IMT 573 Final Exam

Jenny Skytta

Due: June 5, 2022

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. Open the `final_exam.Rmd` file on RStudio Cloud. 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 Prof. Ernest Green 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 PDF file 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 (Dr. Emma Spiro). 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 or from the improper citation of work that is not my own.

(Jenny Skytta)

(May 31, 2022)

Setup

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

```
install.packages("ggplot2")
install.packages("caret")
library(tidyverse) # load package
library(AER) # loading AER package
library(bestglm) # loading package for glm
library(car) # loading for scatterplotmatrix
library(MASS) #loading for stepAIC
library(ggplot2) #caret dependent on ggplot2
library(caret) #loading for k-fold validation
library(ISLR)
library(XML)
library(ipred)
library(randomForest)
library(rpart)
library(rpart.plot)
library(rsample)
library(boot)
```

Problem 1 (15 pts)

```
data(Affairs) # adding Affairs to global environment
```

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

```
colnames(Affairs)
```

```
## [1] "affairs"       "gender"        "age"           "yearsmarried"
## [5] "children"      "religiousness" "education"     "occupation"
## [9] "rating"
```

```
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
```

(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.

```
kids_yes <- round((sum(Affairs$children == "yes") / 601) * 100)
kids_no <- round((sum(Affairs$children == "no") / 601) * 100)

female <- round((sum(Affairs$gender == "female") / 601) * 100)
male <- round((sum(Affairs$gender == "male") / 601) * 100)

college <- round((sum(Affairs$education > 14) / 601) * 100)
```

The Affairs data is comprised of 601 participants from a study examining extramarital affairs. Within these data, there are 315 female participants representing 52% and 286 male participants representing 48%. The average age of the participants is ~ 32 with the youngest being 17 and oldest being 57. The majority of the participants in the study had children 72% while only 28% did not have any children. The dataset was largely college educated with 66% having completed at least 14 years of school.

The main ethical concerns I have with these data are that it wouldn't necessarily generalize to the

overall population. The educational background of the participants doesn't reflect the overall mean education of the US population at large in 1969. With these educational numbers, I would also wonder about the demographics of the participants themselves understanding historical barriers in the 1960s around education of non-white individuals. Put together, one could infer the dataset is largely white, college educated, likely upper socioeconomoically advantaged individuals.

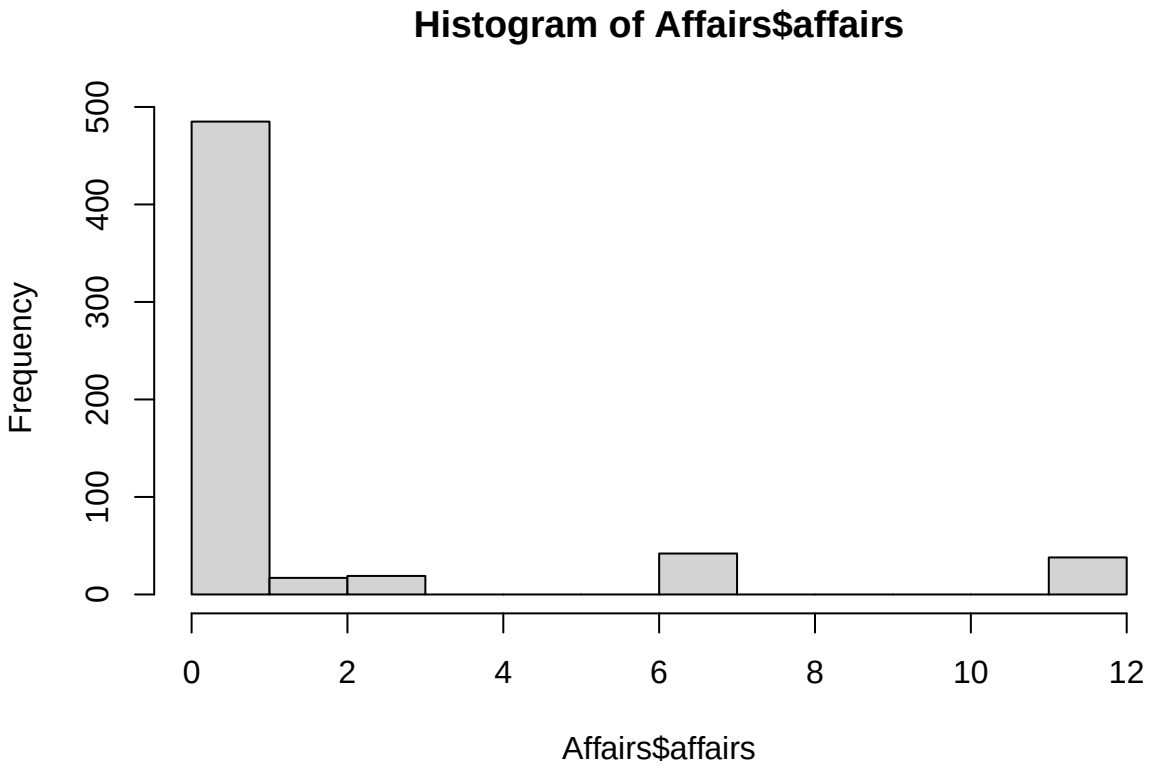
Regarding privacy, if one were to know where these data were collected, some of the parameters could likely lead to identification due to the relatively small sample and outlier educational parameters.

(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?

```
affair_binary <- Affairs %>%
  mutate("unf" = (affairs >= 1)) %>%
  mutate("fai" = (affairs == 0)) %>%
  summarise(sum(unf), sum(fai))

yes_affairs <- round((affair_binary$`sum(unf)` / 601) * 100)

hist(Affairs$affairs)
```




```
table(affair_binary)
```

```
##         sum(fai)
## sum(unf) 451
##      150   1
```

One advantage of looking at whether one has had affairs versus the number of affairs is that you can get a better overall picture of the data's spread in terms of infidelity. In the summary, using the number of instances of sexual infidelity, the data means are skewed to almost suggest the average is approximately

one across the group. In looking at this from a binary perspective, we see that, in fact, about 25% of the participants actually engaged in extramarital affairs.

(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.

```
Affairs <- Affairs %>%
  mutate("unf" = (affairs >= 1)) %>%  #adding binomial yes/no col affairs info
  mutate("fai" = (affairs == 0))

# creating a multivariate model
m1_affairs <- lm(unf ~ gender + age + yearsmarried + children + religiousness + education + occupat
summary(m1_affairs) # summarizing the multivariate model
```

```
##
## Call:
## lm(formula = unf ~ gender + age + yearsmarried + children + religiousness +
##     education + occupation + rating, data = Affairs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6336 -0.2691 -0.1632  0.1151  1.0659
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.736107   0.151502   4.859 1.51e-06 ***
## gendermale     0.045201   0.040022   1.129 0.259180
## age           -0.007420   0.003013  -2.463 0.014057 *
## yearsmarried   0.015981   0.005491   2.911 0.003743 **
## childrenyes    0.054487   0.046642   1.168 0.243198
## religiousness -0.053698   0.014881  -3.608 0.000334 ***
## education      0.003078   0.008542   0.360 0.718699
## occupation     0.005913   0.011838   0.499 0.617643
## rating        -0.087455   0.015984  -5.472 6.59e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4122 on 592 degrees of freedom
## Multiple R-squared:  0.1066, Adjusted R-squared:  0.09452
## F-statistic: 8.829 on 8 and 592 DF,  p-value: 1.884e-11
```

From the multivariate linear model, it looks like happiness rating and religiousness have a statistically significant correlation on outcomes of infidelity. There is a slightly significant influence with the number of years married as well.

