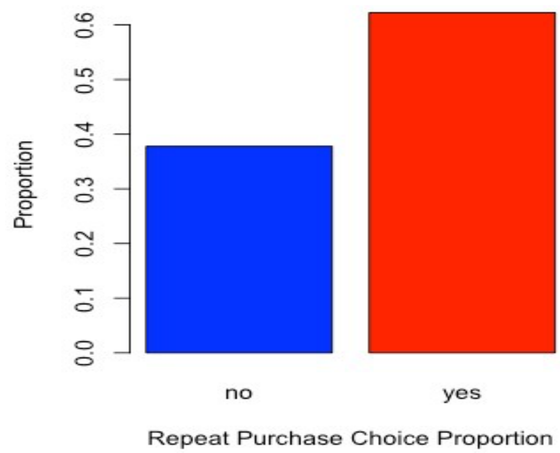
Name: Nithineshwar Songala ID: 16344141
Assignment – Discrete Choice Models – code

```
# Demonstartion of discrete choice models
#
# @author Nithineshwar Songala @date 02/21/2025

#set the working directory
setwd("/Users/lavanyadhaipully/Desktop/Nithin Assign")

#load the dataset
dataset <- read.csv("dataset_online_store.csv")
View(dataset)

#summary of the dataset summary(dataset)
table(dataset$isMultiBrandUser)
table(dataset$promotiontype)

#display a table of proportionsfor categorical type in the dataset (frequency_repeat_buyer
<- table(dataset$isRepeatBuyer))
(proportion_repeat_buyer <- frequency_repeat_buyer/nrow(dataset))
(proportion_repeat_buyer <- frequency_repeat_buyer/length(dataset$isRepeatBuyer))

#Bar Chart of frequency/proportion table
barplot(frequency_repeat_buyer) barplot(proportion_repeat_buyer)
barplot(proportion_repeat_buyer, xlab = "Repeat Purchase Choice Proportion",
ylab = "Proportion", col = c("blue", "red"))
barplot(proportion_repeat_buyer, xlab = "Repeat Purchase Choice Proportion",
ylab = "Proportion", col = c("blue", "red"), horiz = TRUE)


#create indicator variables for all two category variables in dataset dataset$isRepeatBuyer_yes_i
<- 0
dataset$isRepeatBuyer_yes_i[which(dataset$isRepeatBuyer == "yes")] <- 1

dataset$isMultiBrandUser_yes_i <- 0
dataset$isMultiBrandUser_yes_i[which(dataset$isMultiBrandUser == "yes")] <- 1

dataset$hasUserFriend_yes_i <- 0
dataset$hasUserFriend_yes_i[which(dataset$hasUserFriend == "yes")] <- 1

#estimate a logit model of repeat purchase as a function of isMultiBrandUser
#log(p/(1-p)) = b0 + b1 * isMultiBrnadUser_yes_i

logit_model_1 <- glm(isRepeatBuyer_yes_i ~ isMultiBrandUser_yes_i,
family = binomial(link = "logit"), data = dataset) summary(logit_model_1)
#log(p/(1-p)) = 1.11 - 1.07 * isMultiBrnadUser_yes_i

#marginal and multiplicative effects
marginal_effects_1 <- coef(logit_model_1) #doesn't print (marginal_effects_1
<- coef(logit_model_1)) # to print use()
(multi_effects_1 <- exp(marginal_effects_1))
(multi_effects_percent_1 <- (multi_effects_1 - 1)*100)

#estimate a logit model where repeat purchase is a function of all variables
# except promotion type
logit_model_2 <- glm(isRepeatBuyer_yes_i ~ isMultiBrandUser_yes_i +
hasUserFriend_yes_i + cookingRating + recipeRating,                     family =
binomial(link = "logit"), data = dataset) summary(logit_model_2)

#marginal and multiplicative effects
(marginal_effects_2 <- coef(logit_model_2))
```

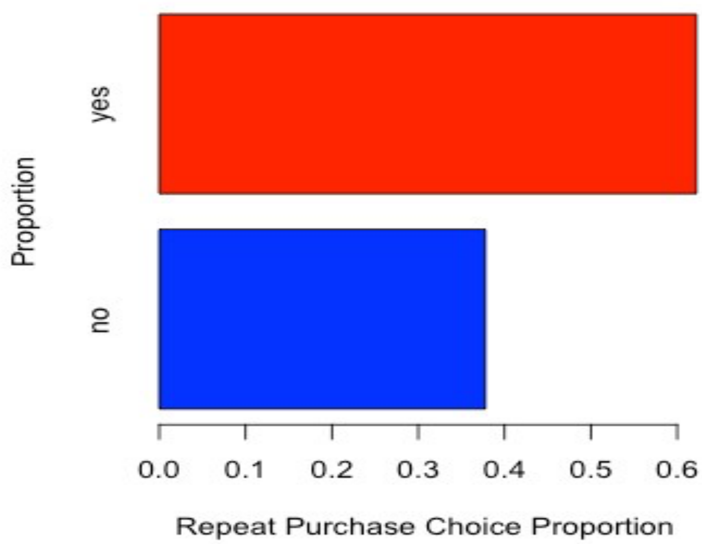
```
(multi_effects_2 <- exp(marginal_effects_2))
(multi_effects_percent_2 <- (multi_effects_2 - 1)*100)
cbind(marginal_effects_2, multi_effects_2, multi_effects_percent_2)
#making prediction to compute predictive accuracy
dataset$probability_predicted_2 <- predict(logit_model_2, dataset[ , c("isMultiBrandUser_yes_i",
                                                     "hasUserFriend_yes_i", "cookingRating",
"recipeRating")], type = "response") dataset$choice_predicted_2 <- 0
dataset$choice_predicted_2[which(dataset$probability_predicted_2 > 0.5)] <- 1 (confusion_matrix_2
<- table(dataset$isRepeatBuyer, dataset$choice_predicted_2))


(accuracy_2 <-sum(diag(confusion_matrix_2))/sum(confusion_matrix_2))


#create Indicators for promotion type dataset$promotion_free_i
<-0
dataset$promotion_free_i[which(dataset$promotionType == "free")] <- 1

dataset$promotion_discount_i <-0
dataset$promotion_discount_i[which(dataset$promotionType == "discount")] <- 1

#estimate a logit model where repeat purchase is a function of all variables
# including promotion type
logit_model_3 <- glm(isRepeatBuyer_yes_i ~ isMultiBrandUser_yes_i +
