0.3909007668  -0.3407067161
##        Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
##    0.0009867781  -1.0734271766  -2.6911320992  -1.9001703535  -1.7896684899
##    South Dakota      Tennessee          Texas           Utah        Vermont
##   -1.3093864041   1.6354269043   0.0486684451   1.6487319771   1.8465924538
##        Virginia     Washington  West Virginia      Wisconsin        Wyoming
##    0.4943513505  -1.9502682143  -2.3853151126  -0.5158798106   0.4876029322
```

```r
ggplot(data = data.frame(x=state.x77$Murder, y=residuals)) + geom_point(aes(x=x,y=y, color = 'red')
```

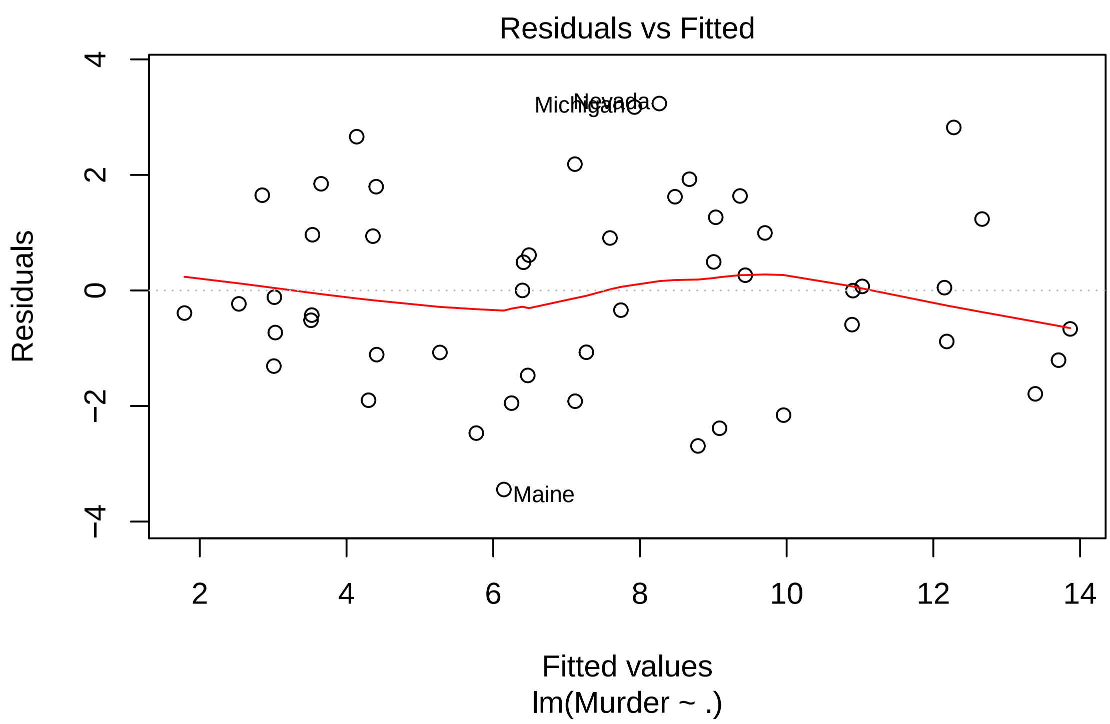
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

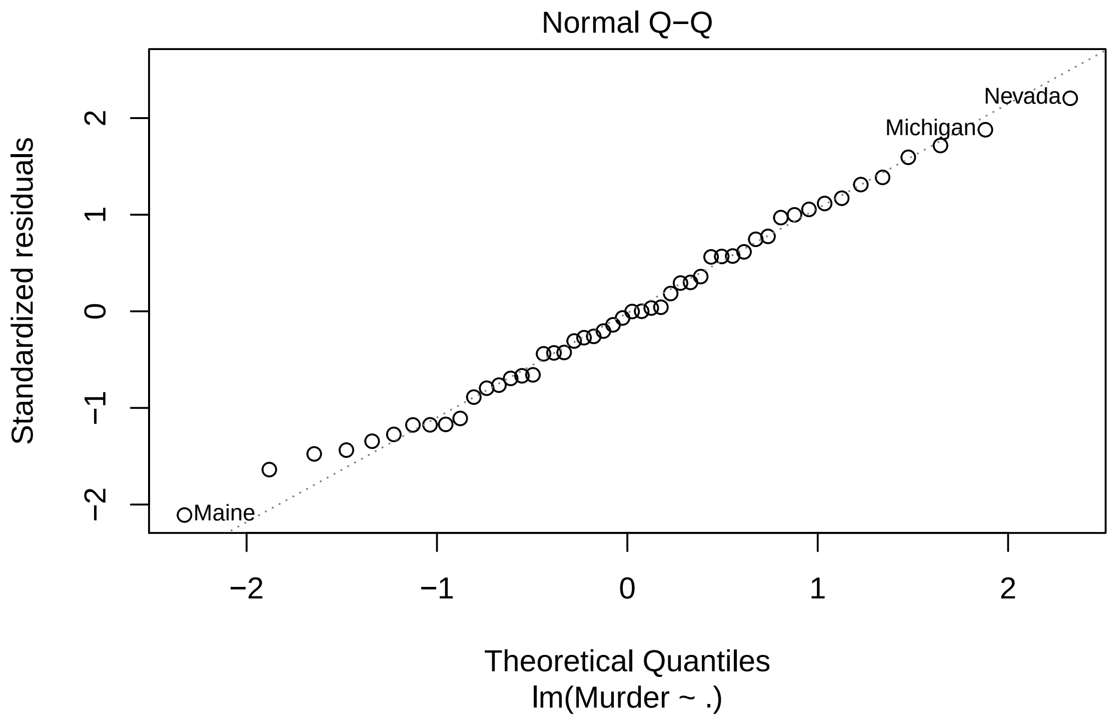


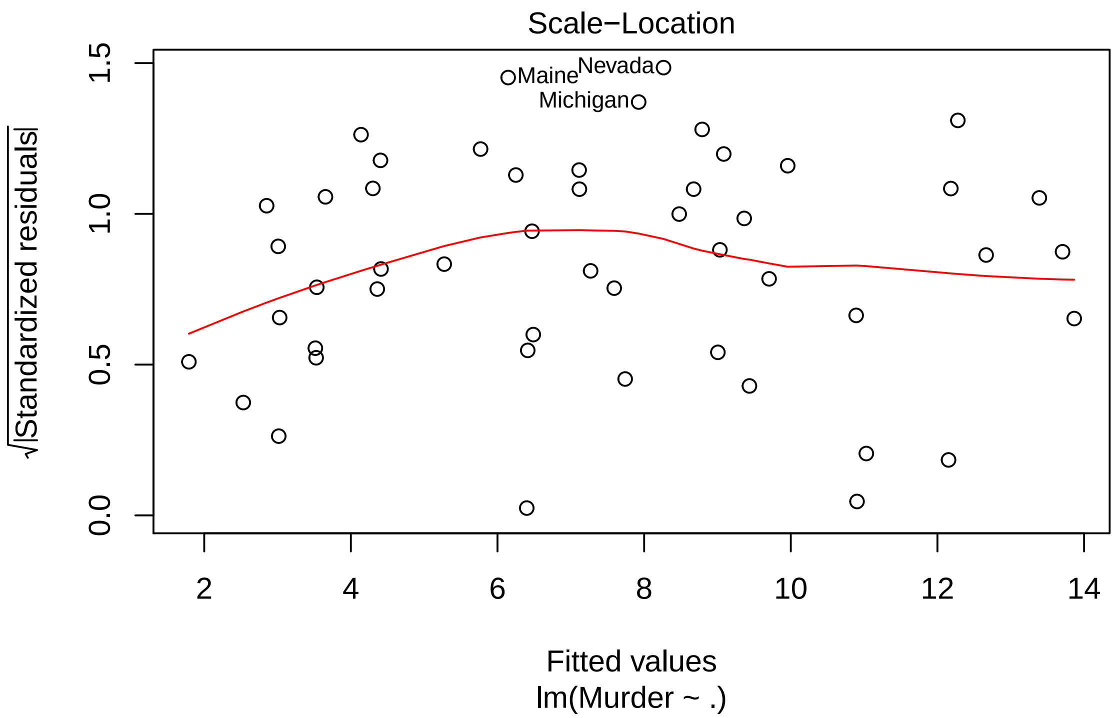