(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.

```
db.affairs.glm <- Affairs[2:10]

glm.affairs <- bestglm(Xy = db.affairs.glm, family = gaussian, IC = "AIC", method = "exhaustive")
```

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

```
summary(glm.affairs$BestModel)
```

```
##
## Call:
## lm(formula = y ~ ., data = data.frame(Xy[, c(bestset[-1], FALSE),
##     drop = FALSE], y = y))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.6304 -0.2663 -0.1586  0.1077  1.0250
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.821582   0.103300   7.953 9.18e-15 ***
## gendermale     0.063607   0.034902   1.822 0.068892 .
## age           -0.007397   0.002988  -2.475 0.013586 *
## yearsmarried   0.018596   0.004970   3.741 0.000201 ***
## religiousness -0.054425   0.014815  -3.674 0.000261 ***
## rating        -0.087599   0.015764  -5.557 4.15e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4117 on 595 degrees of freedom
## Multiple R-squared:  0.1039, Adjusted R-squared:  0.09639
## F-statistic:  13.8 on 5 and 595 DF,  p-value: 9.04e-13
```

Using the bestglm function, the best model covariates are the same as from the previous linear modeling completed in part c which are yearsmarried, religiousness, and happiness rathing. The one difference from this approach is that yearsmarried appears to be more significant.

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

A one-unit increase in yearsmarried is associated with an expected increase of 0.018596 units of Y. y = 0.018596(x) + 0.821582

A one-unit increase in regliousness is associated with an expected decrease of -0.054425 units of Y. y = -0.054425(x) + 0.821582

A one-unit increase in rating is associated with an expected decrease of -0.087599 units of Y. y = -0.087599(x) + 0.821582

(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.

```
#probability average for affairs from Affairs df
affairs_prob <- mean(predict(m1_affairs, Affairs))
# create random set using mean and sd from original data
affairs <- round(abs(rbinom(n = 600, size = 1, prob = 0.25)))
sdage <- sd(Affairs$age)  #stanard deviation of age
# create random set using mean and sd from original data
age <- rnorm(n = 600, mean = 32.49, sd = 9.28)
# create random binomial vector of 0 and 1s with 50% probability
gender <- rbinom(n = 600, size = 1, prob = 0.5) # 0 female and 1 male
sdmarried <- sd(Affairs$yearsmarried)  #stanard deviation of yearsmarried
# create random set using mean and sd from original data
```

```
yearsmarried <- round(abs(rnorm(n = 600, mean = 8.17, sd = 5.57)))
#create random binomial vector of 0 and 1s with 50% probability
children <- rbinom(n = 600, size = 1, prob = 0.5) # 0 no and 1 yes
sdrelig <- sd(Affairs$religiousness) #stanard deviation of regliousness
# create random set using mean and sd from original data
religiousness <- round(rnorm(n = 600, mean = 3.11, sd = 1.16))
sdedu <- sd(Affairs$education)  #stanard deviation of education
# create random set using mean and sd from original data
education <- round(abs(rnorm(n = 600, mean = 16.17, sd = 2.40)))
socu <- sd(Affairs$occupation)  #stanard deviation of occupation
# create random set using mean and sd from original data
occupation <- round(abs(rnorm(n = 600, mean = 4.19, sd = 1.18)))
set.seed(456)
# creating random uniform vector of numbers 1 through 5
rating <- round(runif(600, min=1, max=5))

TestAffairs <- as.tibble(c(affairs[1:600]))
```

```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
```

```
names(TestAffairs)[names(TestAffairs) == 'value'] <- 'affairs' #rename value to affairs
TestAffairs$age <- age[1:600] #add variable from vector
TestAffairs$gender <- gender[1:600] #add variable from vector
TestAffairs$yearsmarried <- yearsmarried[1:600] #add variable from vector
TestAffairs$children <- children[1:600] #add variable from vector
TestAffairs$religiousness <- religiousness[1:600] #add variable from vector
TestAffairs$education <- education[1:600] #add variable from vector
TestAffairs$occupation <- occupation[1:600] #add variable from vector
TestAffairs$rating <- rating[1:600] #add variable from vector

# linear model of training training data
m1_TestAffairs <- lm(affairs ~ age + gender + yearsmarried + children + religiousness + education +

test_affairs_predict <- predict(m1_TestAffairs, TestAffairs, interval = "confidence")

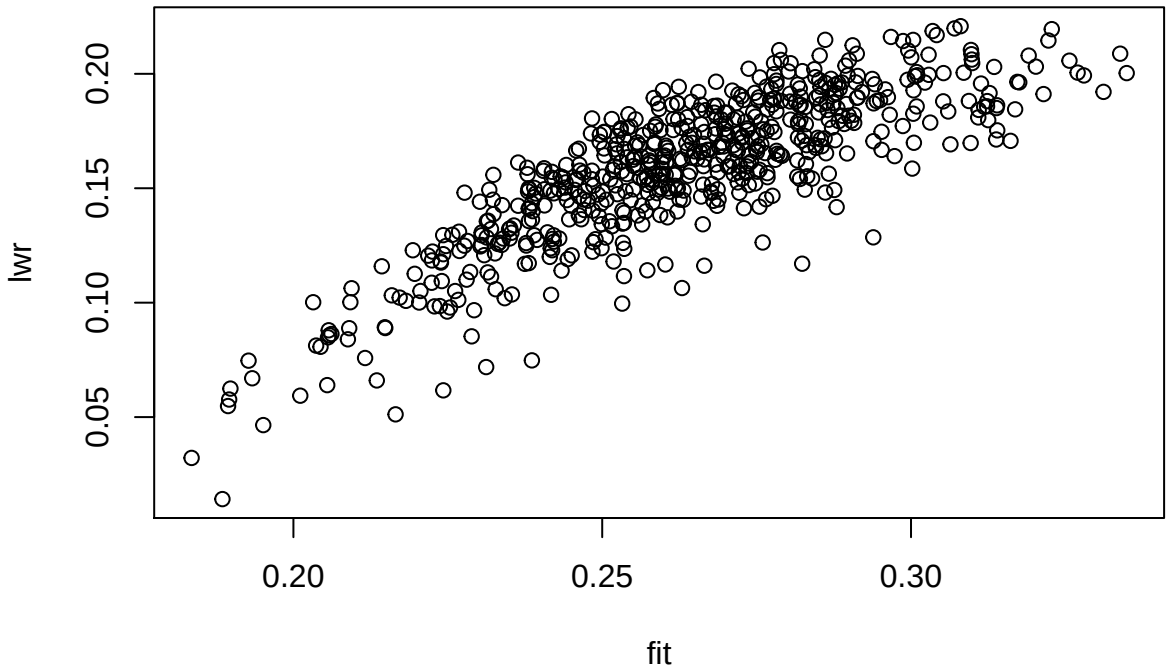
summary(test_affairs_predict)
```

```
##       fit              lwr               upr
##  Min.   :0.1835   Min.   :0.01417   Min.   :0.3062
##  1st Qu.:0.2461   1st Qu.:0.14223   1st Qu.:0.3434
##  Median :0.2634   Median :0.16253   Median :0.3656
##  Mean   :0.2633   Mean   :0.15838   Mean   :0.3683
##  3rd Qu.:0.2817   3rd Qu.:0.18217   3rd Qu.:0.3873
##  Max.   :0.3349   Max.   :0.22080   Max.   :0.4702
```

```
table(TestAffairs[1, ])
```

