

Assignment 6

STAT 3150–Statistical Computing

Due on 11/12/2020

This assignment focuses on the modules on **Numerical Methods** and **Optimisation**. It also uses ideas and concepts from previous modules.

Solutions must be submitted electronically via UM Learn **no later than 11:59PM CDT on Friday December 11th**. Please provide both the Rmd file and a PDF version of the output of compiling it. Also, please use the following filename convention: “LastnameFirstname_STAT3150-Assignment6.Rmd”.

You are allowed to discuss the problems among yourselves, but your submission must reflect your original work. Note that your R code will be analyzed for suspicious similarities.

For problems (or parts thereof) that require you to mathematically compute a derivative or likelihood, you may attach your hand-written solutions as an appendix. In this case your solution should refer to this appendix (e.g. “Please see appendix for the derivation of the likelihood.”).

In Problems 3 and 4, you are asked to use datasets stored in CSV files on UM Learn. To import them into R, you can use the function `readr::read_csv`.

```
library(bootstrap)
```

```
## Warning: package 'bootstrap' was built under R version 4.0.3
```

```
library(boot)
```

```
## Warning: package 'boot' was built under R version 4.0.3
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.3
```

```
## -- Attaching packages ----- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2      v purrr   0.3.4
```

```
## v tibble  3.0.4      v dplyr   1.0.2
```

```
## v tidyr   1.1.2      v stringr 1.4.0
```

```
## v readr   1.4.0      v forcats 0.5.0
```

```
## Warning: package 'tibble' was built under R version 4.0.3

## Warning: package 'readr' was built under R version 4.0.3

## Warning: package 'dplyr' was built under R version 4.0.3

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

Problem 1

Recall the `law` dataset in the `bootstrap` package, which contains information on average LSAT and GPA scores for 15 law schools. We are interested in the correlation ρ between these two variables.

Using numerical methods, compute a p-value for the null hypothesis of *moderate correlation*, i.e.

$$H_0 : \text{cor}(\text{LSAT}, \text{GPA}) = 0.5,$$

by inverting the **percentile** bootstrap confidence interval.

```
n<-nrow(law)
data<-law
func<-function(a,b){ cor(a,b)}
data.booted<-boot(data,func,R=999)
fun<-function(alpha){
  CIdata<-boot.ci(data.booted,conf = alpha,type = "perc")
  data<-CIdata$percent
  CI<-c(data[4],data[5])
  CI[2]-.5#returns the difference between the upper bound of the interval and the null hypothesis value
}

output<- uniroot(f=fun,interval=c(.9,.99),tol=10^-10)

CI<-boot.ci(data.booted,conf = output$root,type = "perc")
CI
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 999 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = data.booted, conf = output$root, type = "perc")
```

```
##
## Intervals :
## Level      Percentile
## 94.0145213125965%   (-0.5165,  0.5000 )
## Calculations and Intervals on Original Scale
```