hasUserFriend_yes_i + cookingRating + recipeRating +
promotion_free_i + promotion_discount_i,                       family =
binomial(link = "logit"), data = dataset) summary(logit_model_3)

#marginal and multiplicative effects for logit_model_3
(marginal_effects_3 <- coef(logit_model_3))
(multi_effects_3 <- exp(marginal_effects_3))
(multi_effects_percent_3 <- (multi_effects_3 - 1)*100)
cbind(marginal_effects_3, multi_effects_3, multi_effects_percent_3)

#making prediction to compute predictive accuracy for logit_model_3
dataset$probability_predicted_3 <- predict(logit_model_3, dataset[ , c("isMultiBrandUser_yes_i",
                                                                       "hasUserFriend_yes_i",
                                                                       "cookingRating",
                                                                       "recipeRating",
                                                                       "promotion_free_i",
"promotion_discount_i")], type = "response") dataset$choice_predicted_3 <- 0
dataset$choice_predicted_3[which(dataset$probability_predicted_3 > 0.5)] <- 1 (confusion_matrix_3
<- table(dataset$isRepeatBuyer, dataset$choice_predicted_3))

(accuracy_3 <-sum(diag(confusion_matrix_3))/sum(confusion_matrix_3))
# making a prediction for a new customer ( multibrand user, has no user friend,
# received a discount type promotion)
new_customer <- data.frame(isMultiBrandUser_yes_i =1, hasUserFriend_yes_i =0,
cookingRating = mean(dataset$cookingRating),                        recipeRating
= mean(dataset$recipeRating),                         promotion_free_i = 0,
promotion_discount_i = 1)

predict(logit_model_3, new_customer, type = "response")

dataset[1:10, c("isRepeatBuyer", "probability_predicted_2", "choice_predicted_2",
"probability_predicted_3", "choice_predicted_3")]
```

Console Output

```
> # Demonstartion of discrete choice models
> #
> # @author Nithineshwar Songala @date 02/21/2025
> #set the working directory
> setwd("/Users/lavanyadhaipully/Desktop/Nithin Assign")
>
> #load the dataset
> dataset <- read.csv("dataset_online_store.csv")
> View(dataset)
>
> #summary of the dataset >
summary(dataset)
 cookingRating     recipeRating    isMultiBrandUser   hasUserFriend      promotionType
isRepeatBuyer
 Min.   : 0.000   Min.   : 0.000   Length:450         Length:450         Length:450
Length:450
 1st Qu.: 4.000   1st Qu.: 4.300   Class :character   Class :character   Class :character   Class
:character
 Median : 4.900   Median : 5.400   Mode  :character   Mode  :character   Mode  :character   Mode
:character
 Mean   : 5.024   Mean   : 5.384
 3rd Qu.: 6.200   3rd Qu.: 6.400
 Max.   :10.000   Max.   :10.000
> table(dataset$isMultiBrandUser)
  no
yes
207 243
> table(dataset$promotiontype)
< table of extent 0 >
> #display a table of proportionsfor categorical type in the dataset
> (frequency_repeat_buyer <- table(dataset$isRepeatBuyer))
  no
yes
170 280
> (proportion_repeat_buyer <- frequency_repeat_buyer/nrow(dataset))

       no       yes  0.3777778
0.6222222
> (proportion_repeat_buyer <- frequency_repeat_buyer/length(dataset$isRepeatBuyer))

       no       yes
0.3777778 0.6222222
> #Bar Chart of frequency/proportion table
> barplot(frequency_repeat_buyer)
> barplot(proportion_repeat_buyer)
> barplot(proportion_repeat_buyer, xlab = "Repeat Purchase Choice Proportion",
+         ylab = "Proportion", col = c("blue", "red"))
```




```
> barplot(proportion_repeat_buyer, xlab = "Repeat Purchase Choice Proportion",
+         ylab = "Proportion", col = c("blue", "red"), horiz = TRUE) >
```




```
> ?barplot
#create indicator variables for all two category variables in dataset