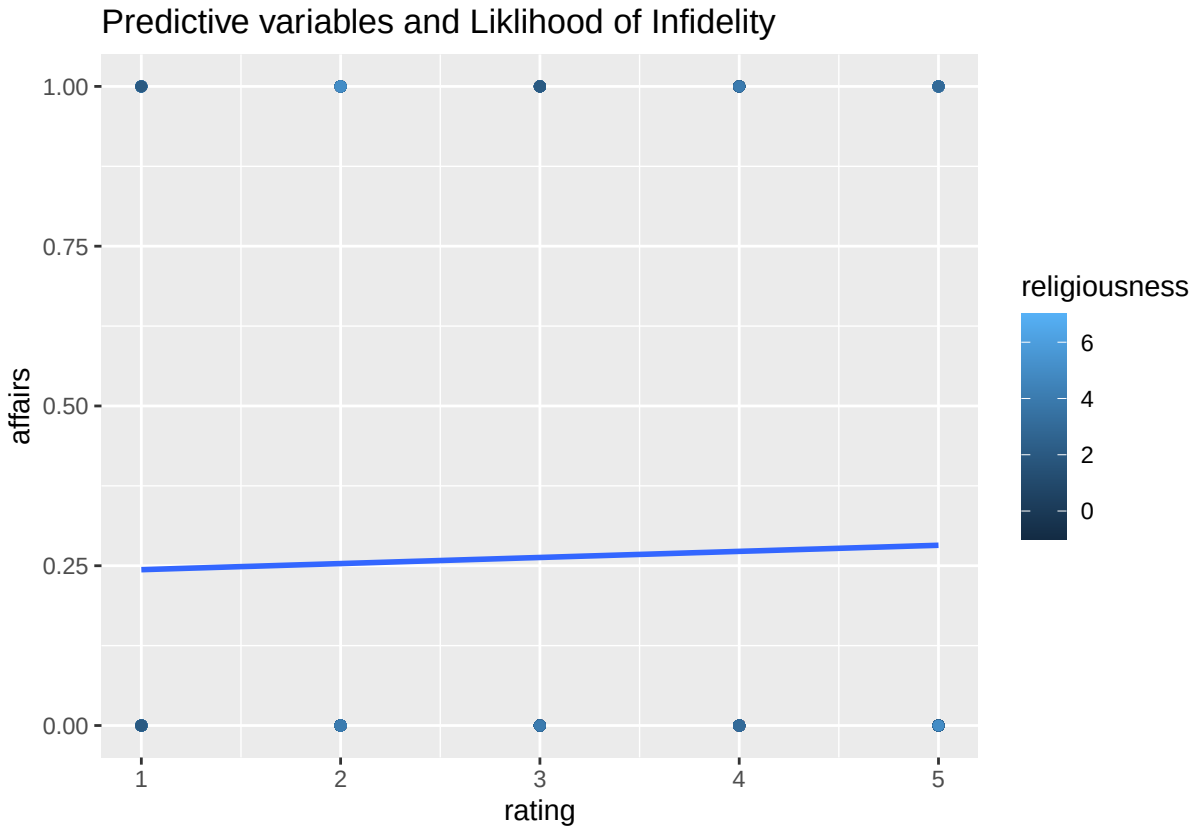
```
## , , gender = 1, yearsmarried = 11, children = 1, religiousness = 3, education = 14, occupation =
##
##        age
## affairs 30.5553354414436
##       0                1
```

```
plot(test_affairs_predict)
```




```
ggplot(TestAffairs, aes(x = rating, y = affairs, color = religiousness) ) +
     geom_point() +
     geom_smooth(method = "lm", se = FALSE) +
     ggtitle("Predictive variables and Liklihood of Infidelity")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



The prediction shows that for instance, a female subject with no children, lower religiousness, 16 years of college, 3 years employed, and a happiness rating of 1, with 95% confidence, there is a 0.02% to 0.03% chance that she would have an affair.

(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 data reflects heteronormative inputs and includes socioeconomic outliers. I do also question the validity of self-disclosure related to infidelity. Overall, the data do not feel sufficient to make valid predictions to a generalized population at large.

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.

```
data("state") # load data

# convert list to tibble
x77 <- as.tibble(state.x77)

# covert from vectors to dataframe
Statedf <- as.tibble(c(state.abb))
# add column for join

Statedf$abb <- state.abb
Statedf$area <- state.area
Statedf$division <- state.division
Statedf$name <- state.name
Statedf$region <- state.region
Statedf$x <- state.center$x
Statedf$y <- state.center$y
# join the data by abb
names(x77)[names(x77) == 'HS Grad'] <- 'HS_grad' #rename value

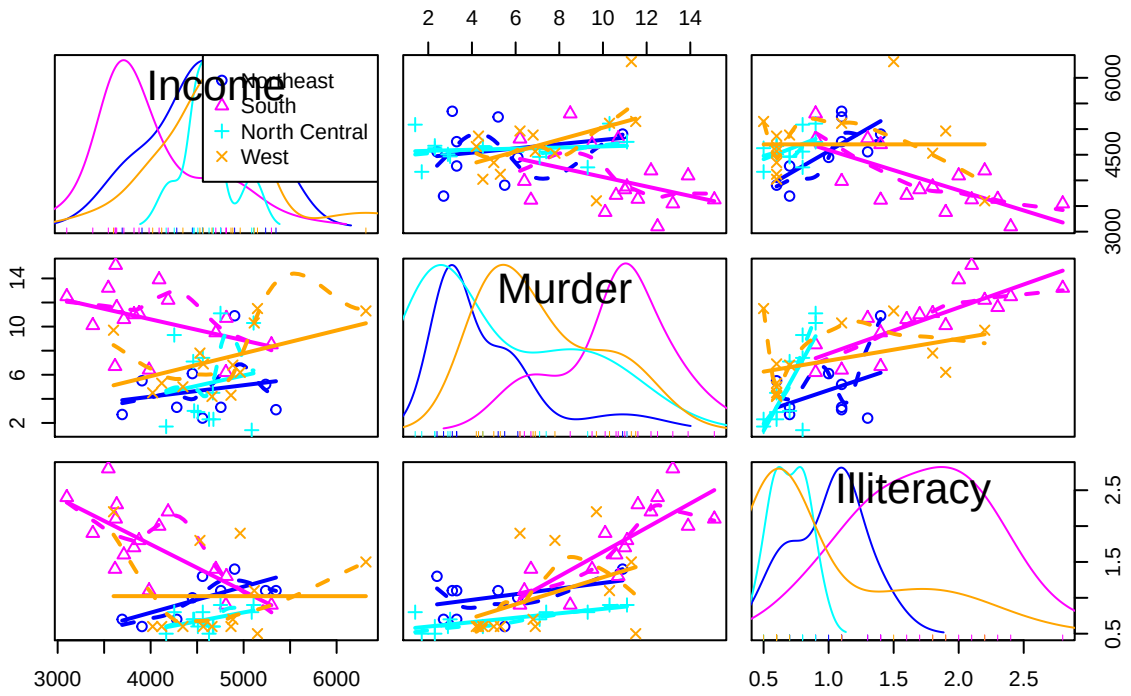
State_data <- cbind(Statedf, x77)
```

(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.