```
p_val<-(1-CI$percent[1])
p_val
```

```
## [1] 0.05985479
```

As you can see from the smallest confidence percent that yielded a confidence interval where $\text{cor}=0.5$ was contained, the p-value is ~ 0.06

Problem 2

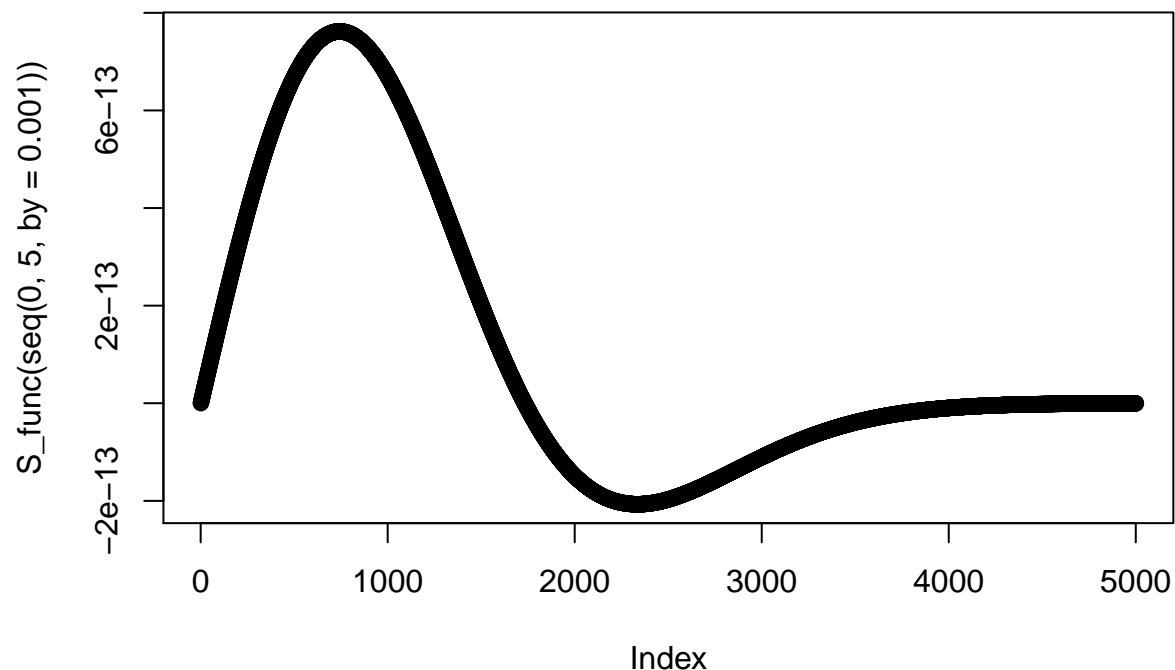
Find the intersection points of the following two curves (as functions of $a \geq 0$)

$$S_{k-1}(a) = P\left(t_{k-1} > \sqrt{\frac{a^2(k-1)}{k-a^2}}\right),$$

$$S_k(a) = P\left(t_k > \sqrt{\frac{a^2k}{k+1-a^2}}\right),$$

for $k = 4, 25, 100$, where t_k is a Student t random variable with k degrees of freedom. There should be exactly two intersection points for each value of k .

```
k=425100
S1_func<- function(a){
  pt(sqrt(a^2*(k-1))/(k-(a^2)),k-1)
}
S_func<-function(a){
  val1<-sqrt(a^2*(k-1)/(k-a^2))
  val2<-sqrt(a^2*(k)/(k+1-a^2))
  pt(val2,k)-pt(val1,k-1)
}
plot(S_func(seq(0,5,by=.001)))#guesses for bounds gained from this plot
```



```
output<- uniroot(f=S_func,interval=c(0.00001,1.8),tol=10-10)
c(output$root,S1_func(output$root))#first intersection point
```

```
## [1] 1.7320360 0.5010598
```

```
output<- uniroot(f=S_func,interval=c(1.8,10),tol=10-10)
intersect<-output$root
intersect
```

