HW7_Conjoint Analysis

Raveena Kamal

2024-04-11

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
design_DF <- read.csv("survey_design_2.csv")
design_DF
```

```
##    Screen Cell Price Battery      OS
## 1       7    Y   300      12 Windows
## 2       7    Y   100       8 Windows
## 3      10    Y   500      12 Android
## 4       7    Y   300       4 Android
## 5       7    N   300       8     iOS
## 6      10    N   300      12 Windows
## 7       7    N   500      12 Android
## 8      10    N   300       8 Android
## 9       7    N   500       4 Windows
## 10     10    Y   100      12     iOS
## 11     10    Y   300       4     iOS
## 12      7    N   100      12     iOS
## 13     10    Y   500       8 Windows
## 14     10    N   500       4     iOS
## 15     10    N   100       4 Windows
## 16     10    N   100       8 Android
## 17      7    Y   500       8     iOS
## 18      7    Y   100       4 Android
```

```
responses_DF <- read.csv("respondent_data_2.csv")
N <- nrow(responses_DF)
summary(responses_DF)
```

```
##  respondent_id     profile_1     profile_2       profile_3       profile_4
##  Min.   : 1.00   Min.   :1.0   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:23.25   1st Qu.:3.0   1st Qu.:4.000   1st Qu.:2.000   1st Qu.:2.000
##  Median :45.50   Median :4.0   Median :5.000   Median :4.000   Median :2.000
##  Mean   :45.50   Mean   :4.2   Mean   :4.556   Mean   :3.667   Mean   :2.533
##  3rd Qu.:67.75   3rd Qu.:5.0   3rd Qu.:6.000   3rd Qu.:5.000   3rd Qu.:3.000
```

```
##  Max.   :90.00   Max.   :7.0   Max.   :7.000   Max.   :7.000   Max.   :6.000
##    profile_5      profile_6       profile_7       profile_8       profile_9
##  Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:3.00   1st Qu.:3.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:1.000
##  Median :4.00   Median :4.000   Median :3.000   Median :3.500   Median :1.000
##  Mean   :4.20   Mean   :4.333   Mean   :2.833   Mean   :3.478   Mean   :1.989
##  3rd Qu.:5.75   3rd Qu.:5.000   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:2.000
##  Max.   :7.00   Max.   :7.000   Max.   :7.000   Max.   :7.000   Max.   :6.000
##    profile_10      profile_11      profile_12      profile_13
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:7.000   1st Qu.:2.000   1st Qu.:4.000   1st Qu.:3.000
##  Median :7.000   Median :3.500   Median :5.000   Median :4.000
##  Mean   :6.433   Mean   :3.633   Mean   :5.067   Mean   :3.644
##  3rd Qu.:7.000   3rd Qu.:5.000   3rd Qu.:6.000   3rd Qu.:5.000
##  Max.   :7.000   Max.   :7.000   Max.   :7.000   Max.   :7.000
##    profile_14      profile_15      profile_16   profile_17      profile_18
##  Min.   :1.000   Min.   :1.000   Min.   :1    Min.   :1.000   Min.   :1.0
##  1st Qu.:2.000   1st Qu.:2.000   1st Qu.:3    1st Qu.:3.000   1st Qu.:2.0
##  Median :3.000   Median :3.000   Median :4    Median :4.000   Median :3.5
##  Mean   :2.989   Mean   :3.389   Mean   :4    Mean   :3.956   Mean   :3.4
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:5    3rd Qu.:5.000   3rd Qu.:5.0
##  Max.   :7.000   Max.   :7.000   Max.   :7    Max.   :7.000   Max.   :7.0
```

Question 2

```
lm_res <- vector(mode="list", length=nrow(responses_DF))
for (i in 1:nrow(responses_DF)) {
   response = as.numeric(responses_DF[i,2:ncol(responses_DF)])
   est_DF = cbind(design_DF, response=response)

   lm_res[[i]] = lm(response ~ (Screen) + factor(Cell) + (Price) +
                      (Battery) + factor(OS),
                    data=est_DF)
 }
summary(lm_res[[50]])
```

```
##
## Call:
## lm(formula = response ~ (Screen) + factor(Cell) + (Price) + (Battery) +
##     factor(OS), data = est_DF)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.04167 -0.43750 -0.08333  0.27708  1.62500
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.633333   1.319570   1.238  0.24157
## Screen           -0.025000   0.128879  -0.194  0.84973
## factor(Cell)Y     0.325000   0.386638   0.841  0.41848
## Price            -0.002500   0.001177  -2.125  0.05708 .
## Battery           0.291667   0.058825   4.958  0.00043 ***
```