```
# viewing column names to contemplate variables of interest
colnames((State_data))
```

```
##  [1] "value"      "abb"        "area"       "division"   "name"
##  [6] "region"     "x"          "y"          "Population" "Income"
## [11] "Illiteracy" "Life Exp"   "Murder"     "HS_grad"    "Frost"
## [16] "Area"
```

```
# create scatterplot matrix of income, murder and illiteracy by region
scatterplotMatrix(~Income+Murder+Illiteracy|region, data=State_data ,
      main="Scatter plot with Three Cylinder Options"
      )
```



From the matrix, it appears that income is normally distributed in the northeast, west while north central has some variation and southern region income is right skewed. Murder is skewed in all regions with northweast, west, north central all right skewed and southern region left skewed. Illiteracy appears to be right skewed for all regions except southern region which almost appears to be a normal distribution. Overall these data bivariate relationships suggest that the southern region has divergent data when compared to the rest of the country.

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

```
m1_states <- lm(Murder ~ Income + Illiteracy + HS_grad + Population, data = State_data)
summary(m1_states)
```

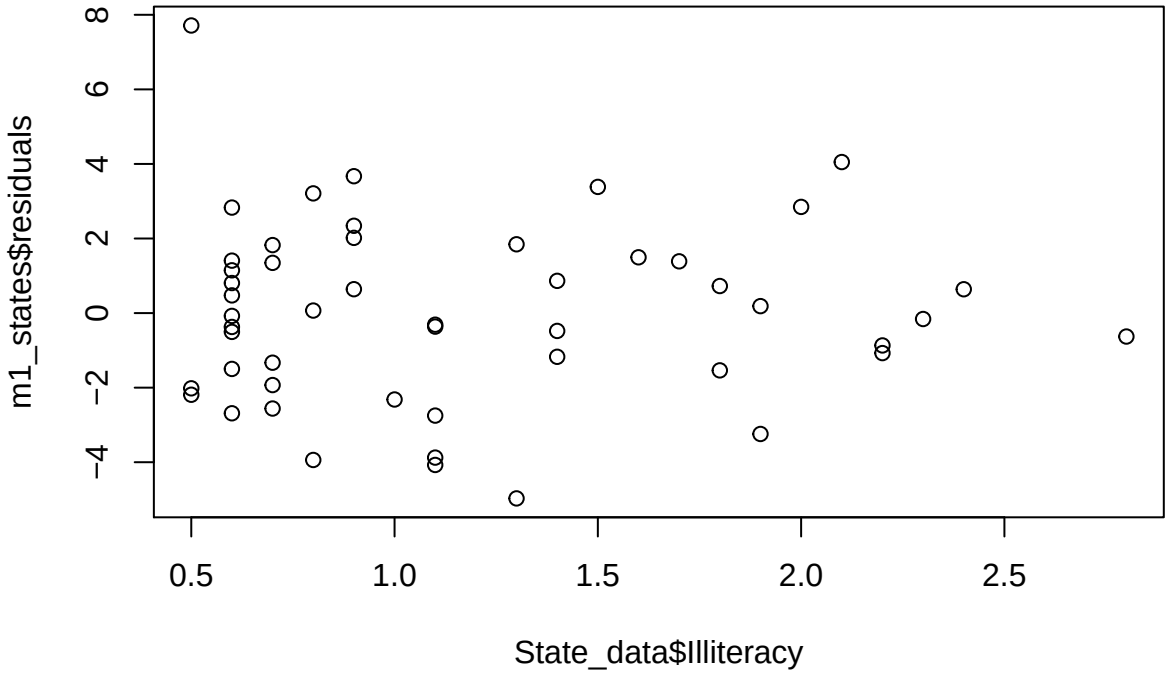
```
##
## Call:
## lm(formula = Murder ~ Income + Illiteracy + HS_grad + Population,
##     data = State_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9700 -1.5271 -0.1173  1.4018  7.7133
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.1840829  4.1636778   0.525   0.6025
## Income       0.0002118  0.0008013   0.264   0.7927
## Illiteracy   3.9674321  0.7906826   5.018 8.67e-06 ***
## HS_grad     -0.0245240  0.0697684  -0.352   0.7268
## Population   0.0002154  0.0000870   2.476   0.0171 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.532 on 45 degrees of freedom
## Multiple R-squared:  0.5681, Adjusted R-squared:  0.5297
## F-statistic:  14.8 on 4 and 45 DF,  p-value: 8.617e-08
```

```
By reviewing the adjust R-squared, it looks like 52% of the variance found in the response variable

\item[(c)] Evaluate the statistical assumptions in your regression analysis from part (b) by perfor
```

```
```r
plot(m1_states$residuals ~ State_data$Illiteracy)
```



Illiteracy does hold statistical significance but we have a low R-squared. Predictions of human behavior will tend to have R-squared values less than 50% so this may not be too important. The p value suggests that our results are not random and that a correlation does exist between the variables.

(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?

```
stepwise_m1 <- stepAIC(m1_states, trace = TRUE) # creating a stepwise model
```

```
## Start:  AIC=97.62
## Murder ~ Income + Illiteracy + HS_grad + Population
##
##              Df Sum of Sq    RSS     AIC
## - Income      1     0.448 288.84  95.693
## - HS_grad     1     0.792 289.19  95.753
## <none>                    288.40  97.616
## - Population  1    39.291 327.69 102.002
## - Illiteracy  1   161.359 449.76 117.834
##
## Step:  AIC=95.69
## Murder ~ Illiteracy + HS_grad + Population
```
```

```
##
##              Df Sum of Sq    RSS     AIC
## - HS_grad     1     0.401 289.25  93.763
## <none>                    288.84  95.693
## - Population  1    48.125 336.97 101.398
## - Illiteracy  1   160.989 449.83 115.843
##
## Step:  AIC=93.76
## Murder ~ Illiteracy + Population
##
##              Df Sum of Sq    RSS     AIC
## <none>                    289.25  93.763
## - Population  1    48.517 337.76  99.516
## - Illiteracy  1   299.646 588.89 127.311
```

```
summary(stepwise_m1) # summary of AIC
```

```
##
## Call:
## lm(formula = Murder ~ Illiteracy + Population, data = State_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7652 -1.6561 -0.0898  1.4570  7.6758
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.652e+00  8.101e-01   2.039  0.04713 *
## Illiteracy  4.081e+00  5.848e-01   6.978 8.83e-09 ***
## Population  2.242e-04  7.984e-05   2.808  0.00724 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.481 on 47 degrees of freedom
## Multiple R-squared:  0.5668, Adjusted R-squared:  0.5484
## F-statistic: 30.75 on 2 and 47 DF,  p-value: 2.893e-09
```

In the stepwise AIC, the ideal model appears to be using Illiteracy and population as the predictive variables of interest. This is identical to the outcomes from the model in part b. The prior adjust R-squared was ~52% with the stepwise model denoting ~54%. Overall, this does suggest that there is a linear relationship between illiteracy and murder in the united states.

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

```
# random seed
set.seed(1978)

# create index matrix with df and outcome variable with 80% of cases
states_split <- createDataPartition(State_data$Murder, p=.8, list = FALSE, times = 1)