```r
plot(residuals)
```

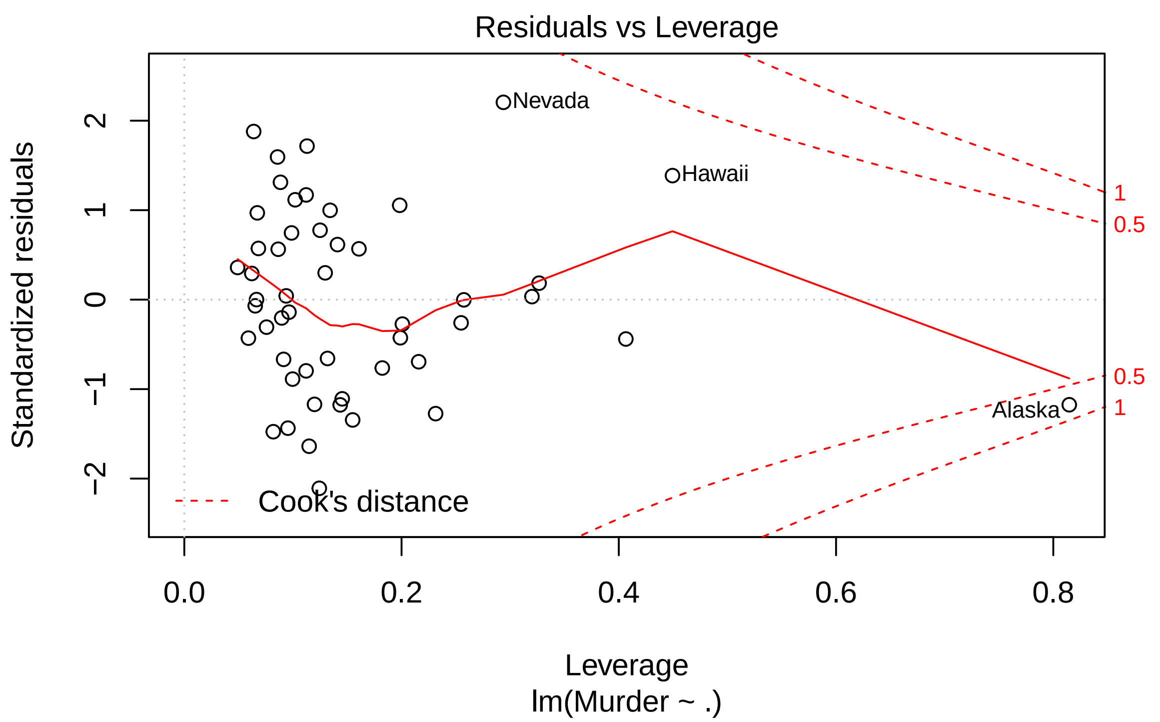



```r
plot(linear_model)
```

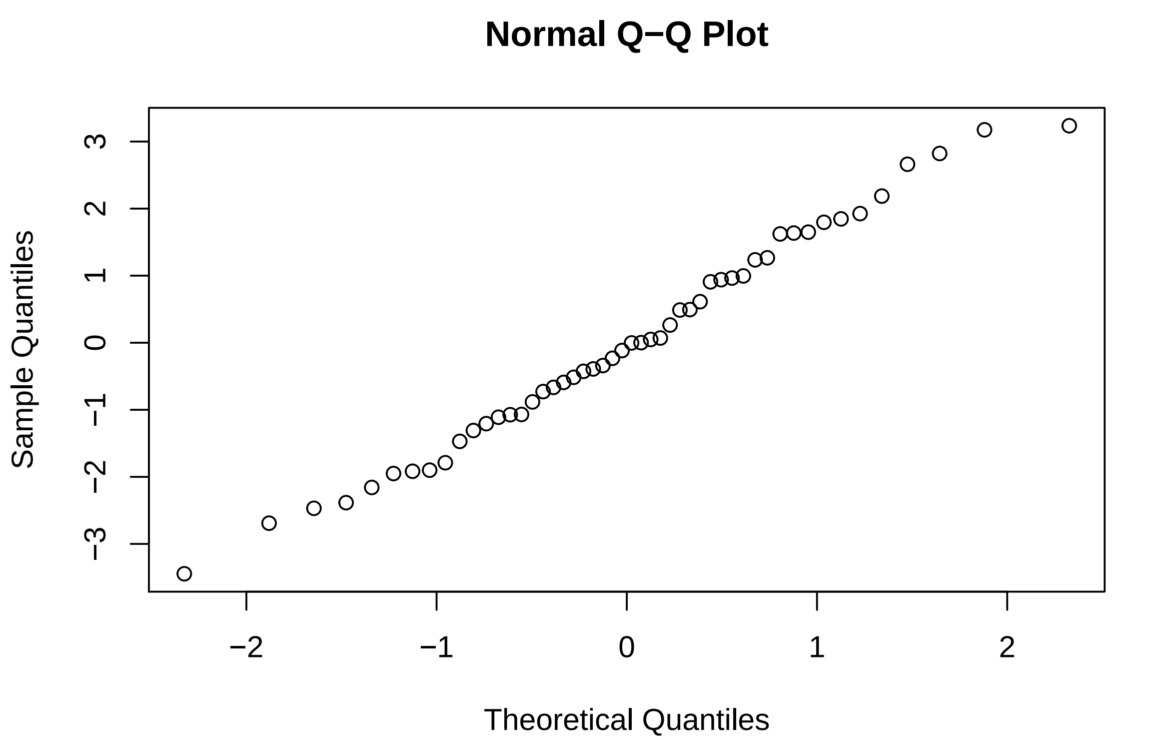



Normal Q−Q
Standardized residuals
−2
−1
0
1
2
Theoretical Quantiles
lm(Murder ~ .)
Maine
Michigan
Nevada


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2
4
6
8
10
12
14
Fitted values
lm(Murder ~ .)
Maine
Nevada
Michigan




```
qqnorm(residuals)
```



(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?

Here, the stepAIC() function was used for model selection. Both the combinations of forward and backward selection models was used and as we can see, that the number of explanatory variables affecting the response variable i.e. Murder have redcued, yet R squared value has increased. The variables Populaton, Illiteracy, Life_Exp, Frost and Area have been incuded this time, out of which Illilteracy is not of great statistical significance, when compared to the critical value which is taken is 0.05 in this case. Also the AIC of this model is lowered and hence we can conclude that it is a better fitting model. The AIC of the linear model was 206 whereas in our selection model, AIC is 203 and hence we can say that the stepwise selection model performs better.

```
selection_model <- stepAIC(linear_model, direction = 'both', trace=FALSE)
summary(selection_model)
```

```
##
## Call:
## lm(formula = Murder ~ Population + Illiteracy + Life_Exp + Frost +
##     Area, data = state.x77)
##
## 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) 120.164031804  17.181610452   6.994 0.0000000117 ***
## Population    0.000177981   0.000059303   3.001      0.00442 **
## Illiteracy    1.172980493   0.680121662   1.725      0.09161 .
## Life_Exp     -1.607836823   0.232377225  -6.919 0.0000000150 ***
## Frost        -0.013730312   0.007079737  -1.939      0.05888 .
## Area          0.000006804   0.000002919   2.331      0.02439 *
## ---
## 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
```

