IMT 573 Final Exam

Vighnesh Misal

Due: December 10, 2019

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. 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 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.

You 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 being accepted.

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.

Vighnesh

12/08/2019

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

```r
library(tidyverse)
library(AER)
library(leaps)
library(bestglm)
library(dplyr)
library(ROCR)
library(caTools)
library(caret)
```

(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 package in . This data comes from a survey conducted by 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).

```
data("Affairs")
```

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.

```
nrow(Affairs)

## [1] 601

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 ...

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
```

```
nrow(filter(Affairs, gender == "female"))
```

```
## [1] 315
```

```
mean(Affairs$age)
```

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

```
max(Affairs$age)
```

```
## [1] 57
```

```
min(Affairs$age)
```

```
## [1] 17.5
```

```
table(Affairs$gender)
```

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

```
mean(Affairs$yearsmarried)
```

```
## [1] 8.177696
```

```
table(Affairs$yearsmarried)
```

```
## 
## 0.125 0.417  0.75   1.5     4     7    10    15 
##    11    10    31    88   105    82    70   204
```

```
table(Affairs$affairs)
```

```
## 
##   0   1   2   3   7  12 
## 451  34  17  19  42  38
```

```
table(Affairs$children)
```

```
## 
##  no yes 
## 171 430
```

```
table(Affairs$rating)
```

```
## 
##   1   2   3   4   5 
##  16  66  93 194 232
```

```
table(Affairs$religiousness)
```

```
##
##   1   2   3   4   5
##  48 164 129 190  70
```

The participants are aged between 17.5 years to 57 years with an average age of 32.49 years.

There are 315 males and 286 females.

The participants have been married for a mean period of 8.2 years.

430 participants have kids and 171 don't have kids

Over 50% of the participants rate their marriage as "very happy" and "happy"

A significant portion of people don't identify as being very religious.

Most people don't have never engaged in an extra-marital affair based off their responses

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?

```
Affairs$affair_binary <-ifelse(Affairs$affairs >0 ,1,0)
```

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.

```
affair_model <- lm(affair_binary ~
gender+age+yearsmarried+children+religiousness+education+occupation+rating,
data=Affairs)
summary(affair_model)

##
## Call:
## lm(formula = affair_binary ~ 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
```

The age, yearsmarried, religiousness and rating variables seem to have an impact on whether or not a person is likely to engage in extra-marital affairs.

The gender, children, education and occupation variables don't seem to share a correlation with whether or not a person is likely to engage in extra marital affair.

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 function available in the package helpful.

```
bestglm(Affairs,IC = "BIC", family = binomial())

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## Warning: glm.fit: algorithm did not converge

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

## BIC
## BICq equivalent for q in (0, 0.96080783109275)
## Best Model:
##               Estimate Std. Error      z value  Pr(>|z|)
## (Intercept) -24.25407   5278.105 -0.004595224 0.9963336
## affairs      46.04833   7709.257  0.005973122 0.9952342
```

```
models <-
regsubsets(affair_binary~gender+age+yearsmarried+children+religiousness+educa
tion+occupation+rating, data = Affairs, nvmax = 8)
summary(models)
```

```
## Subset selection object
## Call: regsubsets.formula(affair_binary ~ gender + age + yearsmarried +
##     children + religiousness + education + occupation + rating,
##     data = Affairs, nvmax = 8)
## 8 Variables  (and intercept)
##               Forced in Forced out
## gendermale        FALSE      FALSE
## age               FALSE      FALSE
## yearsmarried      FALSE      FALSE
## childrenyes       FALSE      FALSE
## religiousness     FALSE      FALSE
## education         FALSE      FALSE
## occupation        FALSE      FALSE
## rating            FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          gendermale age yearsmarried childrenyes religiousness education
## 1  ( 1 ) " "        " " " "          " "         " "           " "      
## 2  ( 1 ) " "        " " " "          " "         "*"           " "      
## 3  ( 1 ) " "        " " "*"          " "         "*"           " "      
## 4  ( 1 ) " "        "*" "*"          " "         "*"           " "      
## 5  ( 1 ) "*"        "*" "*"          " "         "*"           " "      
## 6  ( 1 ) "*"        "*" "*"          "*"         "*"           " "      
## 7  ( 1 ) "*"        "*" "*"          "*"         "*"           " "      
## 8  ( 1 ) "*"        "*" "*"          "*"         "*"           "*"      
##          occupation rating
## 1  ( 1 ) " "        "*"   
## 2  ( 1 ) " "        "*"   
## 3  ( 1 ) " "        "*"   
## 4  ( 1 ) " "        "*"   
## 5  ( 1 ) " "        "*"   
## 6  ( 1 ) " "        "*"   
## 7  ( 1 ) "*"        "*"   
## 8  ( 1 ) "*"        "*"
```

```r
res.sum <- summary(models)
data.frame(
  Adj.R2 = which.max(res.sum$adjr2),
  CP = which.min(res.sum$cp),
  BIC = which.min(res.sum$bic),
  Rsq = res.sum$rsq
)
```

```
##   Adj.R2 CP BIC        Rsq
## 1      6  5   3 0.06442177
## 2      6  5   3 0.07978786
## 3      6  5   3 0.09249694
## 4      6  5   3 0.09891862
## 5      6  5   3 0.10392045
## 6      6  5   3 0.10562559
## 7      6  5   3 0.10640041
## 8      6  5   3 0.10659639
```

The current model is completely different from the one obtained in 1.c.