# creating training dataframe referencing by rows
train_k_states <- State_data[states_split, ]
# create testing data with all other data
test_k_states <- State_data[-states_split, ]
```

```r
# K fold validation cv = cross validation, 10 folds,
# save predictions and save probabilities

states.cv <- trainControl(method = "cv", number = 10, savePredictions = "all", classProbs = TRUE)

# setting seed
set.seed(1978)

# logistic regression model - generalized linear model
m2_cv_states <- train(Murder ~ Income + Illiteracy + HS_grad + Population,
                      data = train_k_states,
                      method = "glm",
                      trControl = states.cv) # model based on our k fold cv
```

```
## Warning in train.default(x, y, weights = w, ...): cannnot compute class
## probabilities for regression
```

```r
summary(m2_cv_states)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.7804  -1.6878   0.0786   1.3581   7.3421
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.596e-01  4.476e+00   0.147   0.8836
## Income      1.519e-04  8.767e-04   0.173   0.8634
## Illiteracy  4.017e+00  8.800e-01   4.564 5.37e-05 ***
## HS_grad     9.031e-03  8.011e-02   0.113   0.9109
## Population  2.017e-04  8.949e-05   2.254   0.0302 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 6.598381)
##
##     Null deviance: 543.82  on 41  degrees of freedom
## Residual deviance: 244.14  on 37  degrees of freedom
## AIC: 205.11
##
## Number of Fisher Scoring iterations: 2
```

```r
varImp(m2_cv_states)
```

```
## glm variable importance
##
##             Overall
## Illiteracy  100.00
## Population   48.11
## Income        1.36
## HS_grad       0.00
```

```r
# Applying model to test data to predict outcome from train data
m2.predictions <- predict(m2_cv_states, newdata = test_k_states)

# create confusion matrix from predictions
RMSE <- RMSE(m2.predictions, test_k_states$Murder)
```

From the generalized linear model with 10 fold of the training set, we see that HS graduation has 0% importance and Illiteracy holds the most importance. Population appears to hold moderate influence in prediction of murder. The RMSE in the 10 fold cross validation test (2.4304424) is similar to the residual starard error in the previous stepwise model: 2.481.

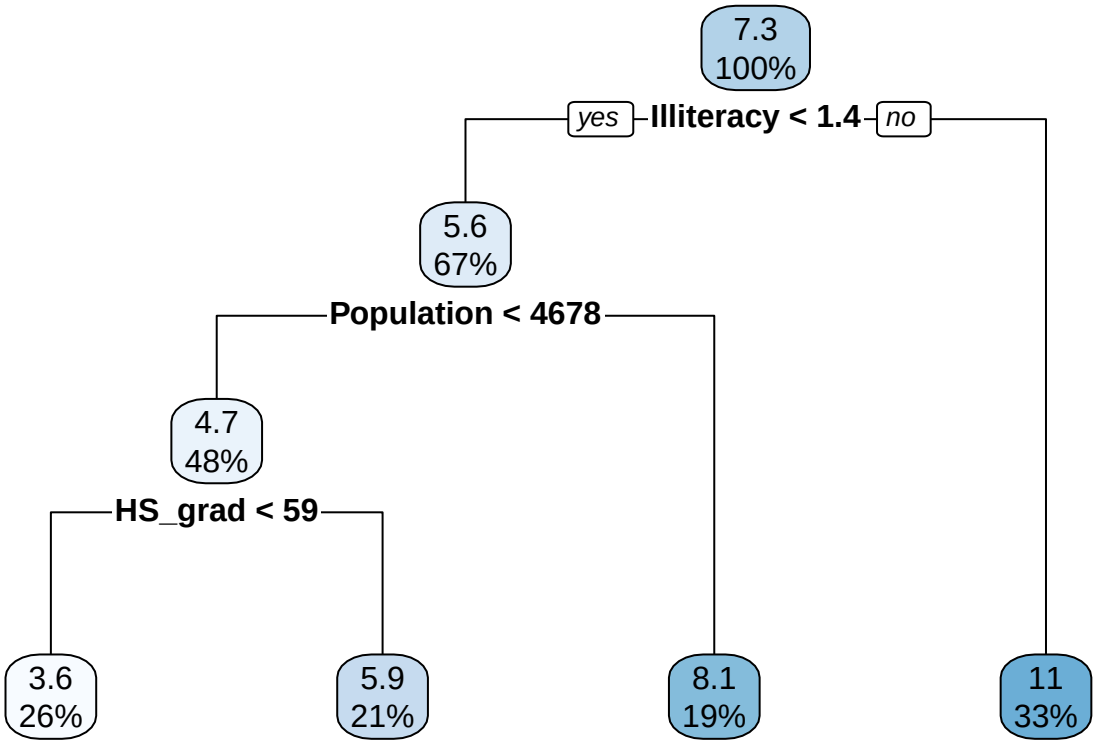
EXTRA CREDIT: Fit a regression tree using the same covariates in your "best" fit model from part (d). Use cross validation to select the "best" tree. Compare the models from part (d) and (f) based on their performance. Which do you prefer? Be sure to justify your preference.

```r
# We can fit a regression tree using 'rpart' and then visualize it using 'rpart.plot'