```
AIC(linear_model)
```

```
## [1] 206.9071
```

```
AIC(selection_model)
```

```
## [1] 203.2956
```

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

Cross-validation is basically a form of resampling the data again because we are fitting the same statistical method multiple times on different subsets of the data. In K-fold cross validation, we test model performance against one data point at each iteration. This may result in higher variation in predicted errors. A model overfits if it is given a small dataset. And also to avoid underfitting, we can implement k fold cross validation.

```
str(state.x77)
```

```
## 'data.frame':    50 obs. of  8 variables:
```

```
##  $ Population: num  3615 365 2212 2110 21198 ...
##  $ Income    : num  3624 6315 4530 3378 5114 ...
##  $ Illiteracy: num  2.1 1.5 1.8 1.9 1.1 0.7 1.1 0.9 1.3 2 ...
##  $ Life_Exp  : num  69 69.3 70.5 70.7 71.7 ...
##  $ Murder    : num  15.1 11.3 7.8 10.1 10.3 6.8 3.1 6.2 10.7 13.9 ...
##  $ HS_Grad   : num  41.3 66.7 58.1 39.9 62.6 63.9 56 54.6 52.6 40.6 ...
##  $ Frost     : num  20 152 15 65 20 166 139 103 11 60 ...
##  $ Area      : num  50708 566432 113417 51945 156361 ...
```

```r
ind = createDataPartition(state.x77$Murder, p = 9/10, list = FALSE)
train_state <- state.x77[ind,]
test_state <- state.x77[-ind,]

control_parameters <- trainControl(method = 'cv', number = 10, savePredictions = TRUE)
cv_model <- train(Murder ~ Population + Illiteracy + Life_Exp + Frost + Area, data = state.x77,
               trControl = control_parameters, method = 'lm')
print(cv_model)
```