Based on the observed results, it depends which parameter we consider for the best model. If we consider the Adjusted R2 scores, then it becomes evident that the 6 variable model consisting of gender, age, yearsmarried, children, religiousness and rating is the viable option and the best model to use.

Based on the observed results, it depends which parameter we consider for the best model. If we consider the CP scores, then it becomes evident that the 5 variable model consisting of gender, age, yearsmarried, religiousness and rating is the viable option and the best model to use.

Based on the observed results, it depends which parameter we consider for the best model. If we consider the BIC scores, then it becomes evident that the 6 variable model consisting of yearsmarried, religiousness and rating is the viable option and the best model to use.

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

Rating: is an important factor that determines how well a marriage is and people who rate their marriage poorly are more prone to have affairs outside marriage.

Religiousness: is an important factor as it determines the beliefs that people have and people who adhere strictly to religious doctrines are less likely to have extra-marital affairs for the fear of retribution.

yearsmarried: is an important factor as people with longer marriages tend to indulge in extra-marital affairs as monotony of the relationship begins to set-in.

gender: is an important factor as males and females are wired differently.

children : is an important factor as people with kids tend to be wary of indulging in extra marital affairs as they might cause their family to breakdown.

Age: is an important factor as older people are more likely to cheat.

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 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.

```
new_affairs <- select(Affairs, -affairs)

new_affairs$gender <- factor(new_affairs$gender, levels=c("male","female"),
labels=c(0,1))

new_affairs$children <- factor(new_affairs$children, levels=c("yes","no"),
labels=c(1,0))

new_affairs$age <- mean(new_affairs$age)
new_affairs$yearsmarried <- mean(new_affairs$yearsmarried)
new_affairs$education <- mean(new_affairs$education)
new_affairs$religiousness <- mean(new_affairs$religiousness)
new_affairs$occupation <- mean(new_affairs$occupation)

bestglm(new_affairs,IC = "BIC", family = binomial())

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

## BIC
## BICq equivalent for q in (2.15825454841223e-07, 0.61230882371022)
## Best Model:
##               Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)  0.8253902 0.32548132  2.535907 1.121566e-02
## rating      -0.5082193 0.08468845 -6.001046 1.960510e-09

var1 <- lm(new_affairs$affair_binary ~ new_affairs$rating +
new_affairs$religiousness + new_affairs$gender + new_affairs$age +
new_affairs$yearsmarried + new_affairs$children + new_affairs$education +
new_affairs$occupation, data=new_affairs)

summary(var1)

##
## Call:
## lm(formula = new_affairs$affair_binary ~ new_affairs$rating +
##     new_affairs$religiousness + new_affairs$gender + new_affairs$age +
##     new_affairs$yearsmarried + new_affairs$children +
new_affairs$education +
##     new_affairs$occupation, data = new_affairs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56501 -0.28588 -0.16765 -0.07461  0.92539
##
## Coefficients: (5 not defined because of singularities)
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.65806    0.06556  10.037  < 2e-16 ***
## new_affairs$rating        -0.09304    0.01577  -5.899 6.13e-09 ***
## new_affairs$religiousness       NA         NA      NA       NA
## new_affairs$gender1       -0.03757    0.03422  -1.098   0.2727
## new_affairs$age                 NA         NA      NA       NA
## new_affairs$yearsmarried        NA         NA      NA       NA
## new_affairs$children0     -0.08066    0.03863  -2.088   0.0372 *
## new_affairs$education           NA         NA      NA       NA
## new_affairs$occupation          NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

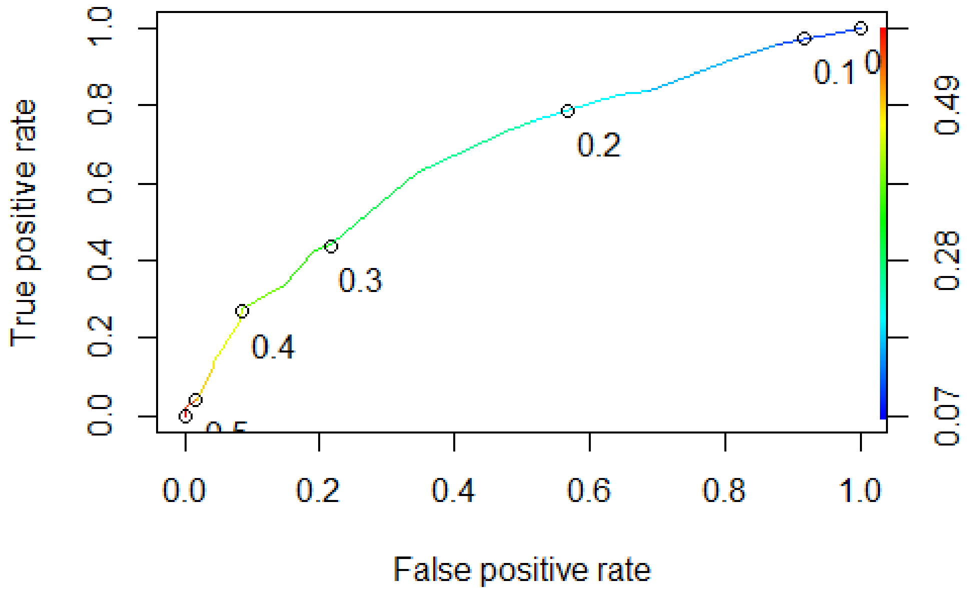
```
##
## Residual standard error: 0.4179 on 597 degrees of freedom
## Multiple R-squared:  0.07359,    Adjusted R-squared:  0.06894
## F-statistic: 15.81 on 3 and 597 DF,  p-value: 6.653e-10