reg_tree_m2_cv_states <- rpart(Murder ~ Income + Illiteracy + HS_grad + Population, data = train_k_
summary(reg_tree_m2_cv_states)
```

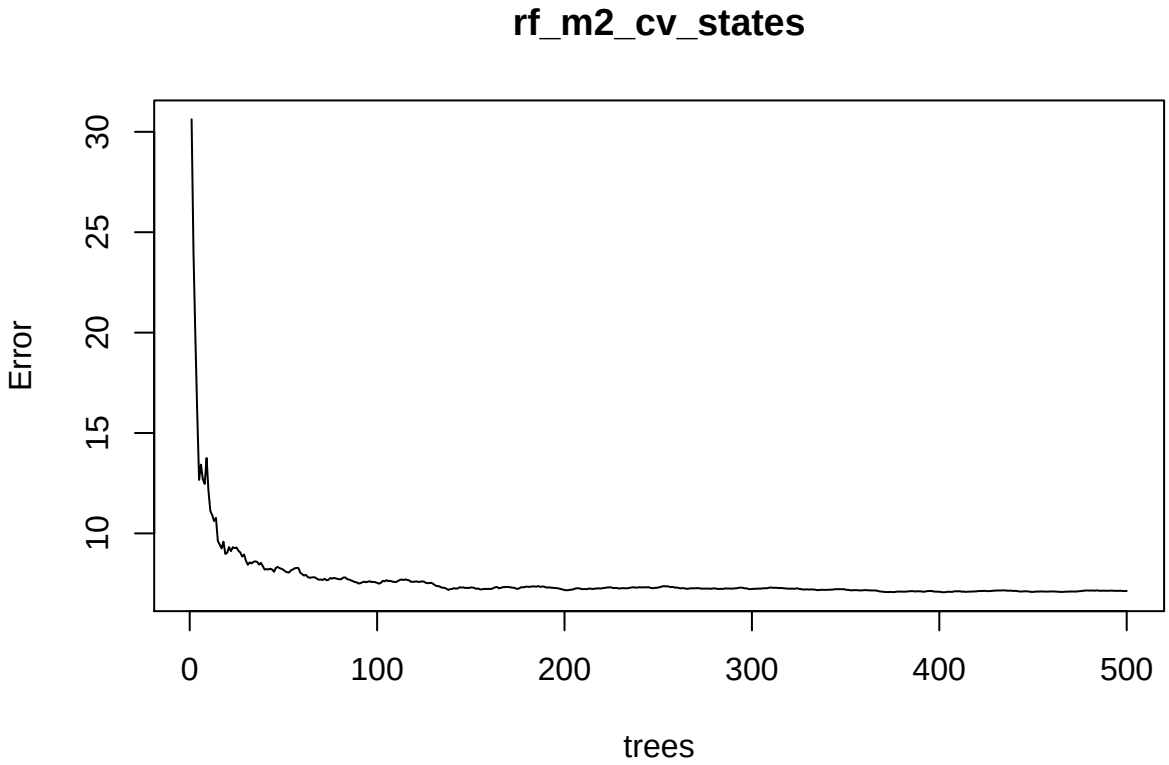
```
## Call:
## rpart(formula = Murder ~ Income + Illiteracy + HS_grad + Population,
##     data = train_k_states, method = "anova")
##   n= 42
##
##           CP nsplit rel error    xerror      xstd
## 1 0.43398189      0 1.0000000 1.0501796 0.1477163
## 2 0.12200546      1 0.5660181 0.7791596 0.1533131
## 3 0.04581182      2 0.4440126 0.7373100 0.1620507
## 4 0.01000000      3 0.3982008 0.7685332 0.1679253
##
## Variable importance
## Illiteracy    HS_grad     Income Population
##         39         30         18         13
##
## Node number 1: 42 observations,    complexity param=0.4339819
##   mean=7.304762, MSE=12.94807
##   left son=2 (28 obs) right son=3 (14 obs)
##   Primary splits:
##       Illiteracy < 1.35    to the left,  improve=0.4339819, (0 missing)
##       HS_grad    < 44.3    to the right, improve=0.3066277, (0 missing)
##       Population < 3587    to the left,  improve=0.2203128, (0 missing)
##       Income     < 3891    to the right, improve=0.2076323, (0 missing)
##   Surrogate splits:
##       HS_grad    < 48.3    to the right, agree=0.881, adj=0.643, (0 split)
##       Income     < 3891    to the right, agree=0.833, adj=0.500, (0 split)
##       Population < 12048.5 to the left,  agree=0.690, adj=0.071, (0 split)
##
## Node number 2: 28 observations,    complexity param=0.1220055
##   mean=5.628571, MSE=8.172041
##   left son=4 (20 obs) right son=5 (8 obs)
##   Primary splits:
##       Population < 4678    to the left,  improve=0.28996470, (0 missing)
##       Income     < 4782    to the left,  improve=0.19220020, (0 missing)
##       HS_grad    < 53.25   to the right, improve=0.11258130, (0 missing)
```

```
##       Illiteracy < 0.75    to the left,  improve=0.08366059, (0 missing)
##   Surrogate splits:
##       HS_grad    < 53.25   to the right, agree=0.821, adj=0.375, (0 split)
##       Illiteracy < 0.75    to the left,  agree=0.786, adj=0.250, (0 split)
##
## Node number 3: 14 observations
##   mean=10.65714, MSE=5.642449
##
## Node number 4: 20 observations,    complexity param=0.04581182
##   mean=4.655, MSE=5.803475
##   left son=8 (11 obs) right son=9 (9 obs)
##   Primary splits:
##       HS_grad    < 59.1    to the left,  improve=0.21464160, (0 missing)
##       Income     < 4739    to the left,  improve=0.16079080, (0 missing)
##       Population < 779     to the right, improve=0.05969941, (0 missing)
##       Illiteracy < 0.65    to the right, improve=0.03516706, (0 missing)
##   Surrogate splits:
##       Illiteracy < 0.65    to the right, agree=0.8, adj=0.556, (0 split)
##       Income     < 4562    to the left,  agree=0.6, adj=0.111, (0 split)
##       Population < 1130.5  to the left,  agree=0.6, adj=0.111, (0 split)
##
## Node number 5: 8 observations
##   mean=8.0625, MSE=5.799844
##
## Node number 8: 11 observations
##   mean=3.645455, MSE=4.313388
##
## Node number 9: 9 observations
##   mean=5.888889, MSE=4.856543
```

```
rpart.plot(reg_tree_m2_cv_states) # plot the tree / we need to prune this
```




```
# random Forest model
rf_m2_cv_states <- randomForest(
  Murder ~ Income + Illiteracy + HS_grad + Population,
  train_k_states
)

plot(rf_m2_cv_states)
```



Overall, I prefer the regression tree because it appears much easier to interpret with the layout in the output. When you plot it, its also a much more visually informative model. This model demonstatively encapsulates the variable importance. They both identify the same information but the latter could have broader appear to a wider audience of interpreters.

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.

```
WBCD <- read.table("https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wiscons
                   header=FALSE,
                   sep=",")

colnames(WBCD)
```

```
##  [1] "V1"  "V2"  "V3"  "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10" "V11" "V12"
## [13] "V13" "V14" "V15" "V16" "V17" "V18" "V19" "V20" "V21" "V22" "V23" "V24"
## [25] "V25" "V26" "V27" "V28" "V29" "V30" "V31" "V32"
```

