MATH3821 Assignment 1

Kevin Li, Kevin Lou, Stephen Sung, Brandon Wong, Jason Zhu

Question 1

For Simple Linear Regression model $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$.

a) Let $\beta_0 = \alpha - \beta_1 \bar{x}$. Then the SLR model can be expressed as $y_i = \alpha + \beta_1(x_i - \bar{x}) + \epsilon_i$.

b) α equivalent to the mean response ($\alpha = \beta_0 + \beta_1 \bar{x}$) of the previous SLR model and is the intercept of the new model

c) To find the closed form formula of the LSE,

$$
\begin{aligned}
RSS(\beta_1) &= \sum_{i=1}^{n}[y_i - (\alpha + \beta_1(x_i - \bar{x}))]^2 \\
\frac{\mathrm{d}RSS(\beta_1)}{\mathrm{d}\alpha} &= -2\sum_{i=1}^{n}(y_i - (\alpha + \beta_1(x_i - \bar{x}))) \qquad (1) \\
\frac{\mathrm{d}RSS(\beta_1)}{\mathrm{d}\beta_1} &= -2\sum_{i=1}^{n}(y_i - (\alpha + \beta_1(x_i - \bar{x})))(x_i - \bar{x})) \qquad (2)
\end{aligned}
$$

Let Equation (1) = 0

$$
\begin{aligned}
&-2\sum_{i=1}^{n}(y_i - (\hat{\alpha} + \hat{\beta}_1(x_i - \bar{x}))) = 0 \\
&\sum_{i=1}^{n} y_i - \sum_{i=1}^{n}\hat{\alpha}_i - \sum_{i=1}^{n}\hat{\beta}_1 x_i + \sum_{i=1}^{n}\hat{\beta}_1\bar{x} = 0 \\
&n\hat{\alpha}_i = \sum_{i=1}^{n} y_i - \hat{\beta}_1\sum_{i=1}^{n} x_i + n\hat{\beta}_1\bar{x} \\
&\hat{\alpha}_i = \frac{1}{n}\sum_{i=1}^{n} y_i - \hat{\beta}_1\bar{x} + \hat{\beta}_1\bar{x} \\
&\hat{\alpha}_i = \bar{y}
\end{aligned}
$$

Let Equation (2) = 0

$$
\begin{aligned}
&-2\sum_{i=1}^{n}\Big(y_i - (\hat{\alpha} + \hat{\beta}_1(x_i - \bar{x}))(x_i - \bar{x})\Big) = 0 \\
&\sum_{i=1}^{n} y_i(x_i - \bar{x}) - \sum_{i=1}^{n}\hat{\alpha}(x_i - \bar{x}) - \sum_{i=1}^{n}\hat{\beta}_1(x_i - \bar{x})^2 = 0 \\
&\hat{\beta}_1\sum_{i=1}^{n}(x_i - \bar{x})^2 = \sum_{i=1}^{n}(y_i - \hat{\alpha})(x_i - \bar{x})
\end{aligned}
$$

since $\hat{\alpha} = \bar{y}$

$$
\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}
$$

d)

$$\begin{aligned}
Var[\hat{\alpha}] &= Var\left[\frac{1}{n}\sum_{i=1}^{n} y_i\right] \\
&= \frac{1}{n^2} Var\left[\sum_{i=1}^{n} y_i\right]
\end{aligned}$$

Since all y_i's are uncorrelated

$$\begin{aligned}
&= \frac{1}{n^2}\sum_{i=1}^{n} Var[y_i] \\
&= \frac{1}{n^2}\sum_{i=1}^{n} \sigma^2 \\
&= \frac{n\sigma^2}{n^2} \\
&= \frac{\sigma^2}{n}
\end{aligned}$$

Therefore $Var[\hat{\alpha}] = \frac{\sigma^2}{n}$.

