Exercise 1

```
data("margarine")
# Average and dispersion in product characteristics.
datchoice=margarine$choicePrice
dataset=datchoice %>% summarise(apply(datchoice[,3:12], 2, mean), colSds(as.matrix(datchoice[,3:12
])))

dataset=dataset %>%
 mutate_if(is.numeric, round, digits=3)
dataset$min=t(datchoice %>% summarise_at(3:12,min))
dataset$max=t(datchoice %>% summarise_at(3:12,max))
rownames(dataset)=c("PPk_Stk", "PBB_Stk", "PFl_Stk",  "PHse_Stk", "PGen_Stk", "PImp_Stk", "PSS_Tu
b",  "PPk_Tub",  "PFl_Tub",  "PHse_Tub")
colnames(dataset)=c("Mean", "Standard_Deviation","Min","Max")
dataset
```

```
##           Mean Standard_Deviation  Min  Max
## PPk_Stk  0.518              0.151 0.19 0.67
## PBB_Stk  0.543              0.120 0.19 1.01
## PFl_Stk  1.015              0.043 0.95 1.16
## PHse_Stk 0.437              0.119 0.19 0.64
## PGen_Stk 0.345              0.035 0.25 0.55
## PImp_Stk 0.781              0.115 0.33 2.30
## PSS_Tub  0.825              0.061 0.50 0.98
## PPk_Tub  1.077              0.030 0.98 1.24
## PFl_Tub  1.189              0.014 0.69 1.47
## PHse_Tub 0.569              0.072 0.33 1.27
```

```
# Market share, and market share by product characteristics.
marketshare=as.matrix(table(datchoice$choice)/nrow(datchoice))
colnames(marketshare)=c("marketshare")
rownames(marketshare)=c("PPk_Stk", "PBB_Stk", "PFl_Stk",  "PHse_Stk", "PGen_Stk", "PImp_Stk", "PSS
_Tub",  "PPk_Tub",  "PFl_Tub",  "PHse_Tub")

marketshare
```

```
##          marketshare
## PPk_Stk   0.39507830
## PBB_Stk   0.15637584
## PFl_Stk   0.05436242
## PHse_Stk  0.13266219
## PGen_Stk  0.07046980
## PImp_Stk  0.01655481
## PSS_Tub   0.07136465
## PPk_Tub   0.04541387
## PFl_Tub   0.05033557
## PHse_Tub  0.00738255
```

```
choice_new=t(apply(datchoice[,3:12], 1,function(x) x > apply(datchoice[,3:12],2,mean)))

choicec=data.frame(cbind(datchoice[,2], choice_new))
colnames(choicec)=c("choice",1:10)
choicef=choicec %>%
  pivot_longer(!choice, names_to = "choicee", values_to = "over_avg") %>%
  filter(choice == choicee) %>%
  select(choice, over_avg)

under=as.character(t(table(choicef))[1,]/length(datchoice$choice))
over=as.character(t(table(choicef))[2,]/length(datchoice$choice))

marketshare_price=cbind(under,over)
colnames(marketshare_price)=c("Under_Mean_Price","Over_Mean_Price")
rownames(marketshare_price)=c("PPk_Stk", "PBB_Stk", "PFl_Stk",  "PHse_Stk", "PGen_Stk", "PImp_Stk"
, "PSS_Tub",  "PPk_Tub",  "PFl_Tub",  "PHse_Tub")

marketshare_price
```

```
##          Under_Mean_Price      Over_Mean_Price
## PPk_Stk  "0.218791946308725"   "0.176286353467562"
## PBB_Stk  "0.0975391498881432"  "0.0588366890380313"
## PFl_Stk  "0.0425055928411633"  "0.0118568232662192"
## PHse_Stk "0.0664429530201342"  "0.0662192393736018"
## PGen_Stk "0.0391498881431767"  "0.0313199105145414"
## PImp_Stk "0.0125279642058166"  "0.00402684563758389"
## PSS_Tub  "0.0266219239373602"  "0.0447427293064877"
## PPk_Tub  "0.0194630872483221"  "0.0259507829977629"
## PFl_Tub  "0.00559284116331096" "0.0447427293064877"
## PHse_Tub "0.00357941834451902" "0.00380313199105145"
```

```
datdemos=margarine$demos
datchoice=left_join(datchoice, datdemos, "hhid")
choiceattribute_1=datchoice %>% group_by(choice) %>% summarize(famsize_1_2=sum(Fs3_4==0&Fs5.==0),
                                                              famsize_3_4=sum(Fs3_4==1&Fs5.==0),
                                                              famsize_5=sum(Fs3_4==0&Fs5.==1),
                                                              college=sum(college==1),
                                                              whitecollar=sum(whtcollar==1),
                                                              retired=sum(retired==1))

choiceattribute_2=datchoice %>% group_by(choice) %>% summarize(not_college=sum(college==0),
                                                              not_whitecollar=sum(whtcollar==0),
                                                              not_retired=sum(retired==0))
choiceattribute=merge(choiceattribute_1, choiceattribute_2)
choiceattribute
```

```
##    choice famsize_1_2 famsize_3_4 famsize_5 college whitecollar retired
## 1       1         622         902       242     561        1007     352
## 2       2         261         360        78     219         380     168
## 3       3         161          62        20     110         132     129
## 4       4         177         298       118     174         351      91
## 5       5          65         187        63      86         225      46
## 6       6          33          18        23      32          42      28
## 7       7         142         157        20     103         184      47
## 8       8          70         122        11      52         116      20
## 9       9         146          68        11      62         130      81
## 10     10           3          12        18      15          31       4
##    not_college not_whitecollar not_retired
## 1         1205             759        1414
## 2          480             319         531
## 3          133             111         114
## 4          419             242         502
## 5          229              90         269
## 6           42              32          46
## 7          216             135         272
## 8          151              87         183
## 9          163              95         144
## 10          18               2          29
```