```
dim(WBCD)
```

```
## [1] 569  32
```

(b) Tidy the data, ensuring that each variable is properly named and cast as the correct data type. Discuss any 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. Be sure to address which of the errors that are identified you consider most problematic in this context.

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.

1) When a child arrives at the ER with a high fever, this can be a classification problem. What is causing the fever? There are multiple possibilities of what could be creating this situation. Some predictors could be infection (viral or bacterial), another could be underlying undiagnosed illness (e.g. Kawasaki's disease), and another could be Muckle-Wells syndrome (MWS) which is a multi-system inflammatory disorder. The predictors for each of these are qualitative in nature. For infection, you're going to culture the child to test for presence and identification of source infection, while Kawasaki's is a presentation of a set of qualitative features including prescence of erythema of conjunctiva, erythema of mouth, and additional predictors. This would be looking at prediction of the condition diagnosis. Once they predict that is what is happening, they would order tests specific to the conditions suspected to confirm diagnosis.

2) The identification of hate speech online is an example of classification modeling. Using a model trained on phrases or terms with conditions that identify hate speech, who the people are and identifier variables included. These terms or phrases can then be deployed in real time to flag phrases for evaluation. We would need to assess the prior set of probabilities for hate speech based on the platform. The model would be circular as it grows additional values with the continuum of time and filtering of the phrase bank. The classification is assessing probability ultimately if the speech is hate speech or not hate speech. This assessment would likely be looking at inference.

3) Elon Musk (the worst) wants to purchase Twitter but only if they "get rid of the bots". Twitter claims there are only about 5% bots on the platform while Elon suggests there are more. Determining who is a bot is a classification problem. They need inputs to assess whether a user is real or a bot which could look at IP, frequency and timestamp of publishing, key to output ratio; are they outputting much more text than they're typing suggested of a copy /paste? Once these data are evaluated, they would draw predict that the user is either a bot or not.

(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) Economic predictions at large are excellent regression problems whether it be the stock market, predicting inflation, or housing markets. Looking at the predictor variables from previous instances of these events can allow the model to point to variables of interest that may signify a change in the economy. This type of model would be drawing inference from the data to make predictions for future.

2) We could use regression to look at the influence of pediatric exposure to lead on a child's IQ. Looking at data of children and data of the houses known to contain lead based on various parameters, we could then compare to collected IQ scores (assuming they exist) to assess these influences. We couldn't do this experience in modern times because of ethical issues associated with allowing or passively encouraging exposure to lead, knowing there is a correlative detriment to humans. We could use historical or discovery (on-site) data. If for example, a child was determined to have exposure to lead and it had previously been unknown, that is data. Obviously that limits potential of collecting information but you could partner with multiple sites to collaboratively collect IQ assessments of kids within an age set if lead exposure was present. This model is inferring.

3) Examining the relationship between the age of a home and the price for the home sold in the last year could be a regression application. We could look at the build date of a home restricting by some bounds and compare the outcome variable of the house price to the age of the home. In this case, we would be predicting that age has some relationship to the current value of the home.

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

The advantages and disadvantages for having a flexible approach are explained by the bias-variance tradeoff. If we are very flexible, we can allow for a potentially better fit. This leads to less bias in our data. The disadvantage is that too much flexibility creates greater variance and can lead to overfitting the data. We must also consider how large our sample size is when leveraging flexibility.

Problem 5 (10 pts)

Suppose we have a dataset with five predictors, X_1 = GPA, X_2 = IQ, X_3 = Gender (1 for Female, and 0 for Male), X_4 = Interaction between GPA and IQ, and X_5 = Interaction between GPA and Gender. The response is starting salary after graduation (in thousands of dollars). Suppose we use least squares to fit the model and get $\hat{\beta}_0 = 50, \hat{\beta}_1 = 20, \hat{\beta}_2 = 0.07, \hat{\beta}_3 = 35, \hat{\beta}_4 = 0.01$, and $\hat{\beta}_5 = -10$.

(a) Which answer is correct and why?

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

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

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

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

Y = B0 + B1x + B2X^2 + B3X^3 + B4X^4 + B5X^5 + 0

Y = 50 + 0.01(x)^2 IQ and GPA + -10(x)^3 gender and GPA

#males_equation = 50 + (0.01 * (120 * 4.0)^2) + (-10 * (4.0 * 0)^3) # Y = 50 + 0.01(x)^2 + -10(x)^3 females #females_equation = 50 + (0.01 * (120 * 4.0)^2) + (-10 * (4.0 * 1)^3) # Y = B0 + B1x + B2X^2 + B3X^3 + B4X^4 + B5X^5 + 0 #checking_y_fem = 50 + (0.01 * (120)^2) + (35 * (1)^3) + (0.01 * (120 * 3.5)^3) + (-10 * (1)^3) # this feels wrong but leaving here to show thought processes #checking_y_mal = 50 + (0.01 * (120)^2) + (35 * (0)^3) + (0.01 * (120 * 3.5)^3) + (-10 * (0)^3)

Answer i is correct as the output Y value for the equation with a fixed value for IQ and GPA immediately shows that GPA is nullified for males with the 0 value. The value for Y shows that males earn just more than the females when GPA and IQ are fixed. The coefficient for interaction between gender and gpa is negative and just on face value of males being 0, we know that it will nullify this input and leave the females with the negative deduction.I used a highest fixed GPA for each which still outputs the same variation. Overall, I wasn't entirely certain of how to model this with the coefficients that have interactions as B4 and B5. The output Y value in my checking_y_fem and checking_y_mal seems quite high for a salary so I'm feeling less than certain of my outputs in this model.

(b) Predict the salary of a female with IQ of 110 and a GPA of 4.0.

Y = 50 + 0.07(x) + 20(1)^2 females

females_salary = 50 + (0.01 * (110 * 4.0)^2) + (-10 * (1 * 4.0)^3) For a female with an IQ of 110 and a GPA of 4.0, her salary would be $1346

(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.

The size of the coefficient estimate itself isn't what significies the interaction effect. I would say this is False. Even reviewing values above, we can see that some coefficient estimates were quite small but very significant. It would be helpful to have the sample size to know whether this is an enormous sample or a small sample.

Problem 6 - Extra Credit ($\leq$ 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.

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

```
# load the Boston data from MASS
data(Boston)

# set the random seed
set.seed(1978)
# create training data with 70/30 split
boston_split <- initial_split(Boston, prop = .7)
boston_train <- training(boston_split)
boston_test <- testing(boston_split)

# linear multivariate model of boston data
m1_boston <- lm(medv ~ ., data=Boston)
summary(m1_boston) #view summary
```

```
##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
## zn           4.642e-02  1.373e-02   3.382 0.000778 ***
## indus        2.056e-02  6.150e-02   0.334 0.738288
## chas         2.687e+00  8.616e-01   3.118 0.001925 **
## nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## age          6.922e-04  1.321e-02   0.052 0.958229
## dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
## ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## black        9.312e-03  2.686e-03   3.467 0.000573 ***
## lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

```
# create the function for the statistic of interest
boot.fn  <-  function(data, index){
  res <- coef(lm(medv ~ .,
```

```
                data = data,
                subset = index))
  return(res)
}

# Boostrap estimated of the standard error
boot(Boston, boot.fn, 1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Boston, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##           original        bias    std. error
## t1*   3.645949e+01 -9.119756e-01 7.758090197
## t2*  -1.080114e-01  5.669286e-03 0.035484271
## t3*   4.642046e-02 -4.581631e-04 0.013660170
## t4*   2.055863e-02 -1.289401e-03 0.052161041
## t5*   2.686734e+00 -1.499077e-02 1.284462755
## t6*  -1.776661e+01  4.908899e-01 3.850740170
## t7*   3.809865e+00  7.112029e-02 0.810803188
## t8*   6.922246e-04 -1.066249e-03 0.015761357
## t9*  -1.475567e+00  1.055678e-02 0.214640290
## t10*  3.060495e-01 -3.481144e-03 0.063505354
## t11* -1.233459e-02 -9.232974e-05 0.002810151
## t12* -9.527472e-01  9.115854e-03 0.117723880
## t13*  9.311683e-03  2.007646e-04 0.002843750
## t14* -5.247584e-01  3.456396e-03 0.095265055
```

```
# bagged model
bagged_boston <- bagging(medv ~ ., data = boston_train, coob = TRUE)
summary(bagged_boston)
```

```
##        Length Class      Mode
## y      354    -none-     numeric
## X       13    data.frame list
## mtrees  25    -none-     list
## OOB      1    -none-     logical
## comb     1    -none-     logical
## err      1    -none-     numeric
## call     4    -none-     call
```

```
boot_pred <- predict(bagged_boston, boston_test)
RMSE(boot_pred, boston_test$medv)
```