> dataset$isRepeatBuyer_yes_i <- 0
> dataset$isRepeatBuyer_yes_i[which(dataset$isRepeatBuyer == "yes")] <- 1
>
> dataset$isMultiBrandUser_yes_i <- 0
> dataset$isMultiBrandUser_yes_i[which(dataset$isMultiBrandUser == "yes")] <- 1
>
> dataset$hasUserFriend_yes_i <- 0
> dataset$hasUserFriend_yes_i[which(dataset$hasUserFriend == "yes")] <- 1
>
> #estimate a logit model of repeat purchase as a function of isMultiBrandUser
> #log(p/(1-p)) = b0 + b1 * isMultiBrnadUser_yes_i
```

```
> logit_model_1 <- glm(isRepeatBuyer_yes_i ~ isMultiBrandUser_yes_i,
+                       family = binomial(link = "logit"), data = dataset)
> summary(logit_model_1)
Call:
glm(formula = isRepeatBuyer_yes_i ~ isMultiBrandUser_yes_i, family = binomial(link = "logit"),
data = dataset)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6739  -1.1949   0.7521   1.1600   1.1600

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)     (Intercept)
1.1180     0.1613   6.931 4.17e-12 *** isMultiBrandUser_yes_i  -1.0769
0.2061  -5.224 1.75e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 596.67  on 449  degrees of freedom
Residual deviance: 567.91  on 448  degrees of freedom
AIC: 571.91

Number of Fisher Scoring iterations: 4

> #log(p/(1-p)) = 1.11 - 1.07 * isMultiBrnadUser_yes_i
>
> #marginal and multiplicative effects
> marginal_effects_1 <- coef(logit_model_1) #doesn't print
> (marginal_effects_1 <- coef(logit_model_1)) # to print use()
           (Intercept) isMultiBrandUser_yes_i
              1.118030              -1.076872
> (multi_effects_1 <- exp(marginal_effects_1))
           (Intercept) isMultiBrandUser_yes_i
             3.0588235              0.3406593
> (multi_effects_percent_1 <- (multi_effects_1 - 1)*100)
           (Intercept) isMultiBrandUser_yes_i
             205.88235              -65.93407
> #estimate a logit model where repeat purchase is a function of all variables
> # except promotion type
> logit_model_2 <- glm(isRepeatBuyer_yes_i ~ isMultiBrandUser_yes_i +
+                         hasUserFriend_yes_i + cookingRating + recipeRating,
+                       family = binomial(link = "logit"), data = dataset)
> summary(logit_model_2)
Call:
glm(formula = isRepeatBuyer_yes_i ~ isMultiBrandUser_yes_i +
    hasUserFriend_yes_i + cookingRating + recipeRating, family = binomial(link = "logit"),
data = dataset)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3872  -0.8280   0.3811   0.7581   2.1664

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)     (Intercept)
-5.31436    0.69767  -7.617 2.59e-14 *** isMultiBrandUser_yes_i -1.55465
0.25496  -6.098 1.08e-09 *** hasUserFriend_yes_i     0.52512    0.26426
1.987   0.0469 *   cookingRating           0.67881    0.09010   7.534
4.93e-14 *** recipeRating            0.61725    0.08707   7.089 1.35e-12
***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 596.67  on 449  degrees of freedom
Residual deviance: 435.37  on 445  degrees of freedom
AIC: 445.37
```

```
Number of Fisher Scoring iterations: 5

>
> #marginal and multiplicative effects
> (marginal_effects_2 <- coef(logit_model_2))
           (Intercept) isMultiBrandUser_yes_i    hasUserFriend_yes_i          cookingRating
recipeRating
            -5.3143561            -1.5546514              0.5251214              0.6788057
0.6172491
> (multi_effects_2 <- exp(marginal_effects_2))
           (Intercept) isMultiBrandUser_yes_i    hasUserFriend_yes_i          cookingRating
recipeRating
           0.004920446           0.211263021            1.690664040            1.971521690
1.853821308
> (multi_effects_percent_2 <- (multi_effects_2 - 1)*100)
           (Intercept) isMultiBrandUser_yes_i    hasUserFriend_yes_i          cookingRating