```
## [1] 10
```

This first intersection point is at (1.7320360 0.5010598) and then around $a=5$, the curves start to line up and be equal to each other as x goes to infinity. # Problem 3

In this problem, we combine numerical methods with bootstrap. We also quickly mention the **survival function**, which is studied in more details in STAT 4170.

We will use the data in `rayleigh.csv`. Assume that the data follow a Rayleigh distribution with density

$$f(x) = \frac{x}{\lambda^2} \exp(-x^2/(2\lambda^2)), \quad x > 0$$

and cumulative distribution function

$$F(x) = 1 - \exp(-x^2/(2\lambda^2)).$$

- Using numerical methods, compute the Maximum Likelihood estimate $\hat{\lambda}$ of λ .
- Transform the data by rounding it down to the nearest integer. Write down the likelihood for this binned data.
- Using numerical methods, compute the Maximum Likelihood estimate $\hat{\lambda}_B$ of λ using the binned data.
- Using bootstrap, compute a 95% confidence interval for both $\hat{\lambda}$ and $\hat{\lambda}_B$. You can use the method of your choice, but use the same method for both intervals. Compare their lengths. Provide a possible explanation for why one is wider than the other.
- The **survival function** of the Rayleigh distribution can be defined by

$$S(t; \lambda) = \exp\left(-\int_0^t \frac{x}{2\lambda^2} dx\right).$$

Using either numerical or Monte Carlo integration, compute an estimate of $S(2.5; \hat{\lambda})$.

- Using bootstrap, compute a 95% confidence interval for $S(2.5; \hat{\lambda})$.

- The likelihood function is

$$\prod_{i=1}^n \frac{X_i}{\lambda^2} e^{-\frac{X_i^2}{2\lambda^2}}$$

The log-Likelihood function is

$$\ln\left(\prod X_i\right) - \ln\prod \lambda^2 + \ln\prod e^{-\frac{X_i^2}{2\lambda^2}} = \sum \ln X_i - 2n \ln \lambda + \sum \ln e^{-\frac{X_i^2}{2\lambda^2}} = \sum \ln X_i - 2n \ln \lambda + \sum \frac{-X_i^2}{2\lambda^2} = \sum \ln X_i - 2n \ln \lambda +$$

The derivative of this is

$$\frac{d}{d\lambda} \sum \ln X_i - 2n \ln \lambda + \frac{\sum -X_i^2}{2\lambda^2} = 0 - \frac{2n}{\lambda} + \frac{-2 \sum -X_i^2}{2\lambda^3} = \frac{-2n}{\lambda} + \frac{\sum X_i^2}{\lambda^3}$$

```
xvals <- read.csv(file = 'rayleigh.csv')
n<-nrow(xvals)
log_lik_der <- function(lambda){
  ((-2*n)/(lambda))+(sum(xvals)/(lambda^3))
}
output<-uniroot(log_lik_der,c(0.0000001,10))
lambda_hat<-output$root
lambda_hat
```

```
## [1] 1.12289
```

The value for lambda hat is the value of \$root = 1.12289.

- The bins are intervals [0,1) [1,2) [2,3) etc.

The probability of being in a bin [a,b) is $F(b) - F(a) = 1 - \exp(-b^2/(2\lambda^2)) - 1 + \exp(-a^2/(2\lambda^2)) = \exp(-a^2/(2\lambda^2)) - \exp(-b^2/(2\lambda^2))$.

The likelihood function is

$$\prod e^{\frac{-a_i^2}{2\lambda^2}} - e^{\frac{-b_i^2}{2\lambda^2}}$$

the log likelihood function is

$$\ln \prod e^{\frac{-a_i^2}{2\lambda^2}} - e^{\frac{-b_i^2}{2\lambda^2}} = \sum \ln (e^{\frac{-a_i^2}{2\lambda^2}} - e^{\frac{-b_i^2}{2\lambda^2}})$$

the derivative of this is

$$\frac{d}{d\lambda} \sum \ln (e^{\frac{-a_i^2}{2\lambda^2}} - e^{\frac{-b_i^2}{2\lambda^2}}) = \sum \frac{e^{\frac{-a_i^2}{2\lambda^2}} \frac{a_i^2}{\lambda^3} + e^{\frac{-b_i^2}{2\lambda^2}} \frac{b_i^2}{\lambda^3}}{e^{\frac{-a_i^2}{2\lambda^2}} - e^{\frac{-b_i^2}{2\lambda^2}}}$$

c)

```
rounded<-floor(xvals)
max<-max(rounded)
n<-nrow(rounded)
a_vec <-0:max
b_vec<-1:max
n_vec <-0:max
for(i in 1:max+1){
  n_vec[i]<-0
}
for(i in 1:n){
  n_vec[rounded$obs[i]+1]<- n_vec[rounded$obs[i]+1]+1
}

log_like_der_binned <- function(lambda){
  num<- exp(-(a_vec^2)/2*lambda^2)*(a_vec^2/(lambda^3)) + ((exp(-(b_vec^2)/2*lambda^2))*(b_vec^2)/(lambda^3))
  num<-num/(exp(-(a_vec^2)/2*lambda^2)*(a_vec^2/(lambda^3)) - exp(-(b_vec^2)/2*lambda^2)*(b_vec^2/(lambda^3)))
  num[max] <-exp(-(max^2)/2*lambda^2)*(max^2/(lambda^3)) + ((exp(-((max+1)^2)/2*lambda^2))*((max+1)^2)/(lambda^3))
  num[max]<- num[max]/(exp(-(max^2)/2*lambda^2)*(max^2/(lambda^3)) - exp(-((max+1)^2)/2*lambda^2))*((max+1)^2/(lambda^3))
  sum(num)
}
output_binned<-uniroot(log_like_der_binned,c(0.001,1))
print("")
```

