<u>Question 1</u>

(a) *R code*:
data <- read.csv("Weekly.csv")
data$Direction <- as.factor(data$Direction)
logit_fit <- glm(data = data, Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, family = binomial)
summary(logit_fit)

Output:

```
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6949  -1.2565   0.9913   1.0849   1.4579

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.26686    0.08593   3.106   0.0019 **
Lag1        -0.04127    0.02641  -1.563   0.1181
Lag2         0.05844    0.02686   2.175   0.0296 *
Lag3        -0.01606    0.02666  -0.602   0.5469
Lag4        -0.02779    0.02646  -1.050   0.2937
Lag5        -0.01447    0.02638  -0.549   0.5833
Volume      -0.02274    0.03690  -0.616   0.5377
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1496.2  on 1088  degrees of freedom
Residual deviance: 1486.4  on 1082  degrees of freedom
AIC: 1500.4

Number of Fisher Scoring iterations: 4
```

(b) Based on the above results, we can only reject the null for for Lag2, with Lag2 positively correlating with likelihood of Direction being “Up”. (Note, the intercept is also statistically significant and nonzero, and non-negative. That is, without any predictors, Direction is more likely to be “Up” than “Down”.)

(c) *R Code:*

```
Direction_hat_logit <- rep("Down", nrow(data))
Direction_hat_logit_prob <- predict(logit_fit, type = "response")
Direction_hat_logit[which(Direction_hat_logit_prob >= 0.5)] <- "Up"
Direction_hat_logit <- as.factor(Direction_hat_logit)
table(Direction_hat_logit,data$Direction)
mean(Direction_hat_logit == data$Direction)
```

Output:

```
> table(Direction_hat_logit,data$Direction)

Direction_hat_logit Down  Up
               Down   54  48
               Up    430 557
> mean(Direction_hat_logit == data$Direction)
[1] 0.5610652
```

(d) *R Code:*

```
lda_fit <- MASS::lda(data = data, Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume)
lda_pred <- predict(lda_fit, data)
Direction_hat_lda <- lda_pred$class
table(Direction_hat_lda,data$Direction)
mean(Direction_hat_lda == data$Direction)
```

Output:

```
> table(Direction_hat_lda,data$Direction)

Direction_hat_lda Down  Up
             Down   52  46
             Up    432 559
> mean(Direction_hat_lda == data$Direction)
[1] 0.5610652
```

Both LDA and Logit perform equally in terms of overall accuracy. Though Logit is *slightly* more sensitive, while LDA is *slightly* more specific.

Can be confirm by outputs of the following code:

```
install.packages("caret")
library(caret)
confusionMatrix(data = as.factor(Direction_hat_logit), reference = data$Direction)
confusionMatrix(data = as.factor(Direction_hat_lda), reference = data$Direction)
```

Question 2

(a)

2a.

$$L(\beta|y) = \prod_{i=1}^{n} p(x_i;\beta)^{y_i}\left(1-p(x_i;\beta)\right)^{1-y_i}$$

log both sides

$$\begin{aligned}
l(\beta|y) &= \sum_{i=1}^{n} y_i \log p(x_i;\beta) + (1-y_i)\log\left(1-p(x_i;\beta)\right) \\
&= \sum_{i=1}^{n} \log\left(1-p(x_i;\beta)\right) + \sum_{i=1}^{n} y_i \log\left(\frac{p(x_i;\beta)}{1-p(x_i;\beta)}\right) \\
&= \sum_{i=1}^{n} \log\left(1-p(x_i;\beta)\right) + \sum_{i=1}^{n} y_i(\beta_0+\beta_1 x_i) \\
&= -\sum_{i=1}^{n} \log\left(1+\exp(\beta_0+\beta_1 x_i)\right) + \sum_{i=1}^{n} y_i(\beta_0+\beta_1 x_i)
\end{aligned}$$

(b)

2b.