recipeRating
             -99.50796             -78.87370               69.06640               97.15217
85.38213
> cbind(marginal_effects_2, multi_effects_2, multi_effects_percent_2)
marginal_effects_2 multi_effects_2 multi_effects_percent_2 (Intercept)
-5.3143561     0.004920446              -99.50796 isMultiBrandUser_yes_i         -
1.5546514     0.211263021              -78.87370 hasUserFriend_yes_i
0.5251214     1.690664040               69.06640 cookingRating
0.6788057     1.971521690               97.15217 recipeRating
0.6172491     1.853821308               85.38213
> #making prediction to compute predictive accuracy
> dataset$probability_predicted_2 <- predict(logit_model_2, dataset[ ,
c("isMultiBrandUser_yes_i",
+                                                    "hasUserFriend_yes_i", "cookingRating",
+                                                    "recipeRating")], type = "response")
> dataset$choice_predicted_2 <- 0
> dataset$choice_predicted_2[which(dataset$probability_predicted_2 > 0.5)] <- 1
> (confusion_matrix_2 <- table(dataset$isRepeatBuyer, dataset$choice_predicted_2))

0   1
  no  100  70
yes  45 235
>
>table(dataset$isRepeatBuyer)
  no
yes
170 280
> (accuracy_2 <-sum(diag(confusion_matrix_2))/sum(confusion_matrix_2))
[1] 0.7444444
> #create Indicators for promotion type
> dataset$promotion_free_i <-0
> dataset$promotion_free_i[which(dataset$promotionType == "free")] <- 1 >
> dataset$promotion_discount_i <-0
> dataset$promotion_discount_i[which(dataset$promotionType == "discount")] <- 1
>
> #estimate a logit model where repeat purchase is a function of all variables
> # including promotion type
> logit_model_3 <- glm(isRepeatBuyer_yes_i ~ isMultiBrandUser_yes_i +
+                         hasUserFriend_yes_i + cookingRating + recipeRating +
+                         promotion_free_i + promotion_discount_i,
+                       family = binomial(link = "logit"), data = dataset)
> summary(logit_model_3)  Call: glm(formula =
isRepeatBuyer_yes_i ~ isMultiBrandUser_yes_i +
    hasUserFriend_yes_i + cookingRating + recipeRating + promotion_free_i +
promotion_discount_i, family = binomial(link = "logit"),      data =
dataset)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
```

```
-2.4985  -0.7810   0.3363   0.7511   2.1585
Coefficients:
                       Estimate Std. Error z value Pr(>|z|)     (Intercept)
-5.79307    0.72943  -7.942 1.99e-15 *** isMultiBrandUser_yes_i -1.54237
0.25923  -5.950 2.68e-09 *** hasUserFriend_yes_i     0.51746    0.26625
1.944 0.051955 .   cookingRating           0.67770    0.09067   7.474
7.77e-14 *** recipeRating            0.61957    0.08790   7.049 1.80e-12
*** promotion_free_i        0.44228    0.28724   1.540 0.123619
promotion_discount_i    1.04890    0.29819   3.518 0.000436 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 596.67  on 449  degrees of freedom
Residual deviance: 422.49  on 443  degrees of freedom
AIC: 436.49

Number of Fisher Scoring iterations: 5
> #marginal and multiplicative effects for logit_model_3
> (marginal_effects_3 <- coef(logit_model_3))
           (Intercept) isMultiBrandUser_yes_i    hasUserFriend_yes_i          cookingRating
recipeRating       promotion_free_i
            -5.7930664             -1.5423718              0.5174560              0.6777001
0.6195650             0.4422792
promotion_discount_i
1.0489038
> (multi_effects_3 <- exp(marginal_effects_3))
           (Intercept) isMultiBrandUser_yes_i    hasUserFriend_yes_i          cookingRating
recipeRating       promotion_free_i
            0.00304862             0.21387324             1.67775403             1.96934325
1.85811965            1.55625021    promotion_discount_i             2.85452033