We note that $\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}) = \sum_{i=1}^{n}(x_i - \bar{x})y_i$, since

$$\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}) = \sum_{i=1}^{n} y_i(x_i - \bar{x}) - \sum_{i=1}^{n} \bar{y}(x_i - \bar{x})$$

and we notice that

$$\sum_{i=1}^{n}(x_i - \bar{x}) = \sum_{i=1}^{n} x_i - \frac{1}{n} n \sum_{i=1}^{n} x_i = 0$$

To calculate $Var[\hat{\beta}_1]$,

$$\begin{aligned}
Var[\hat{\beta}_1] &= Var\left[\frac{\sum_{i=1}^{n}(x_i - \bar{x})y_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right] \\
&= \frac{1}{\left(\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^2} Var\left[\sum_{i=1}^{n}(x_i - \bar{x})y_i\right] \\
&= \frac{1}{\left(\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^2} \sum_{i=1}^{n} Var\left[(x_i - \bar{x})y_i\right] \\
&= \frac{1}{\left(\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^2} \sum_{i=1}^{n}(x_i - \bar{x})^2 Var\left[y_i\right] \\
&= \frac{1}{\left(\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^2} \sum_{i=1}^{n}(x_i - \bar{x})^2\sigma^2 \\
&= \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}
\end{aligned}$$

Let $S_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2$ and $S_{xy} = \sum_{i=1}^{n}(x_i - \bar{x})y_i$.
To calculate $Cov[\hat{\alpha}, \hat{\beta_1}]$

$$
\begin{aligned}
Cov[\hat{\alpha}, \hat{\beta_1}] &= Cov[\bar{y}, \hat{\beta}_1] \\
&= Cov\left[\bar{y}, \frac{S_{xy}}{S_{xx}}\right] \\
&= \frac{1}{S_{xx}} Cov\left[\bar{y}, S_{xy}\right] \\
&= \frac{1}{S_{xx}} Cov\left[\frac{1}{n}\sum_{i=1}^{n} y_i, S_{xy}\right] \\
&= \frac{1}{nS_{xx}} Cov\left[\sum_{i=1}^{n} y_i, \sum_{j=1}^{n} (x_j - \bar{x})y_j)\right] \\
&= \frac{1}{nS_{xx}} \sum_{i=1}^{n}\sum_{j=1}^{n} (x_j - \bar{x}) Cov[y_i, y_j]
\end{aligned}
$$

When $i \neq j$, $Cov[y_i, y_j] = 0$ since all y_i are uncorrelated with each other, and $Cov[y_i, y_j] = Var[y_i]$ when $i = j$

$$
\begin{aligned}
&= \frac{1}{nS_{xx}} \sum_{i=j}^{n} (x_i - \bar{x})\sigma^2 \\
&= 0
\end{aligned}
$$

e)

```
set.seed(1234567)
x = runif(1000)
eps = rnorm(1000)
y = 5 + 10*x + eps
model <- y~x
RSS <- function(b) c(-2 * sum(y - b[1] - b[2] * (x-mean(x))), -2 * sum((y - b[1] - b[2] * (x-mean(x)))
                                                                            * (x-mean(x))))
#This function gives us the gradient of the RSS
bn <- c(0,0)
gamma <- 0.00001
kmax <- 100000

for (k in 0:kmax) {
  bnp1 <- bn - gamma*RSS(bn)
  if(sum(RSS(bn)^2) <= 0.00001){
    cat("b(alpha, beta): ", bnp1, "-- RSS:", RSS(c(bnp1[1],bnp1[2])),"\n","Iterations:",k,"\n")
    break
  }
  bn <- bnp1
}
```

```
## b(alpha, beta):  10.06591 10.07213 -- RSS: -8.663772e-11 -0.003155449
##  Iterations: 8218
```

```
#This algorithm starts at alpha = 0 and b1 = 0
#I get the l2 norm of score since the RSS function gives the gradient which is the score
#thus the sum of squares of gradient (RSS) must b less than 0.00001
#alpha from minimisation
print('Alpha from minimisation')
```

```
## [1] "Alpha from minimisation"
```

```
bnp1[1]
```

```
## [1] 10.06591
```

```
#Using the closed form formula in c) we find alpha
print('Alpha from the closed form formula in 1c')
```

```
## [1] "Alpha from the closed form formula in 1c"
```

```
mean(y)
```

```
## [1] 10.06591
```

```
#Finding beta
print('Beta from minimisation')
```

```
## [1] "Beta from minimisation"
```

```
bnp1[2]
```

```
## [1] 10.07213
```

```
print('Beta from the closed form formula in 1c')
```

```
## [1] "Beta from the closed form formula in 1c"
```

```
sum(((y-mean(y))*(x-mean(x)))/(sum((x-mean(x))^2)))
```

```
## [1] 10.07215
```

```
#We can see that alpha has the same value as does the beta
#about 8218 iterations were required
#Below is the working to find the gradient function for the RSS
```