```
## [1] ""
```

```
print("lambda hat for the binned data is")
```

```
## [1] "lambda hat for the binned data is"
```

```
lambda_binned_hat<-output_binned$root
lambda_binned_hat
```

```
## [1] 0.5695798
```

d)

```
options(warn=-0)
```

```
B<-1000
```

```
boot_results <-replicate(B, {
  indices <-sample(n, n,replace =TRUE)
  log_lik_der_boot <- function(lambda){
    ((-2*n)/(lambda))+(sum(xvals[indices,])/(lambda^3))
  }
  output<-uniroot(log_lik_der_boot,c(0.0000001,10))
  output$root
})
```

```
binned_boot_results <-replicate(B, {
  indices <-sample(n, n,replace =TRUE)
  max<-0
  for(i in 1:n){
    max<-max(max,rounded$obs[indices])
  }
  a_vec <-0:max
  b_vec<-1:max
  n_vec <-0:max
  for(i in 1:max+1){
    n_vec[i]<-0
  }
  n_vec[rounded$obs[indices]+1]<- n_vec[rounded$obs[indices]+1]+1
  log_lik_der_binned_boot <- function(lambda){
    num<- exp(-(a_vec^2)/2*lambda^2)*(a_vec^2/(lambda^3)) + ((exp(-(b_vec^2)/2*lambda^2))*(b_vec^2/(lambda^3))
    num<-num/(exp(-(a_vec^2)/2*lambda^2)*(a_vec^2/(lambda^3)) - exp(-(b_vec^2)/2*lambda^2)*(b_vec^2/(lambda^3))
    num[max] <-exp(-(max^2)/2*lambda^2)*(max^2/(lambda^3)) + ((exp(-((max+1)^2)/2*lambda^2))*((max+1)^2)/(lambda^3))
    num[max]<- num[max]/(exp(-(max^2)/2*lambda^2)*(max^2/(lambda^3)) - exp(-((max+1)^2)/2*lambda^2)*((max+1)^2/(lambda^3))
    sum(num)
  }
  output<-uniroot(log_lik_der_binned_boot,c(0.0000001,1))
  output$root
```

```

})
print("")

## [1] ""

print("The percentile CI for non binned is")

## [1] "The percentile CI for non binned is"

quantile(boot_results,probs =c(0.025,0.975))

##      2.5%      97.5%
## 1.087634 1.159988

```

```

print("The percentile CI for binned is")

## [1] "The percentile CI for binned is"

quantile(binned_boot_results,probs =c(0.025,0.975))

##      2.5%      97.5%
## 0.4055099 0.5696119

```

As you can see the margin in the non binned CI is greater than the margin in the binned CI. This is likely due to the fact that points that may be farther apart get grouped together (1,1.5,1.9,etc. all get grouped to 1) and so the resulting integers don't deviate from each other as much as there are many occurrences of the same value

d)

```

rvals<-runif(1000,min = 0,max=2.5)
mean(exp(-rvals/2*lambda_hat^2))

```

```
## [1] 0.5038287
```

e)


```
survival_func<-function(x,lamb){
  mean(exp(-x/2*lamb^2))
}
survival_boot <-replicate(B, {
  indices <-sample(n, n,replace =TRUE)
  mean(survival_func(rvals[indices],lambda_hat))
})
quantile(survival_boot,probs =c(0.025,0.975))