Exercise 2

```
set.seed(100)
#choice matrix
ni=nrow(datchoice)
nj=ncol(datchoice[,3:12])
Y=matrix(0, ni,nj)
for(i in 1:nj){
  for(j in 2:ni){
    if(datchoice$choice[j]==i){
      Y[j,i]=1
    }
  }
}
Y[1,1]=1
#Likelihood Function
price <- datchoice[,3:12]
likelihood=function(x,beta) {
    coef=exp(matrix(rep(c(0,beta[1:9]),nrow(x)),byrow=TRUE,nrow(x))+x*beta[10])
    coef_sum=apply(coef,1,sum)
    return(coef/coef_sum)
}
llike=function(y,x,beta) {
  lprob=log(likelihood(x,beta))
  return(-sum(Y*lprob))
}
#optimization
model1=optim(function(beta) llike(y=y,x=price,b=beta),par=runif(10),method="BFGS")
as.matrix(model1$par)
```

```
##             [,1]
##  [1,] -0.9543115
##  [2,]  1.2969547
##  [3,] -1.7173309
##  [4,] -2.9040096
##  [5,] -1.5153362
##  [6,]  0.2517688
##  [7,]  1.4648734
##  [8,]  2.3574900
##  [9,] -3.8965893
## [10,] -6.6565873
```

Use conditional logit model in EX2, because price is the same for all households
The last coefficient here means that price and demand are negatively related–that is, higher price results in a less probability for the product to be purchased.
The other coefficients are the intercepts of good 2 to 10. Each means that comparing to good 1, an individual is more likely to choose that good if the coefficient is positive, and less likely to choose that good if the coefficient is negative.