```
## factor(OS)iOS      1.333333   0.470600   2.833  0.01628 *
## factor(OS)Windows  1.000000   0.470600   2.125  0.05708 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8151 on 11 degrees of freedom
## Multiple R-squared:  0.7782, Adjusted R-squared:  0.6572
## F-statistic: 6.431 on 6 and 11 DF,  p-value: 0.004075
```

##Q3

```
res_list<-lm_res

prods_DF <- data.frame(Screen = c(10,10),
                       Cell = c("Y","N"),
                       Price = c(500,300),
                       Battery = c(8,8),
                       OS = c("iOS","Android"))

rownames(prods_DF) = c("iPad","Google_A")

comp_demand <- function(res_list, prods_DF) {
  # initialize
  N = length(res_list) # number subjects
  choices = rep(0,N)
  # loop over subjects: predict ratings/utilities, determine expected choices
  for (i in 1:N) {
    ratings = predict(res_list[[i]], newdata = prods_DF)
    choices[i] = which.max(ratings)
  }
  # calculate demand for each product (rows in prods_DF)
  N_prod = nrow(prods_DF) # number products
  demand = rep(0,N_prod)
  # loop over products: calculate aggregate demand
  for (i in 1:N_prod) {
    demand[i] = sum(choices==i)
  }
  # label the output
  names(demand) = rownames(prods_DF)
  # return values
  return(demand)
}

prods_DF.results <- comp_demand(res_list, prods_DF)
print(prods_DF.results)
```

```
##     iPad Google_A
##       61       29
```

##Q4

```r
comp_cost <- function(prods_DF) {
  N_prod = nrow(prods_DF) # number products
  cost = rep(0, N_prod)
  for (i in 1:N_prod) {
    cost[i] = 80 + 5*(prods_DF$Screen[i]==10) + 15*(prods_DF$Cell[i]=="Y") + 10*(prods_DF$Battery[i]==8)
  }
  names(cost) = rownames(prods_DF)
  return(cost)
}

comp_cost(prods_DF)
```

```
##     iPad Google_A
##      110       95
```

##Q5

```r
profit1 <- function(res_list, prods_DF, sum_ndx) {
  # Calculate demand for all products
  demand <- comp_demand(res_list, prods_DF)

  cost <- comp_cost(prods_DF)

  # Initialize total profit
  total_profit <- 0

  for (i in length(sum_ndx)) {
    ndx = sum_ndx[i]
    Q <- demand[ndx]
    P <- prods_DF[ndx, "Price"]
    MC <- cost[ndx]
    profit <- Q * (P - MC)
    total_profit <- total_profit + profit


  }

  return(total_profit)
}
expected_profit <- profit1(res_list, prods_DF, 2)
print(expected_profit)
```

```
## Google_A
##     5945
```

##Q6

```r
profit2 <- function(lm_res, prods_DF, sum_ndx, price) {
  prods_DF[sum_ndx,"Price"] = price
  pft = profit1(lm_res, prods_DF, sum_ndx)
  return(pft)
}
```

```r
expected_profit2 <- profit2(res_list, prods_DF, 2, 450)
print(expected_profit2)
```

```
## Google_A
##     5680
```

##Q7

```r
pxs <- seq(100, 500, by = 10)
pft <- rep(0, length(pxs))
# Iterate over each price and calculate profit
for (i in seq_along(pxs)) {
pft[i] <- profit2(lm_res, prods_DF, 2, pxs[i])
}
# Find the index of the maximum profit
max_profit_index <- which.max(pft) # Get the profit-maximizing price
profit_maximizing_price <- pxs[max_profit_index]
# Print the profit-maximizing price
print(profit_maximizing_price)
```

```
## [1] 380
```

##Q8

```r
price_points <- c(100, 300, 500)

#  dataframe with all feasible designs that Google can produce
allprods_DF <- expand.grid(Screen = unique(design_DF$Screen),
                           Cell = unique(design_DF$Cell),
                           Price = price_points,
                           Battery = unique(design_DF$Battery),
                           OS = 'Android')
# Filter to include only designs with the Android OS

nrow(allprods_DF)
```

```
## [1] 36
```

##Q9

```r
nProducts <- nrow(allprods_DF)
pft <- rep(0,nProducts)

for (i in 1:nProducts) {
  prods_new <- data.frame(Screen = c(10, allprods_DF[i,"Screen"]),
                      Cell = c("Y", as.character(allprods_DF[i,"Cell"])),
                      Price =  c(500, allprods_DF[i,"Price"]),
                      Battery = c(8, allprods_DF[i,"Battery"]),
                      OS =  c("iOS", "Android"))
```

```
  pft[i] <- profit1(lm_res, prods_new, 2)
}

max_profit <- max(pft)
max_index <- which.max(pft)
print(allprods_DF[max_index,])
```

```
##    Screen Cell Price Battery      OS
## 10     10    Y   500      12 Android
```