##      2.5%      97.5%
## 0.4821102 0.5401666
```

Problem 4

In this problem, we will study **logistic regression**, which is a generalization of linear regression for binary outcomes. This topic will be studied in more details in STAT 3550.

Let Y be a binary outcome, and let X be a covariate. Define

$$\mu(X) = \beta_0 + \beta_1 X,$$

where β_0, β_1 are two regression parameters. In logistic regression, we assume the following relationship between Y and X :

$$P(Y = 1 | X) = \frac{\exp(\mu(X))}{1 + \exp(\mu(X))}.$$

Let $(Y_1, X_1), \dots, (Y_n, X_n)$ be a random sample. In this problem, we are interested in fitting a logistic regression model to the data in `breast_cancer.csv`, where `Class` is the outcome (`1 = "malignant"`, `0 = "benign"`) and `Cell.shape` is the covariate. We will use the Newton-Raphson algorithm to find the Maximum Likelihood estimate.

- Write down the log-likelihood function $\ell(\theta)$ for $\theta = (\beta_0, \beta_1)$.
- Compute the first-order partial derivatives of $\ell(\theta)$.
- Compute the second-order partial derivatives of $\ell(\theta)$.
- Find the Maximum Likelihood estimate $\hat{\theta} = (\hat{\beta}_0, \hat{\beta}_1)$ for the data in `breast_cancer.csv` using the Newton-Raphson algorithm.

- the likelihood function is

$$\prod \frac{e^{\beta_0 + \beta_1 X_i}}{1 + e^{\beta_0 + \beta_1 X_i}}$$

the log likelihood of this function is

$$\ln \prod \frac{e^{\beta_0 + \beta_1 X_i}}{1 + e^{\beta_0 + \beta_1 X_i}} = \ln \sum \frac{e^{\beta_0 + \beta_1 X_i}}{1 + e^{\beta_0 + \beta_1 X_i}} = \sum (\ln e^{\beta_0 + \beta_1 X_i} - \ln (1 + e^{\beta_0 + \beta_1 X_i})) = \sum (\beta_0 + \beta_1 X_i - \ln (1 + e^{\beta_0 + \beta_1 X_i}))$$

b) The first order derivative with respect to β_0 is

$$\begin{aligned}
\frac{d\theta}{d\beta_0} &= \frac{d}{d\beta_0} \sum (e^{\beta_0+\beta_1 X_i} - \ln(1 + e^{\beta_0+\beta_1 X_i})) = \sum (e^{\beta_0+\beta_1 X_i} * (1+0) - \frac{1}{(1 + e^{\beta_0+\beta_1 X_i})} * e^{\beta_0+\beta_1 X_i} * (1+0)) \\
&= \sum (e^{\beta_0+\beta_1 X_i} - \frac{e^{\beta_0+\beta_1 X_i}}{1 + e^{\beta_0+\beta_1 X_i}}) = \sum (e^{\beta_0+\beta_1 X_i} * \frac{(1 + e^{\beta_0+\beta_1 X_i})}{1 + e^{\beta_0+\beta_1 X_i}} - \frac{e^{\beta_0+\beta_1 X_i}}{1 + e^{\beta_0+\beta_1 X_i}}) \\
&= \sum (\frac{e^{\beta_0+\beta_1 X_i} * (1 + e^{\beta_0+\beta_1 X_i}) - e^{\beta_0+\beta_1 X_i}}{1 + e^{\beta_0+\beta_1 X_i}}) = \sum (\frac{e^{\beta_0+\beta_1 X_i} ((1 + e^{\beta_0+\beta_1 X_i}) - 1)}{1 + e^{\beta_0+\beta_1 X_i}}) \\
&= \sum (\frac{e^{\beta_0+\beta_1 X_i} (e^{\beta_0+\beta_1 X_i})}{1 + e^{\beta_0+\beta_1 X_i}}) = \sum (\frac{e^{2(\beta_0+\beta_1 X_i)}}{1 + e^{\beta_0+\beta_1 X_i}})
\end{aligned}$$