> (multi_effects_percent_3 <- (multi_effects_3 - 1)*100)
           (Intercept) isMultiBrandUser_yes_i    hasUserFriend_yes_i          cookingRating
recipeRating       promotion_free_i
             -99.69514              -78.61268               67.77540               96.93433
85.81197              55.62502
promotion_discount_i
185.45203
> cbind(marginal_effects_3, multi_effects_3, multi_effects_percent_3)
marginal_effects_3 multi_effects_3 multi_effects_percent_3 (Intercept)
-5.7930664      0.00304862              -99.69514 isMultiBrandUser_yes_i         -
1.5423718      0.21387324             -78.61268 hasUserFriend_yes_i
0.5174560      1.67775403              67.77540 cookingRating
0.6777001      1.96934325              96.93433 recipeRating
0.6195650      1.85811965              85.81197 promotion_free_i
0.4422792      1.55625021              55.62502 promotion_discount_i
1.0489038      2.85452033             185.45203 >
> #making prediction to compute predictive accuracy for logit_model_3
> dataset$probability_predicted_3 <- predict(logit_model_3, dataset[ ,
c("isMultiBrandUser_yes_i",
+                                                                         "hasUserFriend_yes_i",
"cookingRating",
+                                                                         "recipeRating",
"promotion_free_i",
+
"promotion_discount_i")], type = "response")
> dataset$choice_predicted_3 <- 0
> dataset$choice_predicted_3[which(dataset$probability_predicted_3 > 0.5)] <- 1
> (confusion_matrix_3 <- table(dataset$isRepeatBuyer, dataset$choice_predicted_3))
              0    1
no  108  62   yes  43
237               >
confusion_matrix_2

0  1  no  100
70    yes  45
235
```

```
> (accuracy_3 <-sum(diag(confusion_matrix_3))/sum(confusion_matrix_3))
[1] 0.7666667
> accuracy_2
[1] 0.7444444
> # making a prediction for a new customer ( multibrand user, has no user friend,
> # received a discount type promotion)
> new_customer <- data.frame(isMultiBrandUser_yes_i =1, hasUserFriend_yes_i =0,
+                             cookingRating = mean(dataset$cookingRating),
+                             recipeRating = mean(dataset$recipeRating),
+                             promotion_free_i = 0, promotion_discount_i = 1)
> new_customer
  isMultiBrandUser_yes_i hasUserFriend_yes_i cookingRating recipeRating promotion_free_i
promotion_discount_i
1                      1                   0         5.024        5.384                0
1
>
> predict(logit_model_3, new_customer, type = "response")
        1
0.6115825
>
> dataset[1:10, c("isRepeatBuyer", "probability_predicted_2", "choice_predicted_2",
+                  "probability_predicted_3", "choice_predicted_3")]
   isRepeatBuyer probability_predicted_2 choice_predicted_2 probability_predicted_3
choice_predicted_3
1             no              0.13723202                  0              0.13419770
0
2             yes              0.86788305                  1              0.92065663
1
3             yes              0.98926747                  1              0.98307951
1
4             yes              0.88885944                  1              0.83248426
1
5             yes              0.70740134                  1              0.81092226
1
6             yes              0.82570910                  1              0.82388386
1
7             no              0.40546469                  0              0.30003728
0
8             yes              0.41886005                  0              0.41207314
0
9             no              0.02555554                  0              0.04516668
0
10            no              0.36029668                  0              0.50252509
1
>
```