Exercise 3

```
p=as.matrix(datchoice[,13],ncol=1)
#Likelihood Function
mlike=function(x,beta) {
  coef=exp(matrix(rep(c(0,beta[1:9]),nrow(x)),byrow=TRUE,nrow(x))+t(apply(x,1,function(x)x*c(0,be
ta[10:18]))))
  coef_sum=apply(coef,1,sum)
  return(coef/coef_sum)
}
mllike=function(y,x,beta) {
  lprob=log(mlike(x,beta))
  return(-sum(Y*lprob))
}
#optimization
model2=optim(function(beta) mllike(y=y,x=p,b=beta),par=runif(18),method="BFGS")
as.matrix(model2$par)
```

```
##               [,1]
##  [1,] -0.843533047
##  [2,] -2.397662227
##  [3,] -1.199390367
##  [4,] -1.688620334
##  [5,] -4.137003599
##  [6,] -1.529162746
##  [7,] -2.846028547
##  [8,] -2.573263700
##  [9,] -4.280023111
## [10,] -0.003156315
## [11,]  0.014508085
## [12,]  0.003978461
## [13,] -0.001327650
## [14,]  0.030525080
## [15,] -0.007004836
## [16,]  0.022806695
## [17,]  0.017662810
## [18,]  0.010708847
```

Use multinomial logit model in EX2, because income is not the same for households
The last 9 coefficients here are the effect of income(good 2 to 10), which mean that comparing to the probability of purchasing product 1, individual will more likely to choose that good if positive, and less likely to choose that good if negative. The first 9 coefficients are the intercepts(good 2 to 10).

Exercise 4

```
#marginal effect for model 1(conditional logit)
pij=likelihood(price,model1$par)
mid=array(0,dim = c(nrow(price),10,10))
for (i in 1:nrow(price)) {
  diag(mid[i,,]) <- 1
}
llikem=array(0,dim=c(nrow(price),10,10))
for (i in 1:nrow(price)) {
  for (j in 1:10) {
    for (k in 1:10) {
      llikem[i,j,k]=pij[i,j]*(mid[i,j,k]-pij[i,k])*model1$par[10]
    }
  }
}
me_model1=apply(llikem,c(2,3),mean)
colnames(me_model1)=c("Choice 1" , "Choice 2", "Choice 3",
                            "Choice 4" , "Choice 5", "Choice 6",
                            "Choice 7", "Choice 8", "Choice 9",
                            "Choice 10")
row.names(me_model1)=c("p1", "p2" ,"p3" ,"p4" ,"p5", "p6"
                           , "p7", "p8", "p9" , "p10")
me_model1
```

```
##        Choice 1    Choice 2     Choice 3     Choice 4     Choice 5
## p1  -1.28526978  0.29537065  0.120709754  0.295086008  0.156227754
## p2   0.29537065 -0.74542736  0.055078713  0.133453135  0.072824451
## p3   0.12070975  0.05507871 -0.337447495  0.050543372  0.030280613
## p4   0.29508601  0.13345314  0.050543372 -0.712667402  0.064016214
## p5   0.15622775  0.07282445  0.030280613  0.064016214 -0.428081938
## p6   0.03731977  0.01672548  0.007104370  0.016550686  0.008748446
## p7   0.15359738  0.06927123  0.029268300  0.063744472  0.037948176
## p8   0.09929548  0.04520668  0.019664353  0.039262341  0.025090109
## p9   0.11081939  0.05069878  0.021753610  0.044153488  0.028519378
## p10  0.01684359  0.00679825  0.003044411  0.005857685  0.004426798
##          Choice 6     Choice 7     Choice 8     Choice 9     Choice 10
## p1   0.0373197732  0.153597377  0.099295484  0.110819394  0.0168435932
## p2   0.0167254776  0.069271229  0.045206679  0.050698780  0.0067982497
## p3   0.0071043703  0.029268300  0.019664353  0.021753610  0.0030444108
## p4   0.0165506863  0.063744472  0.039262341  0.044153488  0.0058576853
## p5   0.0087484461  0.037948176  0.025090109  0.028519378  0.0044267977
## p6  -0.1073199177  0.008537635  0.005430082  0.006113331  0.0007901167
## p7   0.0085376346 -0.420295875  0.025793327  0.027921326  0.0042140336
## p8   0.0054300824  0.025793327 -0.282465103  0.019789143  0.0029335849
## p9   0.0061133306  0.027921326  0.019789143 -0.313050536  0.0032820876
## p10  0.0007901167  0.004214034  0.002933585  0.003282088 -0.0481905596
```