The first order derivative with respect to β_1 is

$$\begin{aligned}
\frac{d\theta}{d\beta_1} &= \frac{d}{d\beta_1} \sum (e^{\beta_0+\beta_1 X_i} - \ln(1 + e^{\beta_0+\beta_1 X_i})) = \sum e^{\beta_0+\beta_1 X_i} * (X_i) - \frac{e^{\beta_0+\beta_1 X_i} * (X_i)}{(1 + e^{\beta_0+\beta_1 X_i})} \\
&= \sum e^{\beta_0+\beta_1 X_i} * (X_i) - \frac{e^{\beta_0+\beta_1 X_i} * (X_i)}{(1 + e^{\beta_0+\beta_1 X_i})} = \sum \frac{e^{\beta_0+\beta_1 X_i} * (X_i)(1 + e^{\beta_0+\beta_1 X_i})}{(1 + e^{\beta_0+\beta_1 X_i})} - \frac{e^{\beta_0+\beta_1 X_i} * (X_i)}{(1 + e^{\beta_0+\beta_1 X_i})} \\
&= \sum \frac{e^{\beta_0+\beta_1 X_i} * (X_i)(1 + e^{\beta_0+\beta_1 X_i}) - e^{\beta_0+\beta_1 X_i} * (X_i)}{(1 + e^{\beta_0+\beta_1 X_i})} = \sum \frac{e^{\beta_0+\beta_1 X_i} * (X_i) * ((1 + e^{\beta_0+\beta_1 X_i}) - 1)}{(1 + e^{\beta_0+\beta_1 X_i})} \\
&= \sum \frac{e^{\beta_0+\beta_1 X_i} * (X_i) * (e^{\beta_0+\beta_1 X_i} + 1 - 1)}{(1 + e^{\beta_0+\beta_1 X_i})} = \sum \frac{e^{2(\beta_0+\beta_1 X_i)} * (X_i)}{(1 + e^{\beta_0+\beta_1 X_i})}
\end{aligned}$$

c)

The second order partial derivative with respect to β_0 is

$$\begin{aligned}
\frac{d^2\theta}{d\beta_0^2} &= \frac{d}{d\beta_0} \sum (\frac{e^{2(\beta_0+\beta_1 X_i)}}{1 + e^{\beta_0+\beta_1 X_i}}) \\
&= \sum (\frac{\frac{d}{d\beta_0} e^{2(\beta_0+\beta_1 X_i)} (1 + e^{\beta_0+\beta_1 X_i}) - e^{2(\beta_0+\beta_1 X_i)} \frac{d}{d\beta_0} (1 + e^{\beta_0+\beta_1 X_i})}{(1 + e^{\beta_0+\beta_1 X_i})^2}) \\
&= \sum (\frac{e^{2(\beta_0+\beta_1 X_i)} * 2(1+0)(1 + e^{\beta_0+\beta_1 X_i}) - e^{2(\beta_0+\beta_1 X_i)} (0 + e^{\beta_0+\beta_1 X_i} * (1+0))}{(1 + e^{\beta_0+\beta_1 X_i})^2}) \\
&= \sum (\frac{e^{2(\beta_0+\beta_1 X_i)} * 2(1 + e^{\beta_0+\beta_1 X_i}) - e^{2(\beta_0+\beta_1 X_i)} (e^{\beta_0+\beta_1 X_i})}{(1 + e^{\beta_0+\beta_1 X_i})^2}) \\
&= \sum (\frac{2e^{2(\beta_0+\beta_1 X_i)} + 2e^{3(\beta_0+\beta_1 X_i)} - e^{3(\beta_0+\beta_1 X_i)}}{(1 + e^{\beta_0+\beta_1 X_i})^2}) \\
&= \sum (\frac{2e^{2(\beta_0+\beta_1 X_i)} + e^{3(\beta_0+\beta_1 X_i)}}{(1 + e^{\beta_0+\beta_1 X_i})^2}) \\
&= \sum (\frac{e^{2(\beta_0+\beta_1 X_i)} (2 + e^{\beta_0+\beta_1 X_i})}{(1 + e^{\beta_0+\beta_1 X_i})^2})
\end{aligned}$$

The second order partial derivative with respect to β_1 is

$$\begin{aligned}
\frac{d^2 l\theta}{d\beta_1^2} &= \frac{d}{d\beta_1} \sum \left(\frac{e^{2(\beta_0+\beta_1 X_i)} * (X_i)}{(1 + e^{\beta_0+\beta_1 X_i})} \right) \\
&= \sum \left(\frac{\frac{d}{d\beta_1} e^{2(\beta_0+\beta_1 X_i)} * (X_i) (1 + e^{\beta_0+\beta_1 X_i}) - e^{2(\beta_0+\beta_1 X_i)} * (X_i) \frac{d}{d\beta_1} (1 + e^{\beta_0+\beta_1 X_i})}{(1 + e^{\beta_0+\beta_1 X_i})^2} \right) \\