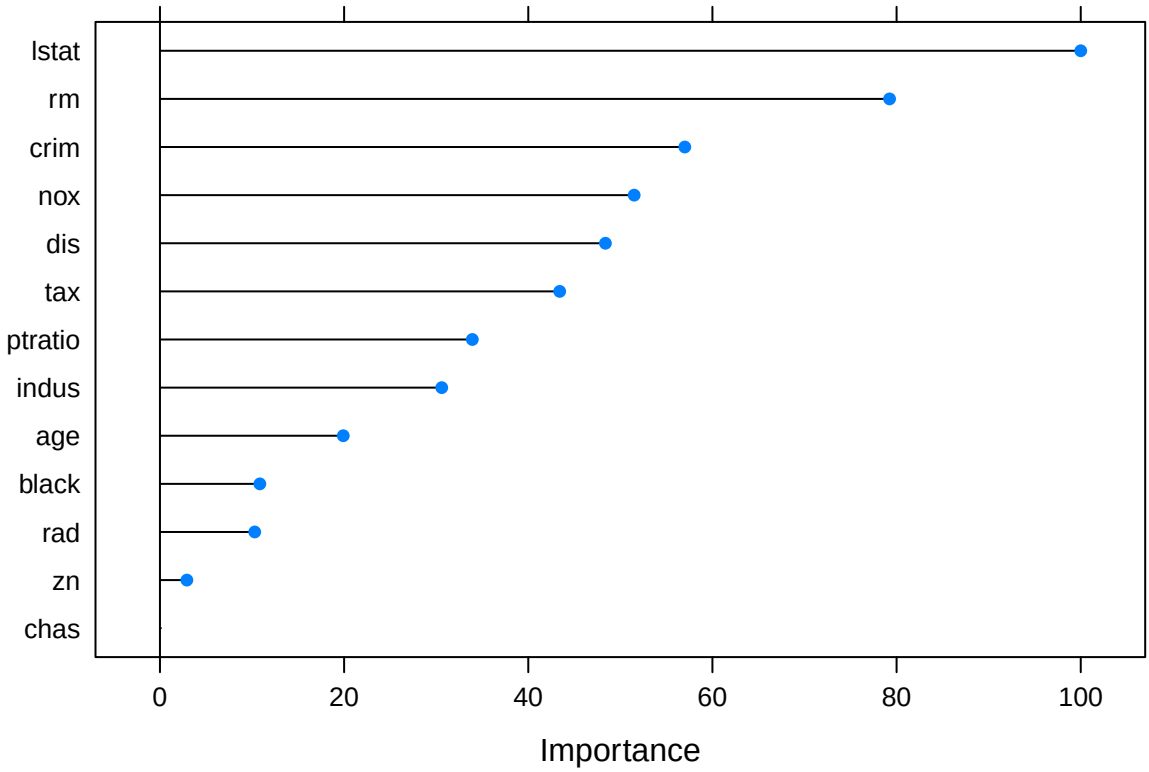
```
## [1] 3.587366
```

```
# cross validation with k fold (10) times
ctrl_bost <- trainControl(method = "cv", number = 10)

# cross validation
bagged_cv_boston <- train(
```

```
  medv ~ .,
  data = boston_train,
  method = "treebag",
  trControl = ctrl_bost,
  importance = TRUE
)

bagged_cv_boston
```

```
## Bagged CART
##
## 354 samples
##  13 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 319, 319, 318, 318, 319, 318, ...
## Resampling results:
##
##   RMSE      Rsquared  MAE
##   4.198643  0.7862    2.877801
```

```
#plot the bagged model
plot(varImp(bagged_cv_boston), 13)
```

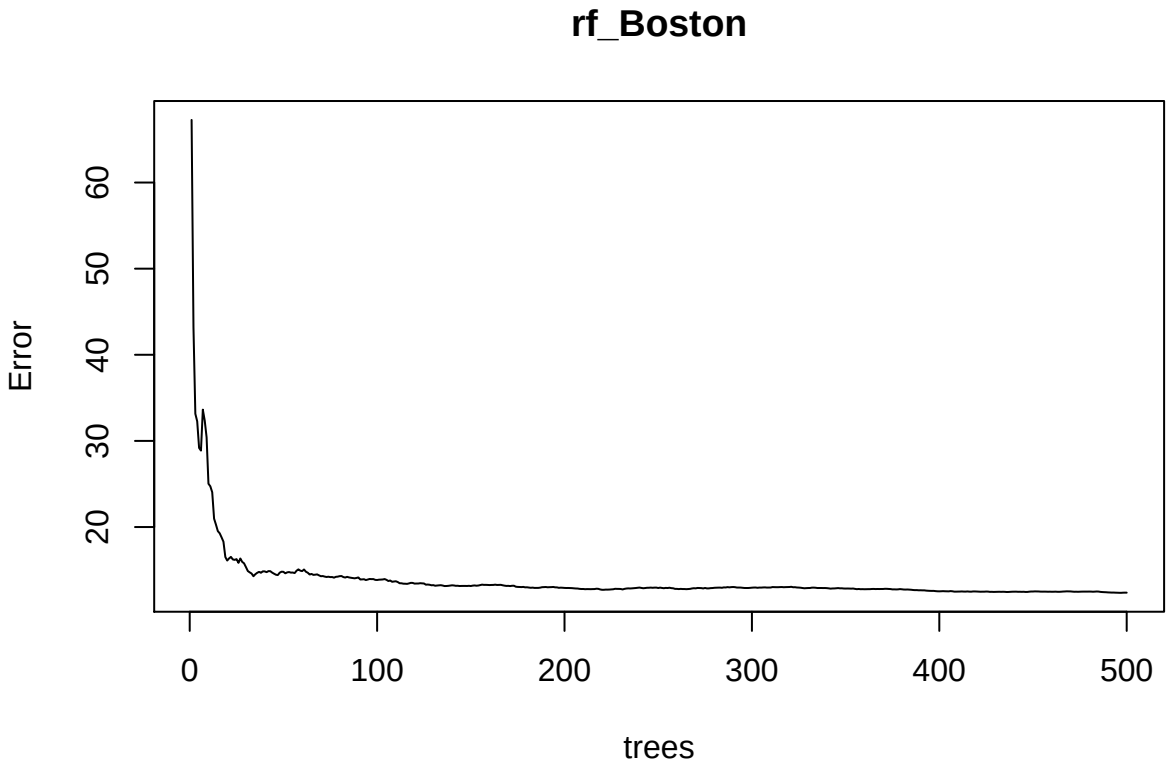



```
# random forest model of boston set
rf_Boston <- randomForest(
  medv ~ .,
  boston_train
)
```

```
rf_Boston
```

```
##
## Call:
##  randomForest(formula = medv ~ ., data = boston_train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 12.38169
##                     % Var explained: 85.11
```

```
plot(rf_Boston)
```



For the Boston set, I find the lm call much easier to interpret. As for accuracy, lm appears to be more valid. I'm not really very familiar with the boost and rnadomForest model interpretations. The standard error in the boostrap model versus the lm model are vastly different which infers I likely messed up the code.

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

Personally, the lm or glm function feels intuitively to be the best methodolgy to utilize.

Problem 7 - Extra Credit ($\leq$ 2 pts)

Suppose that $X_1, \ldots X_n$ form a random sample from a Poisson distribution for which the mean θ is unknown, $(\theta > 0)$.

(a) Determine the MLE of θ, assuming that at least one of the observed values is different from 0. Show your work.

(b) Show that the MLE of θ does not exists if every observed value is 0.

Citations

Access a URL and read Data with R https://stackoverflow.com/questions/6299220/access-a-url-and-read-data-with-r

k-fold Cross Validation in R https://www.youtube.com/watch?v=BQ1VAZ7jNYQ

Machine Learning for Cancer Diagnosis and Prognosis http://archive.ics.uci.edu/ml

BestGLM https://cran.r-project.org/web/packages/bestglm/bestglm.pdf

ScatterplotMatrix https://www.rdocumentation.org/packages/car/versions/2.1-6/topics/scatterplotMatrix

Rpart https://cran.r-project.org/web/packages/rpart.plot/rpart.plot.pdf

M09 - Resampling Methods (IMT573)