predicted_val <- predict(var1,new_affairs,type = 'response')

## Warning in predict.lm(var1, new_affairs, type = "response"): prediction
## from a rank-deficient fit may be misleading

ROCR_pred <- prediction(predicted_val , new_affairs$affair_binary)
ROCR_perf <- performance(ROCR_pred , measure = "tpr", x.measure = "fpr")

plot(ROCR_perf , colorize = TRUE, text.adj = c(-0.2,1.7), print.cutoffs.at =
seq(0,1,0.1))
```



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

Small subset: The amount of data available to train the model is too low to give a correct prediction about someone's tendency to cheat.

There are other factors such as behaviour, socio-economic status, culture of the place that also determine a person's ability to cheat. Some people might cheat due to abusive spouses. Some people might cheat because they have a chance at a better life, some people might have sociopathic tendencies and find it difficult to settle down with a partner.

Different definition of marriages and what is defined as having an affair. A lot of people have said they have never cheated in the current dataset. There is no way of knowing whether they are lying or not. Also, definition of having an affair varies and needs to be defined for the study.

The dataset also fails to mention the reason for someone cheating on their spouse.

(10 pts)

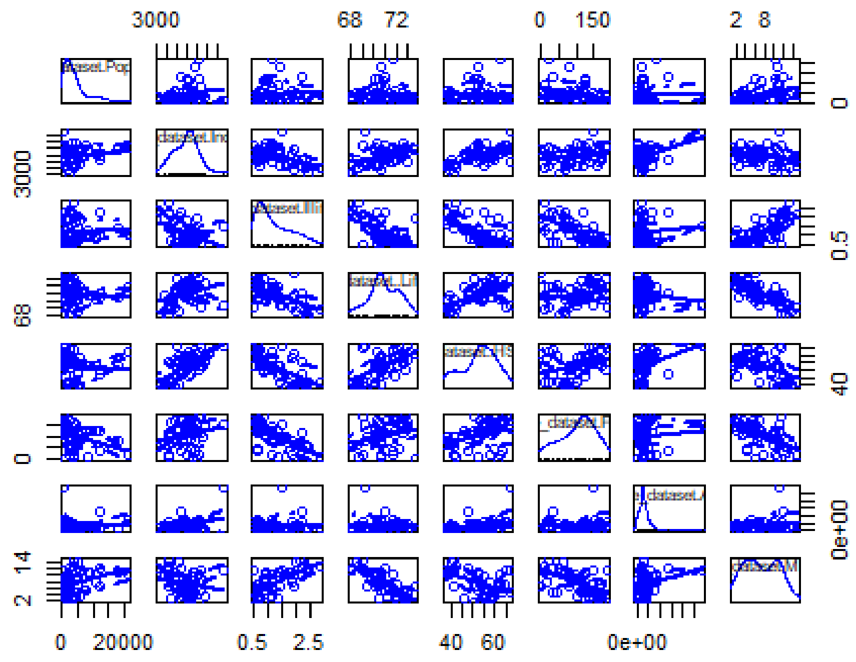
In this problem we will revisit the dataset. This data, available as part of the base package, contains various data related to the 50 states of the United States of America.