&= \sum \left(\frac{e^{2(\beta_0+\beta_1 X_i)} * 2X_i * (X_i) (1 + e^{\beta_0+\beta_1 X_i}) - e^{2(\beta_0+\beta_1 X_i)} * (X_i) (0 + e^{\beta_0+\beta_1 X_i} * X_i)}{(1 + e^{\beta_0+\beta_1 X_i})^2} \right) \\
&= \sum \left(\frac{e^{2(\beta_0+\beta_1 X_i)} * 2X_i^2 (1 + e^{\beta_0+\beta_1 X_i}) - e^{3(\beta_0+\beta_1 X_i)} * (X_i)^2}{(1 + e^{\beta_0+\beta_1 X_i})^2} \right) \\
&= \sum \left(\frac{e^{2(\beta_0+\beta_1 X_i)} * 2X_i^2 + e^{3(\beta_0+\beta_1 X_i)} 2X_i^2 - e^{3(\beta_0+\beta_1 X_i)} * (X_i)^2}{(1 + e^{\beta_0+\beta_1 X_i})^2} \right) \\
&= \sum \left(\frac{(X_i^2) (2e^{2(\beta_0+\beta_1 X_i)} + 2e^{3(\beta_0+\beta_1 X_i)} - e^{3(\beta_0+\beta_1 X_i)})}{(1 + e^{\beta_0+\beta_1 X_i})^2} \right) \\
&= \sum \left(\frac{(X_i^2) (2e^{2(\beta_0+\beta_1 X_i)} + e^{3(\beta_0+\beta_1 X_i)})}{(1 + e^{\beta_0+\beta_1 X_i})^2} \right) \\
&= \sum \left(\frac{(X_i^2) e^{2(\beta_0+\beta_1 X_i)} (2 + e^{\beta_0+\beta_1 X_i})}{(1 + e^{\beta_0+\beta_1 X_i})^2} \right)
\end{aligned}$$

The second order partial derivative with respect to both β_0 and β_1 is

$$\begin{aligned}
\frac{d^2 l\theta}{d\beta_1 d\beta_0} &= \frac{d}{d\beta_0} \sum \frac{e^{2(\beta_0+\beta_1 X_i)} * (X_i)}{(1 + e^{\beta_0+\beta_1 X_i})} \\
&= \sum \frac{\left(\frac{d}{d\beta_0} (e^{2(\beta_0+\beta_1 X_i)} * (X_i)) \right) * (1 + e^{\beta_0+\beta_1 X_i})^2 - (e^{2(\beta_0+\beta_1 X_i)} * (X_i)) * \left(\frac{d}{d\beta_0} (1 + e^{\beta_0+\beta_1 X_i})^2 \right)}{(1 + e^{\beta_0+\beta_1 X_i})^2} \\
&= \sum \frac{e^{2(\beta_0+\beta_1 X_i)} * (X_i) * (2(1 + 0)) * (1 + e^{\beta_0+\beta_1 X_i})^2 - e^{2(\beta_0+\beta_1 X_i)} * (X_i) * 2(1 + e^{\beta_0+\beta_1 X_i}) * (0 + e^{\beta_0+\beta_1 X_i} * (1 + 0))}{(1 + e^{\beta_0+\beta_1 X_i})^2} \\
&= \sum \frac{2X_i (e^{2(\beta_0+\beta_1 X_i)}) (1 + e^{\beta_0+\beta_1 X_i})^2 - (X_i) (e^{2(\beta_0+\beta_1 X_i)}) 2(1 + e^{\beta_0+\beta_1 X_i}) (e^{\beta_0+\beta_1 X_i})}{(1 + e^{\beta_0+\beta_1 X_i})^2} \\
&= \sum \frac{(2X_i) e^{2(\beta_0+\beta_1 X_i)} ((1 + e^{\beta_0+\beta_1 X_i})^2 - (1 + e^{\beta_0+\beta_1 X_i}) (e^{\beta_0+\beta_1 X_i}))}{(1 + e^{\beta_0+\beta_1 X_i})^2} \\
&= \sum \frac{(2X_i) e^{2(\beta_0+\beta_1 X_i)} ((1 + 2e^{\beta_0+\beta_1 X_i} + e^{2(\beta_0+\beta_1 X_i)}) - (e^{\beta_0+\beta_1 X_i} + e^{2(\beta_0+\beta_1 X_i)}))}{(1 + e^{\beta_0+\beta_1 X_i})^2} \\
&= \sum \frac{(2X_i) e^{2(\beta_0+\beta_1 X_i)} (1 + 2e^{\beta_0+\beta_1 X_i} + e^{2(\beta_0+\beta_1 X_i)} - e^{\beta_0+\beta_1 X_i} - e^{2(\beta_0+\beta_1 X_i)})}{(1 + e^{\beta_0+\beta_1 X_i})^2} \\
&= \sum \frac{(2X_i) e^{2(\beta_0+\beta_1 X_i)} (1 + e^{\beta_0+\beta_1 X_i})}{(1 + e^{\beta_0+\beta_1 X_i})^2} \\
&= \sum \frac{(2X_i) e^{2(\beta_0+\beta_1 X_i)}}{(1 + e^{\beta_0+\beta_1 X_i})}
\end{aligned}$$

d)