```
## Linear Regression
##
## 50 samples
##  5 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 43, 45, 46, 46, 45, 45, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   1.857965  0.8065276  1.58509
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
cv_model$finalModel
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Coefficients:
##   (Intercept)     Population     Illiteracy       Life_Exp          Frost
## 120.164031804    0.000177981    1.172980493   -1.607836823   -0.013730312
##          Area
##   0.000006804
```

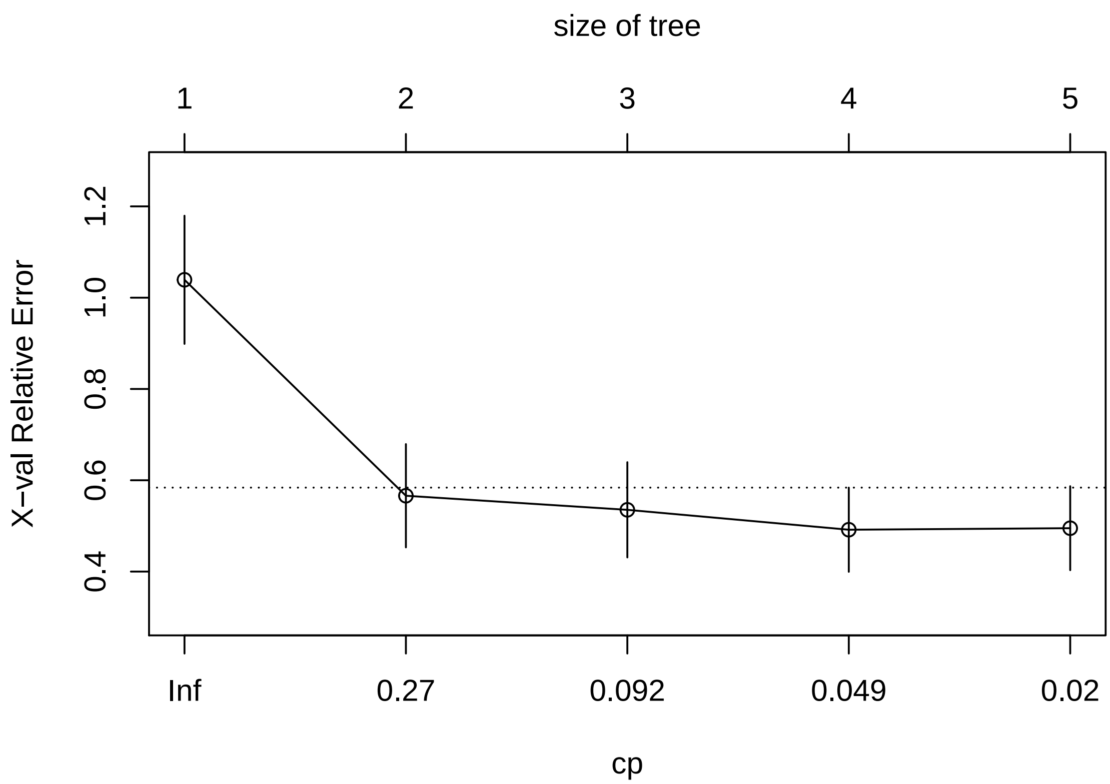
EXTRA CREDIT: Fit a regression tree via CART using the same covariates in your "best" fit model from part (d). Note that CART was not covered in class and you will need to use external resources to learn about/understand it. Use cross validation to select the "best" tree. Compare the models from part (d) and (f) based on their performance. Which do you prefer? Be sure to justify your preference.

After cross validation it was found that the tree was same as the original tree. Hence the regression tree model works well with the data.

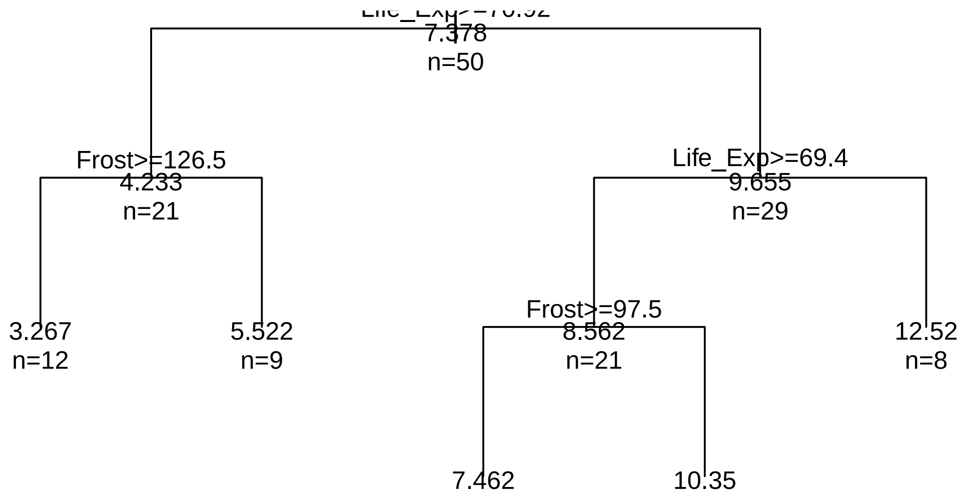
```r
fit2 <- rpart(Murder ~ Life_Exp + Illiteracy + Frost + Population + Area, data = state.x77,
              method = 'anova')
print(fit2)
```

```
## n= 50
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 50 667.74580  7.378000
##    2) Life_Exp>=70.915 21  87.38667  4.233333
##      4) Frost>=126.5 12  27.78667  3.266667 *
##      5) Frost< 126.5 9  33.43556  5.522222 *
##    3) Life_Exp< 70.915 29 222.31170  9.655172
##      6) Life_Exp>=69.395 21 116.90950  8.561905
##       12) Frost>=97.5 13  63.97077  7.461538 *
##       13) Frost< 97.5 8  11.62000 10.350000 *
##      7) Life_Exp< 69.395 8  14.41500 12.525000 *
```

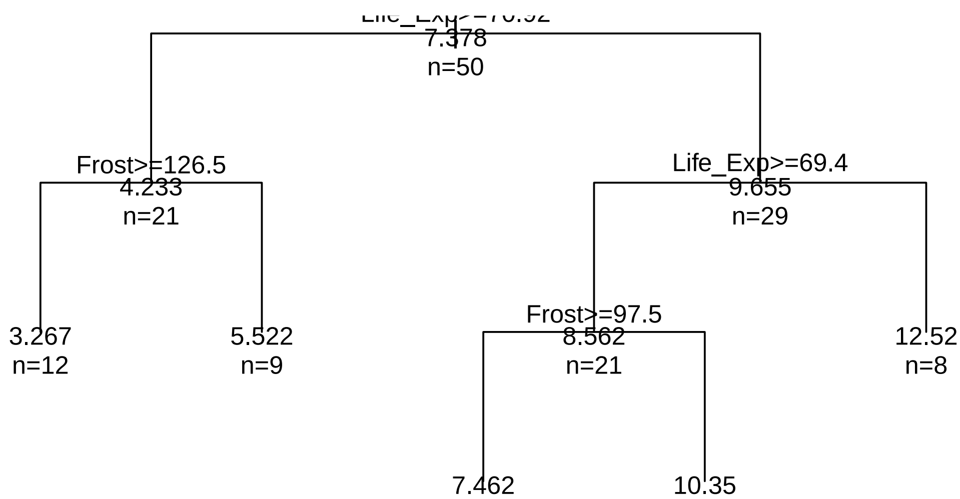
```r
plotcp(fit2)
```