```
state_dataset <- tbl_df(state.x77)
```

Suppose you want to explore the relationship between a state's rate and other characteristics of the state, for example population, illiteracy rate, and more. Follow the questions below to perform this analysis.

Examine the bivariate relationships present in the data. Briefly discuss notable results. You might find the function available in the package helpful.

```
scatterplotMatrix( ~ state_dataset$Population + state_dataset$Income +
state_dataset$Illiteracy + state_dataset$`Life Exp` + state_dataset$'HS Grad'
+ state_dataset$Frost + state_dataset$Area + state_dataset$Murder, data =
state_dataset)
```



```
cor_subset <- state_dataset[c("Population", "Income", "Illiteracy","Life
Exp","HS Grad","Frost","Area","Murder")]

cor(cor_subset, use = "complete.obs")

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

There is a high degree of positive correlation between the murder rate and illiteracy rate.

There is a high degree of negative correlation between the murder rate and life expectancy

There is a high degree of negative correlation between the murder rate and HS grad rate

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

```
fit <- lm(state_dataset$Murder ~ state_dataset$Population +
state_dataset$Income + state_dataset$Illiteracy + state_dataset$`Life Exp` +
state_dataset$'HS Grad' + state_dataset$Frost + state_dataset$Area ,
data=state_dataset)

summary(fit)

##
## Call:
## lm(formula = state_dataset$Murder ~ state_dataset$Population +
##     state_dataset$Income + state_dataset$Illiteracy + state_dataset$`Life
Exp` +
##     state_dataset$"HS Grad" + state_dataset$Frost + state_dataset$Area,
##     data = state_dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4452 -1.1016 -0.0598  1.1758  3.2355
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.222e+02  1.789e+01   6.831 2.54e-08 ***
## state_dataset$Population  1.880e-04  6.474e-05   2.905  0.00584 **
## state_dataset$Income     -1.592e-04  5.725e-04  -0.278  0.78232
## state_dataset$Illiteracy  1.373e+00  8.322e-01   1.650  0.10641
## state_dataset$`Life Exp` -1.655e+00  2.562e-01  -6.459 8.68e-08 ***
## state_dataset$"HS Grad"    3.234e-02  5.725e-02   0.565  0.57519
## state_dataset$Frost       -1.288e-02  7.392e-03  -1.743  0.08867 .
## state_dataset$Area         5.967e-06  3.801e-06   1.570  0.12391
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.746 on 42 degrees of freedom
## Multiple R-squared:  0.8083, Adjusted R-squared:  0.7763
## F-statistic: 25.29 on 7 and 42 DF,  p-value: 3.872e-13
```

The murder rate should ideally increase with an increase in the population as is evident from the results of our calculation.

The murder rate also decreases as income increases as people with money will normally not indulge in crime for materialistic gains.

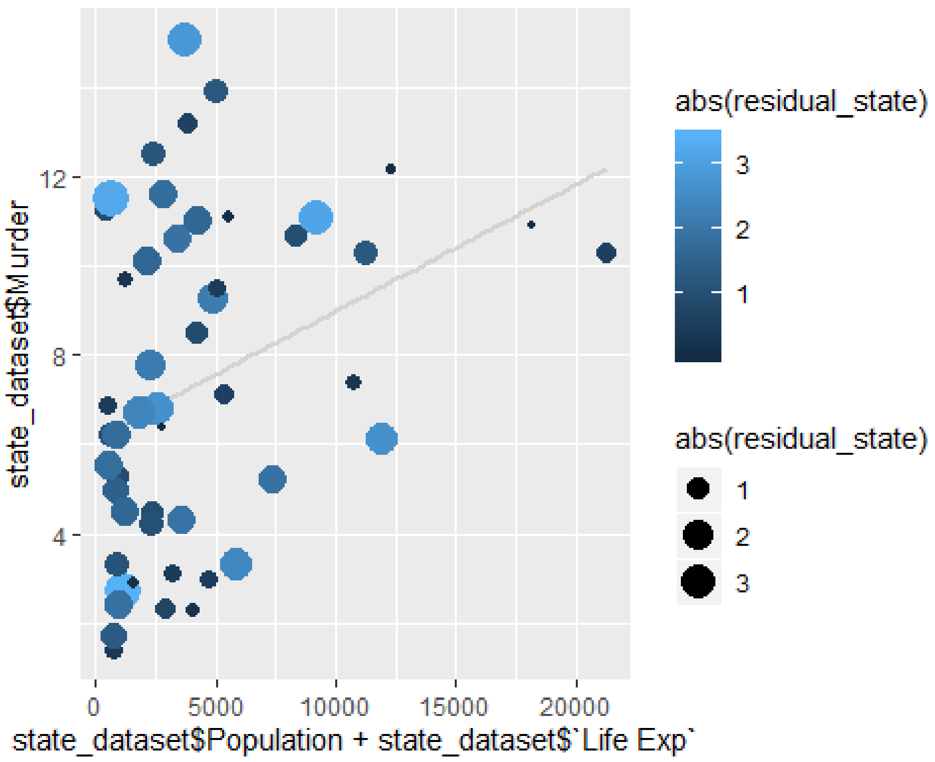
The murder rate increases slightly with an increase in the .