```

bc_data<-read.csv("breast_cancer.csv")
y_vec<- bc_data$Class
x_vec<-bc_data$Cell.shape

gradient <-function(theta) {
  mu_x <-theta[1]+theta[2]*x_vec
  ell_b0 <-sum(y_vec*(exp(mu_x)^2)/(1+exp(mu_x)))#only including values where the outcome was Yi=1, if
  ell_b1 <-sum(y_vec*(exp(mu_x)^2)*x_vec/(1+exp(mu_x)))
  return(c(ell_b0, ell_b1))
}

hessian <-function(theta) {
  mu_x <-theta[1]+theta[2]*x_vec
  ell_b0b0 <--sum(y_vec*(exp(mu_x)^2)*(2+exp(mu_x))/((1+exp(mu_x))^2))
  ell_b1b1 <--sum(y_vec*(x_vec^2)*(exp(mu_x)^2)*(2+exp(mu_x))/((1+exp(mu_x))^2))
  ell_b0b1 <--sum(y_vec*(2*x_vec)*(exp(mu_x)^2)/(1+exp(mu_x)))
  hess <-matrix(c(ell_b0b0, ell_b0b1,
                  ell_b0b1, ell_b1b1)
                ,ncol =2)
  return(hess)
}

x_curr<- c(0,1)
x_next<- x_curr - solve(hessian(x_curr),gradient(x_curr))
tol<-10^-10
iter<-1

while(sum((x_curr-x_next)^2)>tol&iter<100){
  iter<-iter+1
  x_curr<-x_next
  step<- solve(hessian(x_curr),gradient(x_curr))
  x_next<-x_curr-step
}
x_next

```

```
## [1] 33.314690 4.348113
```

```

fit<-glm(Class~Cell.shape, data = bc_data, family = binomial)
coef(fit)

```

```

## (Intercept) Cell.shape
## -5.061758 1.406753

```

*#my answers were off I believe its in how i set up the original likelihood function
#but assuming that my likelihood function was correct,
#every step I did afterwards should be correct
#which includes the log-likelihood function and all the derivatives*