```r
plot(fit2, uniform = TRUE)
text(fit2, use.n = TRUE, all = TRUE, cex=.8)
```




```r
#Pruning the tree

plot(prune(fit2, cp = 0.01160389), uniform = TRUE)
text(prune(fit2, cp = 0.01160389), use.n = TRUE, all=TRUE, cex=.8)
```


7.378
n=50
Frost>=126.5
4.233
n=21
Life_Exp>=69.4
9.655
n=29
3.267
n=12
5.522
n=9
Frost>=97.5
8.562
n=21
12.52
n=8
7.462
10.35

**Problem 3** (5 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.

(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.

Variable informatation - 1. Sample code number: id number 2. Clump Thickness: 1 - 10 3. Uniformity of Cell Size: 1 - 10 4. Uniformity of Cell Shape: 1 - 10 5. Marginal Adhesion: 1 - 10 6. Single Epithelial Cell Size: 1 - 10 7. Bare Nuclei: 1 - 10 8. Bland Chromatin: 1 - 10 9. Normal Nucleoli: 1 - 10 10. Mitoses: 1 - 10 11. Class: (2 for benign, 4 for malignant)

```
#Loading data from the url provided
link <- "http://mlr.cs.umass.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cance
cancer_data <- read.table(link, header = FALSE, sep=",")
```

(b) Tidy the data, ensuring that each variable is properly named and cast as the correct data type. Is there any missing data? Discuss what you see.

All the variables in the dataset have been correctly named. There are 16 N/A values in the Bare Nuclei column. Since there are 699 records and Bare Nuclei is a factor variable, replacing these N/A values with either the mean or imputing them using mice seems inappropriate since they are cell characeteristics. Hence, 16 being a small proportion of 699, we omit those observations.

```
#Adding column names to the data
names(cancer_data)<-c("Sample_code_number","Clump_Thickness","Uniformity_of_Cell_Size",
                      "Uniformity_of_Cell_Shape","Marginal_Adhesion",
                      "Single_Epithelial_Cell_Size","Bare_Nuclei","Bland_Chromatin",
                      "Normal_Nucleoli","Mitoses","Class")

#View(cancer_data)

#Inspecting the data
summary(cancer_data)
```

```
##  Sample_code_number Clump_Thickness  Uniformity_of_Cell_Size
##  Min.   :   61634   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:  870688   1st Qu.: 2.000   1st Qu.: 1.000
##  Median : 1171710   Median : 4.000   Median : 1.000
##  Mean   : 1071704   Mean   : 4.418   Mean   : 3.134
##  3rd Qu.: 1238298   3rd Qu.: 6.000   3rd Qu.: 5.000
##  Max.   :13454352   Max.   :10.000   Max.   :10.000
##
##  Uniformity_of_Cell_Shape Marginal_Adhesion Single_Epithelial_Cell_Size
##  Min.   : 1.000           Min.   : 1.000    Min.   : 1.000
##  1st Qu.: 1.000           1st Qu.: 1.000    1st Qu.: 2.000
##  Median : 1.000           Median : 1.000    Median : 2.000
##  Mean   : 3.207           Mean   : 2.807    Mean   : 3.216
##  3rd Qu.: 5.000           3rd Qu.: 4.000    3rd Qu.: 4.000
##  Max.   :10.000           Max.   :10.000    Max.   :10.000
##
##   Bare_Nuclei  Bland_Chromatin  Normal_Nucleoli     Mitoses
##  1      :402   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  10     :132   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000
```

```
##  2       : 30   Median : 3.000   Median : 1.000   Median : 1.000
##  5       : 30   Mean   : 3.438   Mean   : 2.867   Mean   : 1.589
##  3       : 28   3rd Qu.: 5.000   3rd Qu.: 4.000   3rd Qu.: 1.000
##  8       : 21   Max.   :10.000   Max.   :10.000   Max.   :10.000
##  (Other): 56
##      Class
##  Min.   :2.00
##  1st Qu.:2.00
##  Median :2.00
##  Mean   :2.69
##  3rd Qu.:4.00
##  Max.   :4.00
##
```

```r
str(cancer_data)
```

```
## 'data.frame':    699 obs. of  11 variables:
##  $ Sample_code_number         : int  1000025 1002945 1015425 1016277 1017023 1017122 1018099 101
##  $ Clump_Thickness            : int  5 5 3 6 4 8 1 2 2 4 ...
##  $ Uniformity_of_Cell_Size    : int  1 4 1 8 1 10 1 1 1 2 ...
##  $ Uniformity_of_Cell_Shape   : int  1 4 1 8 1 10 1 2 1 1 ...
##  $ Marginal_Adhesion          : int  1 5 1 1 3 8 1 1 1 1 ...
##  $ Single_Epithelial_Cell_Size: int  2 7 2 3 2 7 2 2 2 2 ...
##  $ Bare_Nuclei                : Factor w/ 11 levels "?","1","10","2",..: 2 3 4 6 2 3 3 2 2 2 ...
##  $ Bland_Chromatin            : int  3 3 3 3 3 9 3 3 1 2 ...
##  $ Normal_Nucleoli            : int  1 2 1 7 1 7 1 1 1 1 ...
##  $ Mitoses                    : int  1 1 1 1 1 1 1 1 5 1 ...
##  $ Class                      : int  2 2 2 2 2 4 2 2 2 2 ...
```

```r
nrow(cancer_data)
```

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

```r
#Checking for NA values
sum(is.na(cancer_data))
```

```
## [1] 0
```

```r
#omitting NA values
cancer_data$Bare_Nuclei <- na.omit(cancer_data$Bare_Nuclei)

#Adding another variable with values 0 and 1 corresponding to benign and malignant cells respective
#This computation is based on the class column from the dataset.
cancer_data$Class1 <- ifelse(cancer_data$Class == '2', 0,  ifelse(cancer_data$Class == '4', 1, NA))
```

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