The murde rate increases with rising illiteracy as people who aren't well educated have trouble finding jobs which makes them likely to perpetrate a crime.

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.

```
predicted_state <- predict(fit)
residual_state <- residuals(fit)

ggplot(state_dataset, aes(x = state_dataset$Population + state_dataset$`Life
Exp`, y = state_dataset$Murder)) + geom_smooth(method = "lm", se = FALSE,
color = "lightgrey") + geom_point(aes(color = abs(residual_state), size =
abs(residual_state)))
```



Some variables have a very low degree of correlation to the murder rate and I haven't included them as they have a minimal impact but I feel that the accuracy of the model might be sacrificed in this instance similar to the butterfly effect.

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?

```
nullmodel=lm(Murder~1, data=state_dataset)
fullmodel=lm(Murder~., data=state_dataset)
model.step = step(nullmodel, scope=list(lower=nullmodel, upper=fullmodel),
direction='both')

## Start:  AIC=131.59
## Murder ~ 1
## 
##                Df Sum of Sq    RSS     AIC
## + `Life Exp`   1    407.14 260.61  86.550
## + Illiteracy   1    329.98 337.76  99.516
## + Frost        1    193.91 473.84 116.442
## + `HS Grad`    1    159.00 508.75 119.996
## + Population   1     78.85 588.89 127.311
## + Income       1     35.35 632.40 130.875
## + Area         1     34.83 632.91 130.916
## <none>                     667.75 131.594
## 
## Step:  AIC=86.55
## Murder ~ `Life Exp`
```

```
## 
##                Df Sum of Sq    RSS     AIC
## + Frost         1     80.10 180.50  70.187
## + Illiteracy    1     60.55 200.06  75.329
## + Population    1     56.62 203.99  76.303
## + Area          1     14.12 246.49  85.764
## <none>                      260.61  86.550
## + `HS Grad`     1      1.12 259.48  88.334
## + Income        1      0.96 259.65  88.366
## - `Life Exp`    1    407.14 667.75 131.594
## 
## Step:  AIC=70.19
## Murder ~ `Life Exp` + Frost
## 
##                Df Sum of Sq    RSS     AIC
## + Population    1    23.710 156.79  65.146
## + Area          1    21.084 159.42  65.976
## <none>                      180.50  70.187
## + Illiteracy    1     6.066 174.44  70.477
## + Income        1     5.560 174.94  70.622
## + `HS Grad`     1     2.068 178.44  71.610
## - Frost         1    80.104 260.61  86.550
## - `Life Exp`    1   293.331 473.84 116.442
## 
## Step:  AIC=65.15
## Murder ~ `Life Exp` + Frost + Population
## 
##                Df Sum of Sq    RSS     AIC
## + Area          1    19.040 137.75  60.672
## + Illiteracy    1    11.826 144.97  63.225
## <none>                      156.79  65.146
## + `HS Grad`     1     1.821 154.97  66.561
## + Income        1     0.739 156.06  66.909
## - Population    1    23.710 180.50  70.187
## - Frost         1    47.198 203.99  76.303
## - `Life Exp`    1   296.694 453.49 116.247
## 
## Step:  AIC=60.67
## Murder ~ `Life Exp` + Frost + Population + Area
## 
##                Df Sum of Sq    RSS     AIC
## + Illiteracy    1     8.723 129.03  59.402
## <none>                      137.75  60.672
## + Income        1     1.241 136.51  62.220
## + `HS Grad`     1     0.771 136.98  62.392
## - Area          1    19.040 156.79  65.146
## - Population    1    21.666 159.42  65.976
## - Frost         1    52.970 190.72  74.940
## - `Life Exp`    1   272.927 410.68 113.290
## 
```

```
## Step:  AIC=59.4
## Murder ~ `Life Exp` + Frost + Population + Area + Illiteracy
## 
##                Df Sum of Sq    RSS    AIC
## <none>                      129.03 59.402
## - Illiteracy  1     8.723 137.75 60.672
## + `HS Grad`   1     0.763 128.27 61.105
## + Income      1     0.026 129.01 61.392
## - Frost       1    11.030 140.06 61.503
## - Area        1    15.937 144.97 63.225
## - Population  1    26.415 155.45 66.714
## - `Life Exp`  1   140.391 269.42 94.213
```

Yes, the step wise selection model is different from the one in b. We obtain a model that consists of Life Exp + Frost + Population + Area + Illiteracy and it considers a lot more variables and is more accurate than the model envisioned in b.