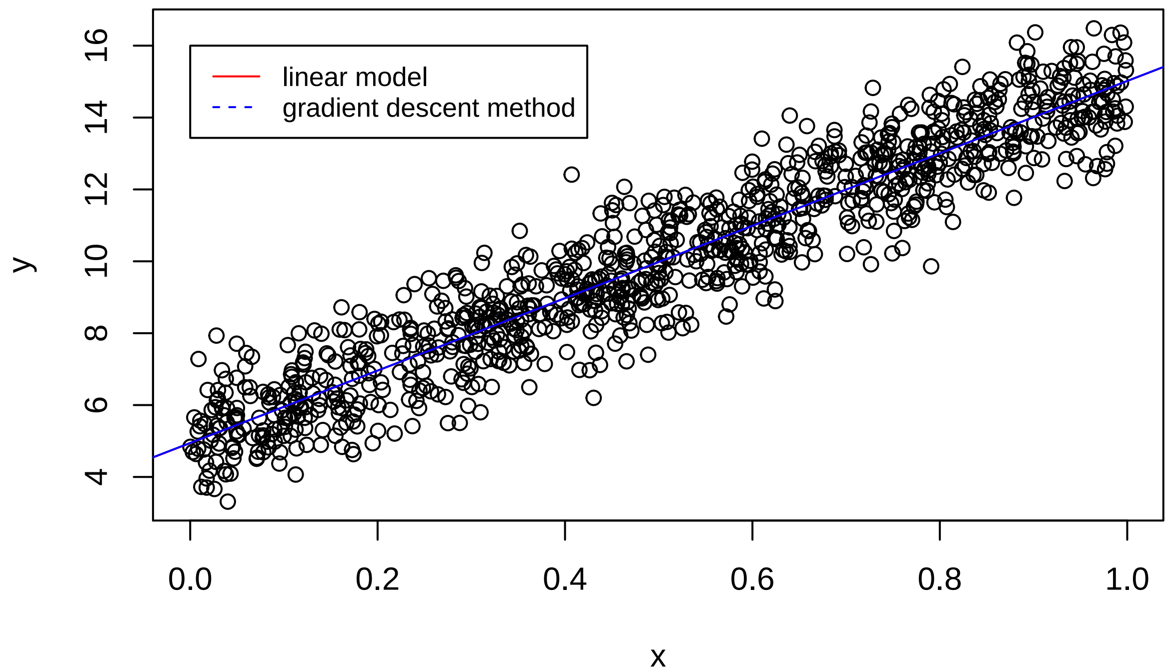
To get the gradient of the RSS we must get the first derivative of:

$$S(\beta_1, \beta_2) = \sum_{i=1}^{n} (y_i - \beta_1 - \beta_2 x_i)^2$$

$$S'(\beta_1, \beta_2) = (-2 \sum_{i=1}^{n} (y_i - \beta_1 - \beta_2 x_i), -2 \sum_{i=1}^{n} x_i (y_i - \beta_1 - \beta_2 x_i))$$

f)

```
plot(x,y)
lm = lm(y~x)
abline(lm, col = "red")
abline(bnp1[1]-bnp1[1]*mean(x),bnp1[2], col = 'blue')
legend(0,16,legend=c("linear model", "gradient descent method"),
       col=c("red", "blue"), lty=1:2, cex=0.8)
```