```r
#Splitting the data into training and testing datasets in the ratio 70:30
set.seed(123)
sample <- sample.int(n = nrow(cancer_data), size = floor(.70*nrow(cancer_data)))
train_cancer_data <- cancer_data[sample, ]
test_cancer_data <- cancer_data[-sample, ]
str(cancer_data)
```

```
## 'data.frame':    699 obs. of  12 variables:
##  $ Sample_code_number         : int  1000025 1002945 1015425 1016277 1017023 1017122 1018099 101
##  $ Clump_Thickness            : int  5 5 3 6 4 8 1 2 2 4 ...
```

```
##  $ Uniformity_of_Cell_Size    : int  1 4 1 8 1 10 1 1 1 2 ...
##  $ Uniformity_of_Cell_Shape   : int  1 4 1 8 1 10 1 2 1 1 ...
##  $ Marginal_Adhesion          : int  1 5 1 1 3 8 1 1 1 1 ...
##  $ Single_Epithelial_Cell_Size: int  2 7 2 3 2 7 2 2 2 2 ...
##  $ Bare_Nuclei                : Factor w/ 11 levels "?","1","10","2",..: 2 3 4 6 2 3 3 2 2 2 ...
##  $ Bland_Chromatin            : int  3 3 3 3 3 9 3 3 1 2 ...
##  $ Normal_Nucleoli            : int  1 2 1 7 1 7 1 1 1 1 ...
##  $ Mitoses                    : int  1 1 1 1 1 1 1 1 5 1 ...
##  $ Class                      : int  2 2 2 2 2 4 2 2 2 2 ...
##  $ Class1                     : num  0 0 0 0 0 1 0 0 0 0 ...
```

(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. Be sure to address which of the errors that are identified you consider most problematic in this context.

```r
#Logistic Regression
logistic_model <- glm(Class1 ~ Clump_Thickness+ Uniformity_of_Cell_Size
                      + Uniformity_of_Cell_Shape + Marginal_Adhesion
                      + Single_Epithelial_Cell_Size+ Bare_Nuclei
                      + Bland_Chromatin  + Normal_Nucleoli + Mitoses,
                      family = "binomial", data = train_cancer_data)
summary(logistic_model)
```

```
##
## Call:
## glm(formula = Class1 ~ Clump_Thickness + Uniformity_of_Cell_Size +
##     Uniformity_of_Cell_Shape + Marginal_Adhesion + Single_Epithelial_Cell_Size +
##     Bare_Nuclei + Bland_Chromatin + Normal_Nucleoli + Mitoses,
##     family = "binomial", data = train_cancer_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.02207  -0.08046  -0.04214   0.00696   2.46428
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -12.89231    2.84819  -4.526 0.000006 ***
## Clump_Thickness               0.53195    0.26687   1.993  0.04623 *
## Uniformity_of_Cell_Size       0.36832    0.37391   0.985  0.32460
## Uniformity_of_Cell_Shape      0.66546    0.47926   1.389  0.16498
## Marginal_Adhesion             0.09943    0.15814   0.629  0.52951
## Single_Epithelial_Cell_Size  -0.06682    0.27716  -0.241  0.80948
## Bare_Nuclei1                  2.85848    1.98858   1.437  0.15059
## Bare_Nuclei10                 6.71841    2.07496   3.238  0.00120 **
## Bare_Nuclei2                  3.40805    2.00693   1.698  0.08948 .
## Bare_Nuclei3                  5.31995    1.92194   2.768  0.00564 **
## Bare_Nuclei4                  7.42234    2.59246   2.863  0.00420 **
## Bare_Nuclei5                  5.45250    2.15155   2.534  0.01127 *
## Bare_Nuclei6                 22.86328 5259.11631   0.004  0.99653
## Bare_Nuclei7                  5.23122    2.25350   2.321  0.02027 *
## Bare_Nuclei8                 21.43200 2205.98803   0.010  0.99225
## Bare_Nuclei9                 20.92615 3346.21511   0.006  0.99501
## Bland_Chromatin               0.41660    0.26688   1.561  0.11853
## Normal_Nucleoli               0.03791    0.16542   0.229  0.81873
## Mitoses                       0.18850    0.32979   0.572  0.56761
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 644.533  on 488  degrees of freedom
## Residual deviance:  55.961  on 470  degrees of freedom
## AIC: 93.961
##
## Number of Fisher Scoring iterations: 18
```

```r
#Predictions
logistic_model_result <- predict(logistic_model, newdata = test_cancer_data, type='response')
nrow(test_cancer_data)
```

```
## [1] 210
```

```r
View(logistic_model_result)

#Confusion Matrix with threshold 0.5
table(test_cancer_data$Class1, logistic_model_result > 0.5)
```