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

```
set.seed(123)
train.control <- trainControl(method = "repeatedcv",number = 10, repeats = 3)
# Train the model
model <- train(Murder ~ ., data = state_dataset, method = "lm",trControl = 
train.control)
# Summarize the results
print(model)
```

```
## Linear Regression 
## 
## 50 samples
##  7 predictor
## 
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times) 
## Summary of sample sizes: 44, 46, 45, 45, 44, 46, ... 
## Resampling results:
## 
##   RMSE      Rsquared   MAE     
##   1.922634  0.7550216  1.608819
## 
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

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.

(5 pts)

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

Obtain the data, and load it into by pulling it directly from the web. (Do download it and import it from a CSV file.) Give a brief description of the data.

```
url1 = 'http://archive.ics.uci.edu/ml/machine-learning-databases/breast-
cancer-wisconsin/breast-cancer-wisconsin.data'

cancer_data = read.csv(url(url1), header = FALSE, sep = ",")

str(cancer_data)
```

```
## 'data.frame':    699 obs. of  11 variables:
##  $ V1 : int  1000025 1002945 1015425 1016277 1017023 1017122 1018099
1018561 1033078 1033078 ...
##  $ V2 : int  5 5 3 6 4 8 1 2 2 4 ...
##  $ V3 : int  1 4 1 8 1 10 1 1 1 2 ...
##  $ V4 : int  1 4 1 8 1 10 1 2 1 1 ...
##  $ V5 : int  1 5 1 1 3 8 1 1 1 1 ...
##  $ V6 : int  2 7 2 3 2 7 2 2 2 2 ...
##  $ V7 : Factor w/ 11 levels "?","1","10","2",..: 2 3 4 6 2 3 3 2 2 2 ...
##  $ V8 : int  3 3 3 3 3 9 3 3 1 2 ...
##  $ V9 : int  1 2 1 7 1 7 1 1 1 1 ...
##  $ V10: int  1 1 1 1 1 1 1 1 5 1 ...
##  $ V11: int  2 2 2 2 2 4 2 2 2 2 ...
```

The data consists of ID, Diagnosis and ten variables that define the various attributes of a breast cell that have been observed i.e. radius,texture,perimeter,smoothness,compactness,concavity,concave_points,symmetry,fractal_dimension and each variable has three types i.em mean, standard error and worst (the max value of that variable for that particular cell)

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.

```
cancer_data <- cancer_data %>%
  rename(
    ID = V1,
    Clump_Thickness = V2,
    Uniformity_of_Cell_Size  = V3,
    Uniformity_of_Cell_Shape = V4,
    Marginal_Adhesion = V5,
    Single_Epithelial_Cell_Size = V6,
    Bare_Nuclei = V7,
    Bland_Chromatin = V8,
    Normal_Nucleoli = V9,
    Mitoses = V10,
    Class = V11,
        )
cancer_data$Class <- as.factor(cancer_data$Class)


sum(is.na(cancer_data))

## [1] 0

table(cancer_data$Class)

##
##   2   4
## 458 241
```

There is no missing data in this dataset.

There are 458 benign tumors and 241 malignant tumors in the dataset

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

```
set.seed(101)
sample = sample.split(cancer_data, SplitRatio = .70)
cancer_data_train = subset(cancer_data, sample == TRUE)
cancer_data_test  = subset(cancer_data, sample == FALSE)
```

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.

```
logitmod <- glm(cancer_data_train$Class ~
cancer_data_train$Clump_Thickness+cancer_data_train$Uniformity_of_Cell_Size+c
ancer_data_train$Uniformity_of_Cell_Shape+cancer_data_train$Marginal_Adhesion
+cancer_data_train$Single_Epithelial_Cell_Size+cancer_data_train$Bare_Nuclei+
cancer_data_train$Bland_Chromatin+cancer_data_train$Normal_Nucleoli+cancer_da
ta_train$Mitoses, family = "binomial", data=cancer_data_train)

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

summary(logitmod)