Note that the lines overlap due to both methods achieving the same result

g)

```
beta = c(0,0)
imax = 100000
RSShess = function(b)cbind(c(2*length(x), 2*sum(x-mean(x))),c(2*sum(x-mean(x)),2*sum((x-mean(x))^2)))
#This function gives us the hessian matrix for the RSS

for (i in 0:imax) {
  beta1 <- beta - solve(RSShess(beta))%*%RSS(beta)
  if(sum(RSS(beta)^2) <= 0.00001){
    cat("beta: ", beta1, "-- RSS:", RSS(c(beta1[1],beta1[2])),"\n","Iterations:",i,"\n")
    break
  }
  beta <- beta1
}
```

```
## beta:  10.06591 10.07215 -- RSS: -1.37159e-12 -1.039592e-13
##  Iterations: 1
```

```
#alpha from NR
print('Alpha from Newton-Raphson')
```

```
## [1] "Alpha from Newton-Raphson"
```

```
beta1[1]
```

```
## [1] 10.06591
```

```r
#Using the closed form formula in c) we find alpha
print('Alpha from closed form formula in 1c')
```

```
## [1] "Alpha from closed form formula in 1c"
```

```r
mean(y)
```

```
## [1] 10.06591
```

```r
#Finding beta
print('Beta from Newton-Raphson')
```

```
## [1] "Beta from Newton-Raphson"
```

```r
beta1[2]
```

```
## [1] 10.07215
```

```r
print('Beta from closed form formula')
```

```
## [1] "Beta from closed form formula"
```

```r
sum(((y-mean(y))*(x-mean(x)))/(sum((x-mean(x))^2)))
```

```
## [1] 10.07215
```

```r
#We can see that values are the same
#1 iterations was required
#It took alot less iterations than the gradient descent method
#This could be due to the fact that the Newton-raphson method
#accounts for the curvature of the loglikelihood and as such should need less iterations
#Below is the working to find the hessian matrix for the RSS
```

To get the Hessian matrix of the RSS we need to further derive the gradient found in Question 1e with respect to α and β_1.

$$
\begin{aligned}
S'(\alpha, \beta_1) &= (-2\sum_{i=1}^{n}(y_i - \alpha - \beta_1(x_i - \bar{x})), -2\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \alpha - \beta_1(x_i - \bar{x}))) \\
\frac{\partial^2 S(\alpha, \beta_1)}{\partial \alpha^2} &= 2\sum_{i=1}^{n} 1 \\
&= 2n \\
\frac{\partial^2 S(\alpha, \beta_1)}{\partial \beta_1^2} &= 2\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x}) \\
&= 2\sum_{i=1}^{n}(x_i - \bar{x})^2 \\
\frac{\partial^2 S(\alpha, \beta_1)}{\partial \alpha \partial \beta_1} &= 2\sum_{i=1}^{n}(x_i - \bar{x}) \\
&= \frac{\partial^2 S(\alpha, \beta_1)}{\partial \beta_1 \partial \alpha}
\end{aligned}
$$

Question 2

Given n independent binary random variables $Y_1 \cdots Y_n$ with

$$P(Y_i = 1) = \pi_i \text{ and } P(Y_i = 0) = 1 - \pi_i$$

The probability function of Y_i is:

$$\pi_i^{Y_i}(1 - \pi_i)^{1-Y_i}$$

where $Y_i = 0$ or 1

a) For a probability function to belong to the exponential family of distributions, it must follow the formula:

$$f(y; \theta, \phi) = K(y, \frac{p}{\phi}) \exp\left(\frac{p}{\phi}\{y\theta - c(\theta)\}\right)$$

For the given probability density function:

$$\begin{aligned}
f(y; \pi) &= \pi_i^y(1 - \pi_i)^{1-y} \\
&= \exp\left(\log \pi_i^y(1 - \pi_i)^{1-y}\right) \\
&= \exp\left(\log \pi_i^y + \log(1 - \pi_i)^{1-y})\right) \\
&= \exp\left(y \log \pi_i + (1 - y) \log(1 - \pi_i)\right) \\
&= \exp\left(y \log(\frac{\pi}{1 - \pi}) + \log(1 - \pi)\right)
\end{aligned}$$

With $p = 1$ and $\phi = 1$, the above equation follows the form of the exponential family of distribution where $K(y, \frac{p}{\phi}) = 1$, $\theta = \log(\frac{\pi}{1-\pi})$ and $c(\theta) = -\log(1 - \pi) = -\log(1 - \frac{e^\theta}{1+e^\theta})$ where $\pi = \frac{e^\theta}{1+e^\theta}$.

b) As seen in 2a, the naturalised parameter is $\theta = \log(\frac{\pi}{1-\pi})$

c) As seen in 2a, the cumulant generator is $c(\theta) = -\log(1 - \frac{e^\theta}{1+e^\theta})$. Since $\mathrm{E}[Y] = c'(\theta)$, $c'(\theta) = -(\frac{e^\theta}{1+e^\theta}) = -(-\pi) = \pi$. Therefore, $\mathrm{E}[Y] = \pi$.