Coefficients on diagonal are negative while others are positive. This shows that people go ahead and switch to other goods if price of one good increases.

```
#marginal effect for model 2(multinomial logit)
pij_m2=mlike(p,model2$par)
mb=c(0,model2$par[10:18])
me_model2=array(0,dim=c(nrow(p),10))
for (i in 1:nrow(p)) {
  be=sum(pij_m2[i,]*mb)
  for (j in 1:10) {
    me_model2[i,j] <- pij_m2[i,j]*(mb[j]-be)
  }
}
for (i in 1:nrow(p)) {
  be=sum(pij_m2[i,]*mb)
  me_model2[i,]=pij_m2[i,]*(mb-be)
}
me_model2=apply(me_model2, 2, mean)
me_model2
```

```
##  [1] -1.050347e-03 -9.016117e-04  6.267569e-04  1.658219e-04 -2.794050e-04
##  [6]  4.430863e-04 -6.822410e-04  8.861428e-04  7.339613e-04  5.783514e-05
```

Exercise 5 IIA

```
#beta_f
mixlike=function(y,x,beta,prob) {
  lprob=log(prob(x,beta))
  return(-sum(y*lprob))
}
D=as.matrix(datchoice[,3:13],ncol=1)
mixprob=function(x,beta) {
    coef=exp(matrix(rep(c(0,beta[1:9]),nrow(x)),byrow = TRUE,nrow(x))+x[,1:10]*beta[10]+t(apply(ma
trix(x[,11],ncol=1),1,function(x)x*c(0,beta[11:19]))))
    coef_sum <- apply(coef,1,sum)
    return(coef/coef_sum)
}
mixmodel=optim(function(beta) mixlike(y=Y,x=D,beta=beta,prob=mixprob),par=runif(19),method="BFGS"
)
mix_beta_f=as.matrix(mixmodel$par)
mix_beta_f
```

```
##                [,1]
##  [1,] -0.840489424
##  [2,]  0.883034490
##  [3,] -1.830065411
##  [4,] -2.877735367
##  [5,] -2.458062981
##  [6,]  0.501398330
##  [7,]  0.805186136
##  [8,]  1.847978652
##  [9,] -4.121845355
## [10,] -6.660770052
## [11,] -0.004281645
## [12,]  0.014485122
## [13,]  0.004133811
## [14,] -0.001075222
## [15,]  0.029844956
## [16,] -0.009388809
## [17,]  0.021968056
## [18,]  0.017369572
## [19,]  0.008186322
```

```
#beta_r
#remove first choice
D_new=D[,-1]
mixlike2=function(x,beta) {
    coef=exp(matrix(rep(c(0,beta[1:8]),nrow(x)),byrow=TRUE,nrow(x))+x[,1:9]*beta[9]+t(apply(matrix
(x[,10],ncol=1),1,function(x)x*c(0,beta[10:17]))))
    coef_sum <- apply(coef,1,sum)
    return(coef/coef_sum)
}
mixmodel2=optim(function(beta) mixlike(y=Y[,-1],x=D_new,beta=beta,prob=mixlike2),par=runif(17),me
thod="BFGS")
mix_beta_r=as.matrix(mixmodel2$par)
mix_beta_r
```

```
##                 [,1]
##  [1,]  1.634153686
##  [2,] -0.941193155
##  [3,] -1.966480219
##  [4,] -1.642871816
##  [5,]  1.222007155
##  [6,]  1.561820515
##  [7,]  2.581206747
##  [8,] -3.229496776
##  [9,] -6.420216685
## [10,]  0.018403993
## [11,]  0.007447584
## [12,]  0.003006638
## [13,]  0.033444126
## [14,] -0.004327203
## [15,]  0.025761506
## [16,]  0.020857034
## [17,]  0.012314322
```

```
#MTT
lbf=mixlike(y=Y[,-1],x=D_new,beta=mixmodel$par[-c(1,11)],prob=mixlike2)
lbr=mixlike(y=Y[,-1],x=D_new,beta=mixmodel2$par,prob=mixlike2)
MTT=2*(lbf-lbr)
t=qchisq(0.99, length(mixmodel2$par))
MTT<t
```

```
## [1] FALSE
```

Since MTT is greater than t, we reject the null hypothesis that IIA holds.