```
##
##     FALSE TRUE
##   0   143    7
##   1     2   58
```

```r
#Accuracy
accuracy <- (144+55)/(144+6+5+55)
accuracy
```

```
## [1] 0.947619
```

```r
#Precision
precision <- 58/(7+58)
precision
```

```
## [1] 0.8923077
```

```r
#Recall
recall <- 58/(58+2)
recall
```

```
## [1] 0.9666667
```

```r
#Sensitivity
sensitivity  <- 58/(58+2)
sensitivity
```

```
## [1] 0.9666667
```

```r
#Specificity
specificity <- 143/(143+7)
specificity
```

```
## [1] 0.9533333
```

```r
#GGPLOT STATING ALL THE METRICS
```

**Problem 4** (10 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.

The 3 real life applications where classification might be useful are as follows -

1. Whether the product will fail or succeed - The response variable would be a factor, with 2 levels - 'success' and 'failure'. The predictor variables that can be considered are as follows - money spent on marketing, category of the product, brand value, average duration spent on R&D.
2. To know whether the cancer is benign or malignant - The response variable would be a factor, with 2 levels - 'malignant' and 'beinign'. The predictor variables in this case could be the characteristics of body cells like uniformity of cell size, cell shape, clump thickness.
3. Classification can be used to decide whether a student would be admitted into a particular university or not. The response variable would be a factor, simply a 'yes' or 'no'. The response variables can be, test scores of a student, average income of family, gpa of the student, how good the statement of purpose is, letter of recommendtions, number of extra-curricular activities the student was involved in.

(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.

The 3 real-life examples in which regression might be useful are as follows -

1. Regression can be used to predict the apartment value. The response variable would be price of the apartment and the response variable can be as follows - average income of family in the neighborhood, crime rate, number of graduate, undergraduate, high school students.
2. Linear Regression can be used to predict the crime rate in a region. The response variable would be crime rate and the predictor variable would be - life expectancy, percentage of diseased patients, number of cases filed, average income, illiteracy rate.
3. Sports analyst use linear regression to predict the number of goals a player would score in the coming matches based on previous performances. The predictors to be considered may look like the following - number of matches played in the last month, opponents played against, number of goals scored, number of chances created, number of attempts, goals per match ratio of the year.

(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?

A flexible model when receives large samples of data performs better than a less flexible model. However, with a small dataset, a flexible model would overfit the data and also increase the variance. Hence, we can also say that a flexible model would perform better with higher degress of freedom.

The advantages of a flexible approach are that it may give a better fit for non-linear models and it decreases the bias. A more flexible approach would be preferred in prediction and not the interpretability of the results predicting the crime rates in a region. A less flexible approach would be preferred in inference and the interpretability of the results, for example, whether the person is a republican or a democrat. (logistic regression useful here).

**Problem 5** (10 pts)

Suppose that large classes at a liberal arts college were divided into sections. The math class (M201) has 5 sections, the chemistry class (C105) has 8 sections, the physics class (P130) has 6 sections, and the history class (H202) has 4 sections. The likelihood of being enrolled in any section for a given class is random and uniformly distributed. Enrollment in a section is not controlled by the students. Selection of a particular class is controlled by the students unless indicated. Each section is referred to by a letter designation (e.g. 'A', 'B', 'C', etc.).

Suppose that Rick and Marty are friends who are enrolling for classes. For Questions a-c and g, it is OK to assume the enrollment of one student in a section will not affect the probability of the enrollment of another in the same section.

(a) What is the probability of Rick and Marty both being enrolled in section A of M201?

P(A) = P(Rick gets enrolled in section A of M201) = 1/5 P(B) = P(Martin gets enrolled in section A of M201) = 1/5 Therefore, P(Rick and Martin both get enrolled in section A of M201) = P(A)*P(B) = 1/25

(b) What is the probability of Rick and Marty both being enrolled in section F of C105?

P(A) = P(Rick gets enrolled in section F of C105) = 1/8 P(B) = P(Martin gets enrolled in section F of C105) = 1/8 P(Rick and Martin both get enrolled in section F of C105) = P(A)*P(B) = 1/64

(c) What is the probability of Rick and Marty being concurrently enrolled in the same M201 and C105 sections?

P(Rick getting enrolled in 1 section of M201) = 1/5 P(Marty getting enrolled in the same section as that of Rick in C105) = 1/8

But, Rick can get enrolled in any of the 5 sections of M201. Therefore P(Rick getting a section in M201) = 5*1/5 = 1 P(Marty getting the same section as Rick) = 1/8

Thus, P(Rick and Marty being concurrently enrolled in the same M201 and C105 sections) = 1/8

(d) What is the probability of Rick being enrolled in section A or section D of M201?

Rick has 2 possible sections to get enrolled in out of the 5 available. Therefore, P(Rick getting enrolled in section A or section D of M201) = 2/5

(e) What is the probability of Marty being enrolled in section B, C, or D of C105?

Marty has 3 available sections to get enrolled in out of the 8 available. Therefore, P(Marty getting enrolled in section B,C, or D of C105) = 3/8

(f) Suppose that each section for every class only has one more seat remaining. Rick and Marty create a random class selector that randomly selects any class across *all* the four classes listed above that have a seat remaining. The random class selector weighs each class based on the number of available sections. What is the probability that Rick uses this random selector first, gets assigned into a M201 section, and then Marty uses the selector and also gets assigned into a M201 section?

P(Rick gets to use the random class selector first) = 1/2 P(Rick gets assigned into a M201 class) = 1/4 Thus, P(Rick uses the random selector first and gets assigned into a M201 section) = 1/2 * 1/4 = 1/8

P(Marty gets assigned to M201 class) = 1/4 P(Marty gets other sections except for the one occupied by Rick) = 4/5 Therefore, P(Marty gets into a M201 section) = 1/4 * 4/5 = 1/5

P(Rick and Marty get into a M201 section) = 1/8 * 1/5 = 1/40

(g) Now suppose that each section for every class has multiple seats remaining. What is the probability of both Rick and Marty each using the random class selector once and being assigned to the same class, regardless of which class it is and which section they're in?

P(Rick gets into any one of the class) = 4 * 1/4 P(Marty gets into the same class as Rick) = 1/4 Therefore, P(Rick and Marty get assigned to the same class) = 1/4

Bruce Wayne goes to his trusted mechanic with car issues. Upon inspecting the vehicle, the mechanic, Alfred, determines the issue is either with the transmission, with the spark plugs, or with both. Alfred determines there is a probability of 0.8 that the issue is with the transmission and there is a probability of 0.3 that there is an issue with the spark plugs.

(h) What is the probability that there is an issue with both? Assume there is zero chance that the car has no issue; assume there is zero chance the car has any other issue. Show your work.

P(issue with Transmission)=0.8 P(No issue with Transmission)=1-0.8=0.2

P(issue with plugs)=0.3 P(No issue with plugs)=1-0.3=0.7

P(issue with both)=1-(0.8x0.7)-(0.2x0.3)=0.38

**Problem 6 - Extra Credit** (≤ 3 pts)

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.
Here, in-built Boston dataset has been used. This dataset has been split into training and testing dataset in the ratio 80:20. Hence, as we can see the Boston dataset has 506 observations and the training and testing dataset have 404 and 102 observations respetively. We calculate RMSE (root mean squared error) for each of these models and use it for comparison between the different models.