##
## Call:
## glm(formula = cancer_data_train$Class ~ cancer_data_train$Clump_Thickness
+
##     cancer_data_train$Uniformity_of_Cell_Size +
cancer_data_train$Uniformity_of_Cell_Shape +
##     cancer_data_train$Marginal_Adhesion +
cancer_data_train$Single_Epithelial_Cell_Size +
##     cancer_data_train$Bare_Nuclei + cancer_data_train$Bland_Chromatin +
##     cancer_data_train$Normal_Nucleoli + cancer_data_train$Mitoses,
##     family = "binomial", data = cancer_data_train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.73635  -0.05510  -0.03000   0.00793   2.10464
##
## Coefficients:
##                                                  Estimate Std. Error
## (Intercept)                                    -15.06741    3.37266
## cancer_data_train$Clump_Thickness                0.51830    0.20562
## cancer_data_train$Uniformity_of_Cell_Size        0.01777    0.32540
## cancer_data_train$Uniformity_of_Cell_Shape       0.69788    0.38582
## cancer_data_train$Marginal_Adhesion              0.26550    0.17080
## cancer_data_train$Single_Epithelial_Cell_Size   -0.08922    0.21552
## cancer_data_train$Bare_Nuclei1                   3.29734    2.15847
## cancer_data_train$Bare_Nuclei10                  6.68219    2.19700
## cancer_data_train$Bare_Nuclei2                   2.92392    2.31667
## cancer_data_train$Bare_Nuclei3                   5.99859    2.15461
## cancer_data_train$Bare_Nuclei4                   7.79599    2.57932
## cancer_data_train$Bare_Nuclei5                   2.94993    2.13234
## cancer_data_train$Bare_Nuclei6                  23.70947 3800.80060
## cancer_data_train$Bare_Nuclei7                   2.86555    2.78030
## cancer_data_train$Bare_Nuclei8                   3.18857    2.25897
## cancer_data_train$Bare_Nuclei9                  21.86667 2320.95877
## cancer_data_train$Bland_Chromatin                0.68979    0.29092
## cancer_data_train$Normal_Nucleoli                0.23717    0.20239
## cancer_data_train$Mitoses                        0.60814    0.46803
##                                                z value Pr(>|z|)
## (Intercept)                                     -4.468 7.91e-06 ***
## cancer_data_train$Clump_Thickness                2.521  0.01171 *
## cancer_data_train$Uniformity_of_Cell_Size        0.055  0.95646
## cancer_data_train$Uniformity_of_Cell_Shape       1.809  0.07048 .
## cancer_data_train$Marginal_Adhesion              1.554  0.12008
## cancer_data_train$Single_Epithelial_Cell_Size   -0.414  0.67888
## cancer_data_train$Bare_Nuclei1                   1.528  0.12660
## cancer_data_train$Bare_Nuclei10                  3.042  0.00235 **
## cancer_data_train$Bare_Nuclei2                   1.262  0.20691
## cancer_data_train$Bare_Nuclei3                   2.784  0.00537 **
## cancer_data_train$Bare_Nuclei4                   3.023  0.00251 **
## cancer_data_train$Bare_Nuclei5                   1.383  0.16653
## cancer_data_train$Bare_Nuclei6                   0.006  0.99502
## cancer_data_train$Bare_Nuclei7                   1.031  0.30270
## cancer_data_train$Bare_Nuclei8                   1.412  0.15809
## cancer_data_train$Bare_Nuclei9                   0.009  0.99248
## cancer_data_train$Bland_Chromatin                2.371  0.01774 *
## cancer_data_train$Normal_Nucleoli                1.172  0.24127
## cancer_data_train$Mitoses                        1.299  0.19382
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 571.445  on 444  degrees of freedom
## Residual deviance:  51.034  on 426  degrees of freedom
## AIC: 89.034
##
## Number of Fisher Scoring iterations: 17
```

```
pred <- predict(logitmod, newdata = cancer_data_test, type = "response")
```

```
## Warning: 'newdata' had 254 rows but variables found have 445 rows
```

```
y_pred_num <- ifelse(pred > 0.5, 4, 2)
y_pred <- factor(y_pred_num, levels=c('4', '2'))
y_act <- cancer_data_train$Class

mean(y_pred == y_act)
```

```
## [1] 0.9775281
```

```
confusionMatrix(y_pred,y_act)
```

```
## Warning in confusionMatrix.default(y_pred, y_act): Levels are not in the
## same order for reference and data. Refactoring data to match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   2   4
##          2 287   4
##          4   6 148
##
##                Accuracy : 0.9775
##                  95% CI : (0.9591, 0.9892)
##     No Information Rate : 0.6584
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9502
##
##  Mcnemar's Test P-Value : 0.7518
##
##             Sensitivity : 0.9795
##             Specificity : 0.9737
##          Pos Pred Value : 0.9863
##          Neg Pred Value : 0.9610
##              Prevalence : 0.6584
##          Detection Rate : 0.6449
##    Detection Prevalence : 0.6539
##       Balanced Accuracy : 0.9766
```