d) Given the link function:

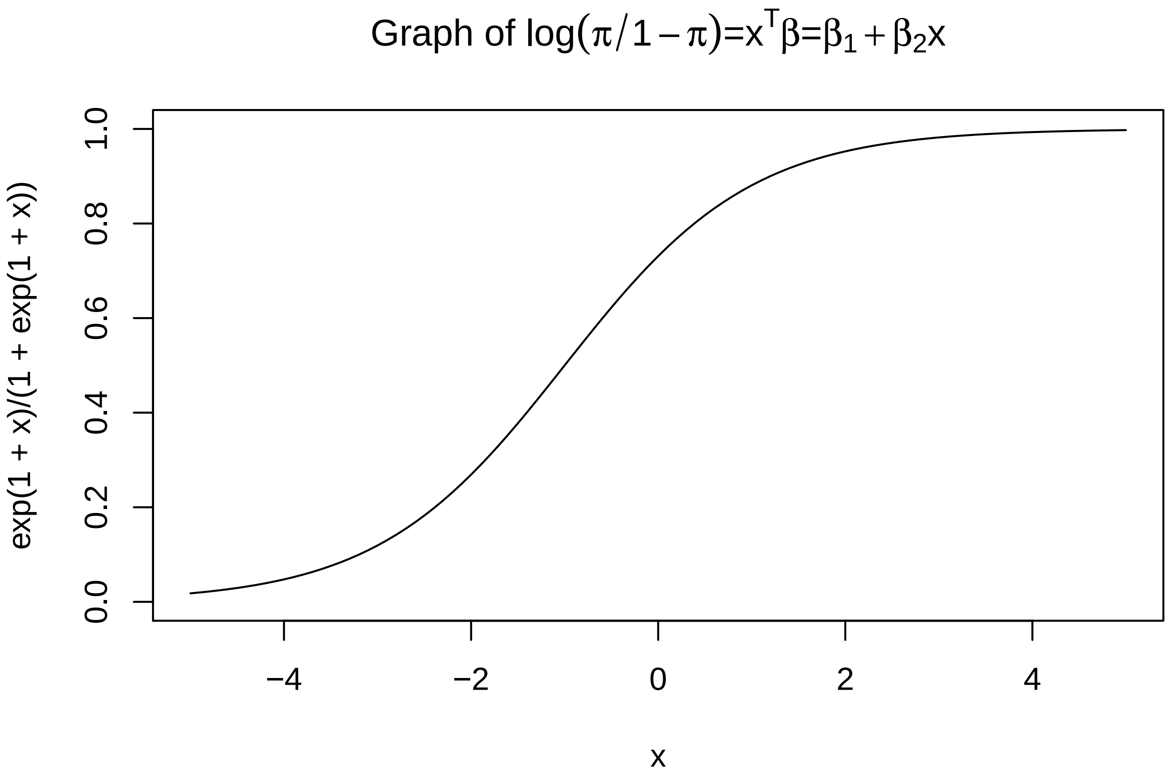
$$g(\pi) = \log(\frac{\pi}{1 - \pi}) = e^{x^T\beta}$$

it can be rearranged in terms of the probability π,

$$\begin{aligned}
e^{x^T\beta} &= \log(\frac{\pi}{1 - \pi}) \\
e^{x^T\beta} - \pi e^{x^T\beta} &= \pi \\
\pi &= \frac{e^{x^T\beta}}{1 + e^{x^T\beta}}
\end{aligned}$$

e)

```
curve(exp(1+x)/(1+exp(1+x)), xlim = c(-5, 5), ylim = c(0, 1),
      main=expression(paste("Graph of ", log(pi/1-pi),'=',x^{T},beta,'=',beta[1]+beta[2],'x'))
      )
```



It shows the log odds of the insecticide working with a given probability.

f) The following probability density function:

$$f(y;\theta) = \frac{1}{\phi}\exp\left(\frac{(y-\theta)}{\phi} - \exp\left[\frac{(y-\theta)}{\phi}\right]\right)$$

is NOT in the exponential family of distributions as it does not follow the form of a probability density function in the exponential family. There is not a way to rearrange the probability density function such that it follows the form of a function in the exponential family, given in 2a.