```r
#Using the Boston dataset

boston <- Boston
str(boston)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```r
summary(boston)
```

```
##       crim                zn             indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000  
##       nox               rm             age              dis        
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67  
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23  
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90  
##      lstat            medv      
##  Min.   : 1.73   Min.   : 5.00  
```

```
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

```r
#Splitting the state dataset into train and test in the ration 80/20
set.seed(101)
sample1 <- sample.int(n = nrow(boston), size = floor(.80*nrow(boston)))
train_boston <- boston[sample1, ]
test_boston <- boston[-sample1, ]
nrow(boston)
```

```
## [1] 506
```

```r
nrow(train_boston)
```

```
## [1] 404
```

```r
nrow(test_boston)
```

```
## [1] 102
```

```r
#Bagging - mtry is set to the number of predictor variables and hence randomForest is used as a case of

bagging_boston = randomForest(medv ~ ., data = train_boston, mtry = 13,
                           importance = TRUE, ntrees = 500)
bagging_boston
```

```
##
## Call:
##  randomForest(formula = medv ~ ., data = train_boston, mtry = 13,      importance = TRUE, ntrees = 5
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 13
##
##           Mean of squared residuals: 9.646282
##                     % Var explained: 88.79
```

```r
#Predictions - Bagging
boston_predict_bagging = predict(bagging_boston, newdata = test_boston)
summary(boston_predict_bagging)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   7.966  18.001  20.771  22.769  25.199  46.689
```

```r
boston_predict_bagging
```

```
##        19        24        25        29        32        41        48        51
## 18.165603 15.321923 16.922863 19.946017 19.413697 33.600660 20.083177 20.977720
##        60        61        63        66        67        69        70        71
## 20.729663 19.117053 24.061127 26.158007 20.515237 19.274853 21.111517 23.933587
##        86        91        96        98        99       112       113       115
## 25.911807 22.671120 25.647460 45.634770 45.528910 23.046560 19.601503 21.351117
##       116       118       121       123       126       133       134       138
## 18.842510 20.673507 21.727877 18.373100 19.046163 20.149473 17.544707 18.295537
##       142       148       166       168       174       179       181       183
## 13.178613 13.569543 20.908710 19.306370 23.126733 27.766537 42.391360 35.559073
##       193       212       218       221       223       224       228       229
```

```
## 34.627340 20.121900 22.993503 27.513643 25.155977 24.603823 31.181257 44.003947
##       235       251       259       264       271       273       277       278
## 25.204230 24.554777 35.762717 30.220310 21.158210 24.733197 34.864360 31.978100
##       279       281       283       284       287       297       308       309
## 24.728110 46.688900 44.513513 45.840473 22.052697 24.360633 29.090940 29.588350
##       334       335       336       348       358       365       368       382
## 25.182207 24.789727 20.236907 24.087387 20.911970 45.009833 17.946470 11.335260
##       385       388       395       396       404       408       413       418
##  9.127993  8.698997 12.653427 14.600153 11.972513 29.185153 14.233790  7.965897
##       423       424       426       427       439       448       449       452
## 18.239887 12.598290  9.405247 14.710847  8.723747 16.204797 15.952777 15.918597
##       456       457       461       462       466       472       476       477
## 15.407990 16.166203 15.917680 19.394023 19.751077 20.813310 15.282513 16.414493
##       487       488       494       496       498       504
## 19.297947 21.565810 20.267620 19.555440 20.063387 28.165993
```

```r
bagging_rmse <- sqrt(mean(test_boston$medv - boston_predict_bagging)^2)
bagging_rmse
```

```
## [1] 0.3311963
```

```r
forest_boston = randomForest(medv ~ ., data = train_boston, mtry = 4,
                             importance = TRUE, ntrees = 500)

forest_boston
```

```
##
## Call:
##  randomForest(formula = medv ~ ., data = train_boston, mtry = 4,      importance = TRUE, ntrees = 50
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 11.68105
##                     % Var explained: 86.43
```

```r
#Predictions
boston_predict_rf = predict(forest_boston, newdata = test_boston)

#RMSE - RF
rf_rmse <- sqrt(mean(test_boston$medv - boston_predict_rf)^2)
rf_rmse
```

```
## [1] 0.1916558
```

```r
#Boosting

boost_boston = gbm(medv ~ ., data = train_boston, distribution = "gaussian",
                   n.trees = 5000, interaction.depth = 4, shrinkage = 0.01)
boost_boston
```

```
## gbm(formula = medv ~ ., distribution = "gaussian", data = train_boston,
##     n.trees = 5000, interaction.depth = 4, shrinkage = 0.01)
## A gradient boosted model with gaussian loss function.
## 5000 iterations were performed.
## There were 13 predictors of which 13 had non-zero influence.
```

```
boston_predict_boost = predict(boost_boston, newdata = test_boston, n.trees = 5000)

boost_rmse <- sqrt(mean(test_boston$medv - boston_predict_boost)^2)
boost_rmse
```

```
## [1] 0.1821272
```

(a) How are the results compared to simple methods like linear or logistic regression?

Here RMSE values are considered as benchmarks to measure the performance of the respective regression models. The RMSE value of the linear regression model is approximately 0.52 which is higher than other methods.

```
#Linear Model

boston_lm <- lm(medv ~ ., data = train_boston)

#Prediction using linear model

boston_predict_lm <- predict(boston_lm, newdata = test_boston, type = 'response')

head(boston_predict_lm)
```

```
##       19       24       25       29       32       41
## 15.15858 13.98882 16.02703 20.31552 18.57521 34.32306
```

```
#Calculate RMSE
lm_rmse <- sqrt(mean( test_boston$medv - boston_predict_lm)^2)
lm_rmse
```

```
## [1] 0.5281051
```

(b) Which of the approaches yields the best performance?

Random Forest Model has the lowest RMSE value and hence can be considered to yield the best performance.

```
(rmse = data.frame(
  Model = c ("Linear Model", "Bagging",  "Random Forest",  "Boosting"),
  rmse_values = c(lm_rmse, bagging_rmse, rf_rmse, boost_rmse)
  )
)
```

```
##           Model rmse_values
## 1  Linear Model   0.5281051
## 2       Bagging   0.3311963
## 3 Random Forest   0.1916558
## 4      Boosting   0.1821272
```

```
rmse
```


```
##           Model rmse_values
## 1  Linear Model   0.5281051
## 2       Bagging   0.3311963
## 3 Random Forest   0.1916558
## 4      Boosting   0.1821272
```