```
##
##        'Positive' Class : 2
##
```

According to the confusion matrix there are 4 Type-I errors and 6 Type-II errors. Type-I errors occur when the null hypothesis is true but is rejected i.e. it gives rise to a false positive which is the test validated a difference and statistically significant even though there isn’t one.

Type-II errors occur when the null hypothesis is false but we fail to reject it. Getting a type-II error is dangerous as it means that we stated that our hypothesis is incorrect despite it actually being valid.

(10 pts)

Please answer the questions below by writing a short response.

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

Email spam classification. The goal is to classify emails that are recieved by an email address as spam and not and send the emails that are classified as spam to a spam folder. The predictor variables are origin of email address, subject line of email, text within the email,human classification of whether the email is spam or not. Response variable is whether the email is spam or not. The application of this is for prediction purposes. Based on previous knowledge of spam address origin as well as the frequency of certain terms within the message the model can predict whether or not an email is classified as spam or not. This will use the Naive-Bayesian classifier.

Weather Prediction. The goal is to classify the weather in advance based on historical weather data as well the measured variables that exist in real time to create a model for classification. The predictor variables are humidity, temperature, pressure, wind direction, wind speed, min temperature, max temperature, visibility, precipitation, observed weather. The response variable is predicted weather such as windy, snowy, cloudy, stormy etc. The application of this is for predicting the weather by computing a model that takes into consideration the various factors that have a major impact as well as give a clear indication of the type of weather to expect.

Credit card application processing. The goal is to classify applicants who apply for a credit card. The bank will compare the data of these new applicants with previous applicants to observe and classify similar applicants as being worthy of being a credit card or not. The predictor variables are: credit rating, monthly expense, monthly income, credit card bill overwithdrawal, timely payments, number of registered credit card offences. The response variable is classifying the candidate as safe or unsafe for credit card approval. This is a prediction problem as we are trying to classify individuals as low-risk v/s high risk who can be given a credit card and it can be used to identify individuals who will not default on their payments. The data from historical transactions of known defaulters can help us predict this in advance as the data points are a tell tale sign of this phenomenon.

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

Predicting the price of houses. We can use regression to extrapolate the price of houses based on certain factors. Predictor variables: house area, type of house, area, locality, crime rate, closest distance to hospitals,closest distance to schools, number of bedrooms, type of neighbourhood. Response variable is price of the house. This is a prediction problem. It's possible to fit a linear regressoin model that fits the data and then predict values that lie on the line of best fit.

Sports analysis. Sports analysts can use regression to predict player performance based on previous data as well as current player fitness variables and can form a team based upon the predicted player rating value for the next season. Predictor variables:current rating, player age, goals scored, distance run, touchdowns, ball drops, tackles. The response variable is a value normalized between 0 and 1 that shows how ready a player is for the next season.

Predicting the value of a car based on make,model,manufacture year, condition, type, mileage.

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 advantage of having a flexible approach is that we can model highly complex systems and nonlinear problems as well as keep bias as low as possible.

The disadvantage of a flexible approach is that we can have a higher variance or overfit the training data as well as make it incredibly difficult to interpret.

The flexible approach is used when the model is underfitted and when the data has characteristics that can be considered as nonlinear.

A less flexible apporach might be preferred for data that has very few rows of or when data has characteristics that can be considered linear.

(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.

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

```
# P(Math_sectionA) = 1/5
# P(Math_sectionA)*P(Math_sectionA)
1/5 * 1/5

## [1] 0.04
```

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

```
# P(Chemistry_sectionF) = 1/8
# P(Chemistry_sectionF)*P(Chemistry_sectionF)
1/8 * 1/8

## [1] 0.015625
```

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

```
# P(Math_section) = 1/5
# P(Chemistry_section) = 1/8
comb = function(n, x) {
  factorial(n) / factorial(n-x) / factorial(x)
}
#
5C1*P(Math_sectionA)*P(Math_sectionA)*8C1*P(Chemistry_sectionF)*P(Chemistry_s
ectionF)

comb(5,1)*(1/5)*(1/5)*comb(8,1)*(1/8)*(1/8)

## [1] 0.025
```

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

```
# P(sectionA and sectionD) = 1/4
# P(sectionA) = P(sectionD) = 1/2
# 2C1*P(sectionA and sectionD)*P(sectionA)
comb(2,1)*1/4 * 1/2

## [1] 0.25
```

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

```
# P(SECTION B,C,D) = 1/6
# P(SECTION B) = 1/3
# 3C1*P(SECTION B,C,D)*P(SECTION B)

comb(3,1)*1/6 * 1/3

## [1] 0.1666667
```

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 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(MATH)*P(SECTION)*P(MATH)*P(SECTION)

1/4 * 1/5 *1/4 * 1/4

## [1] 0.003125
```

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?

```
#4C1*(P(MATH_SECTION)*P(MATH_SECTION)+P(CHEM_SECTION)*P(CHEM_SECTION)+P(PHYSI
CS_SECTION)*P(PHYSICS_SECTION)+P(HISTORY_SECTION)*P(HISTORY_SECTION))

comb(4,1)*((1/5 * 1/5) + (1/8 * 1/8)+(1/6 * 1/6)+(1/4 * 1/4))

## [1] 0.5836111
```

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.

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(T + SP) = P(T) + P(SP) - P(T and SP)
# P(T and SP)=
0.8+0.3-1

## [1] 0.1
```

(≤ 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.

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

Which of the approaches yields the best performance?