```
setwd("~/Workspace/College/DIT/MATH9952/Data")

# read in the data
retention = read.csv("retention.csv", header = T)

# remove unnecessary columns
retention$X           = NULL
retention$lc_points.1 = NULL

# convert columns to factore
retention$mathgrd     = as.factor(retention$mathgrd)
retention$address     = as.factor(retention$address)

# remove rows where NULLs or NAs are present
retention             = retention[complete.cases(retention),]

# have a look at the data
head(retention)

attach(retention)

# list the column names
colnames(retention)




# Question 1

# fit a model with all interractions
fit1 = glm(passed ~ .*., family = binomial(link = "logit"), data = retention)
summary(fit1)

# prune unnecessary predictors
step(fit1, scope = list(lower = ~ 1, upper = ~ .*.), direction = "backward", trace = 1)

# fit the final model
fitf = glm(passed ~ gender + mathgrd + CAO_choice + lc_points + gender:CAO_choice, family = binomial(link =
"logit"), data = retention)
summary(fitf)




# Question 3

nd = data.frame(gender = 1, lc_points = 300, mathgrd = "50-60", CAO_choice = "3")
p  = predict(fitf, newdata = nd, se = T)

prob = exp(p$fit) / (1 + exp(p$fit))
ciu  = exp(p$fit + 1.96 * p$se.fit) / (1 + exp(p$fit + 1.96 * p$se.fit))
cil  = exp(p$fit - 1.96 * p$se.fit) / (1 + exp(p$fit - 1.96 * p$se.fit))

data.frame(prob = prob, upperCI = ciu, lowerCI = cil)
#        prob   upperCI   lowerCI
# 1 0.6629496 0.8398823 0.4244774




# Question 4

fitz = glm(passed ~ gender + lc_points, family = binomial(link = "logit"), data = retention)
summary(fitz)

# NR Method

x1     = gender
x2     = lc_points
y      = passed

beta0  = 1
beta1  = 0
beta2  = 0
beta   = matrix(c(beta0, beta1, beta2), nrow = 3)
```

```
# iterations start...

eta     = (beta[1, 1] + beta[2, 1] * x1 + beta[3, 1] * x2)

score1  = sum(y        - ((1 * exp(eta)) / (1 + exp(eta))))
score2  = sum((y * x1) - ((1 * x1 * exp(eta)) / (1 + exp(eta))))
score3  = sum((y * x2) - ((1 * x2 * exp(eta)) / (1 + exp(eta))))

h11     = sum( (1 * exp(eta)^2)           / (1 + exp(eta))^2 - ((1 * exp(eta)) / (1 + exp(eta))))
h12     = sum( (1 * x1 * exp(eta)^2)      / (1 + exp(eta))^2 - ((1 * x1 * exp(eta)) / (1 + exp(eta))))
h13     = sum( (1 * x2 * exp(eta)^2)      / (1 + exp(eta))^2 - ((1 * x2 * exp(eta)) / (1 + exp(eta))))
h22     = sum( (1 * x1^2 * exp(eta)^2)    / (1 + exp(eta))^2 - ((1 * x1^2 * exp(eta)) / (1 + exp(eta))))
h23     = sum( (1 * x1 * x2 * exp(eta)^2) / (1 + exp(eta))^2 - ((1 * x1 * x2 * exp(eta)) / (1 + exp(eta))))
h33     = sum( (1 * x2^2 * exp(eta)^2)    / (1 + exp(eta))^2 - ((1 * x2^2 * exp(eta)) / (1 + exp(eta))))

u       = matrix(c(score1, score2, score3), nrow = 3)
h       = matrix(c(h11, h12, h13, h12, h22, h23, h13, h23, h33), nrow = 3, byrow = T)

betanew = beta - solve(h) %*% u
beta    = betanew

result  = data.frame(beta = beta, score = u, hessian = h)
result

# iterations end...
```

```
# Appendix

# fit a model without interractions - simpler!

# fit1  = glm(passed ~ ., family = binomial(link = "logit"), data = retention)
# formula(fit1)
# summary(fit1)

# use the step function to help you find the predictors to drop...

# step(fit1, scope = list(lower = ~ 1, upper = ~ .), direction = "backward", trace = 1)

# or use the drop1 function to manually drop predictors...

# drop1(fit1, test = 'LRT')
# fit2 = update(fit1, ~. - CAO_choice)
# summary(fit2)
#
# drop1(fit2, test = 'LRT')
# fit3 = update(fit2, ~. - address)
# summary(fit3)
#
# drop1(fit3, test = 'LRT')
```