$$\begin{aligned}
\frac{\partial l}{\partial \beta_0} &= -\sum_{i=1}^{n}\left(1\cdot e^{\beta_0+\beta_1 x_i}\cdot\frac{1}{1+e^{\beta_0+\beta_1 x_i}}\right) + \sum_{i=1}^{n} y_i \\
&= \sum_{i=1}^{n}\left(y_i - \frac{\exp(\beta_0+\beta_1 x_i)}{1+\exp(\beta_0+\beta_1 x_i)}\right)
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial l}{\partial \beta_1} &= -\sum_{i=1}^{n}\left(x_i\cdot e^{\beta_0+\beta_1 x_i}\cdot\frac{1}{1+e^{\beta_0+\beta_1 x_i}}\right) + \sum_{i=1}^{n} x_i y_i \\
&= \sum_{i=1}^{n}\left(x_i y_i - \frac{x_i\cdot\exp(\beta_0+\beta_1 x_i)}{1+\exp(\beta_0+\beta_1 x_i)}\right)
\end{aligned}$$

so

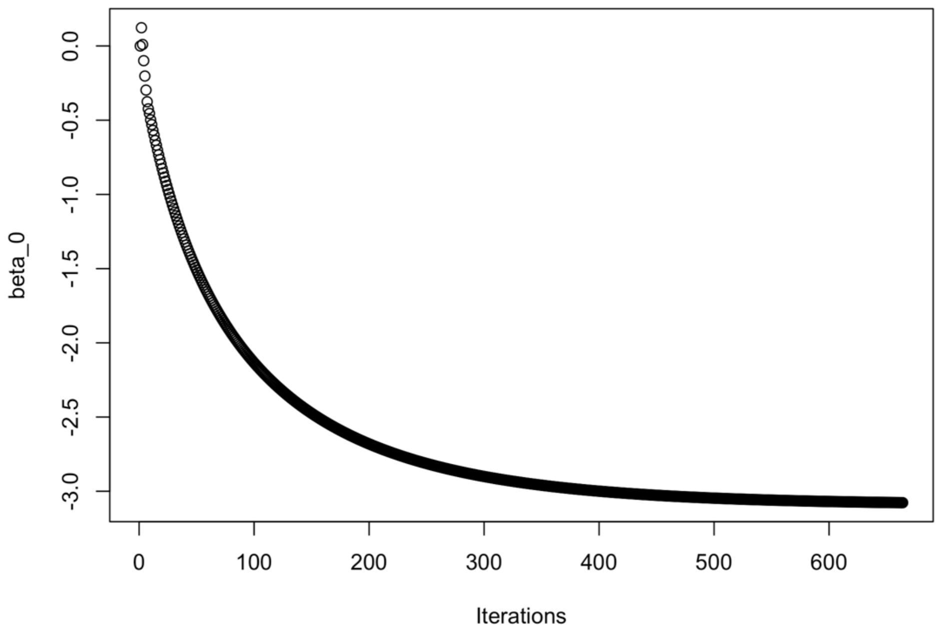
$$\nabla l(\beta|y) = \begin{pmatrix} \sum_{i=1}^{n}\left(y_i - \frac{\exp(\beta_0+\beta_1 x_i)}{1+\exp(\beta_0+\beta_1 x_i)}\right) \\ \sum_{i=1}^{n}\left(x_i y_i - \frac{x_i\cdot\exp(\beta_0+\beta_1 x_i)}{1+\exp(\beta_0+\beta_1 x_i)}\right) \end{pmatrix}$$

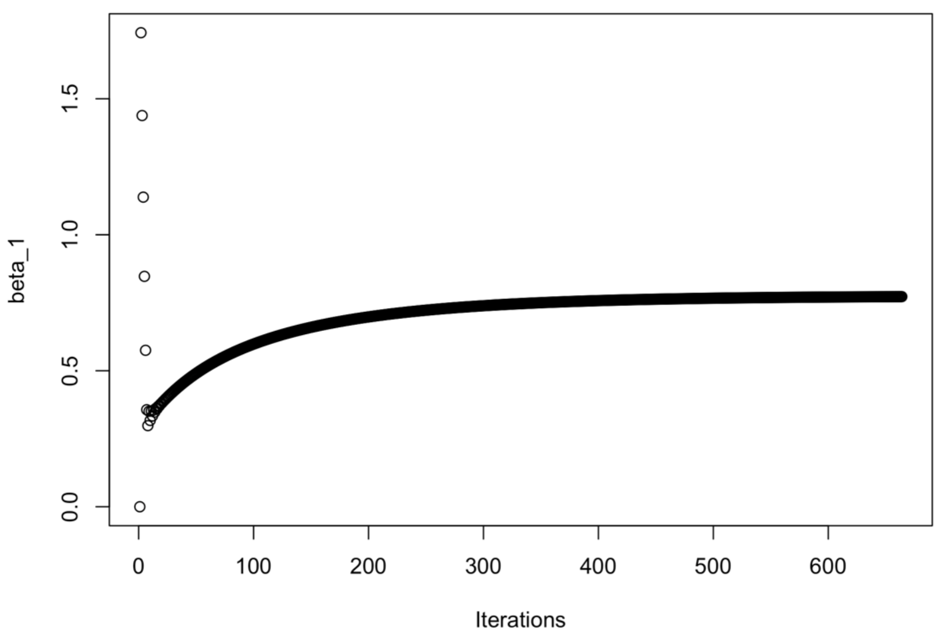
(c) *R Code:*

```
#clear environment
rm(list=ls())

#read data, set column names, code outcomes as factors
logit_data <- read.csv("logit_data.csv", header = FALSE)
colnames(logit_data) <- c("A","study_hours")

#assignment of initial variables
y <- logit_data$A
x <- logit_data$study_hours
alpha <- 0.001
b_last <- matrix(ncol=2,nrow=1,data=0)
colnames(b_last) <- c("b0","b1")
b <- matrix(ncol=2,nrow=1,data=0)
colnames(b_last) <- c("b0","b1")
err <- 100
eps <- 10^-4
b_history <- matrix(ncol=2, data = 0)
colnames(b_history) <- c("b0","b1")
grad <- matrix(ncol=1,nrow=2)

#calculation of grad descent
while(err > eps) {
  grad[1,] <- sum(y-exp(b_last[,1]+b_last[,2]*x)/(1+exp(b_last[,1]+b_last[,2]*x)))
  grad[2,] <- sum(x*y-x*exp(b_last[,1]+b_last[,2]*x)/(1+exp(b_last[,1]+b_last[,2]*x)))
  b[,1] = b_last[,1] + alpha*grad[1,]
  b[,2] = b_last[,2] + alpha*grad[2,]
  err_temp <- abs(b - b_last)
  err <- sqrt(err_temp[,1]^2+err_temp[,2]^2)
  b_last <- b
  b_history <- rbind(b_history, b_last)
}

#print final betas
cat("b_0 =",b_last[,1])
cat("b_1 =",b_last[,2])

#plot betas
plot(b_history[,1], xlab = "Iterations", ylab = "beta_0")
plot(b_history[,2], xlab = "Iterations", ylab = "beta_1")
```

Output:

```
> cat("b_0 =",b_last[,1])
b_0 = -3.076811
> cat("b_1 =",b_last[,2])
b_1 = 0.772469
```





(d) *R Code:*

```
b_0 <- b_last[,1]
b_1 <- b_last[,2]
x_input <- 5
prob_a <- exp(b_0+b_1*x_input)/(1+exp(b_0+b_1*x_input))
prob_not_a <- 1-prob_a
prob_not_a
```

Output:

The probability of not getting an A with 5 study hours per week is 0.3131283.