MSPA PREDICT 450-DL-55 LEC

Micro 2: Discrete Choice Experiment

Darryl Buswell

```
# Load dataset/functions
load("data/efCode.RData")
load("data/stc-cbc-respondents-v6.RData")
df_resp.raw <- resp.data.v5mod
df_task <- read.csv("data/stc-dc-task-cbc-v6.csv")

# Generate effects coded version of task.mat
df_task.att <- as.matrix(df_task[, 3:7])
df_X <- efcode.attmat.f(df_task.att)
colnames(df_X) <- c("screen_2", "screen_3",
                    "RAM_2", "RAM_3",
                    "processor_2", "processor_3",
                    "price_2", "price_3",
                    "brand_2", "brand_3", "brand_4")

# Get vector of prices centered on mean
pricevec <- df_task$price - mean(df_task$price)

# Get the columns from X.mat that represent brand
df_X.brand <- df_X[, 9:11]
colnames(df_X.brand) <- colnames(df_X[,9:11])

# Multiply each column in X.brands by pricevec
df_X.brandbyprice <- df_X.brand * pricevec
colnames(df_X.brandbyprice) = c("brand_2_by_Price", "brand_3_by_Price", "brand_4_by_Price")

# Combine X.mat and X.BrandsByPrice to get the X matrix for choice modelling
df_X <- cbind(df_X, df_X.brandbyprice)

# Get the survey responses
df_y.resp <- df_resp.raw[, 3:38]
df_y.resp <- na.omit(df_y.resp)
df_y.resp <- as.matrix(df_y.resp)

# Create list of data for each respondent
ls_lgtdata <- NULL
for (i in 1:360)  { # 360 should be the number of respondents w/o NA"s
  ls_lgtdata[[i]] <- list(y = df_y.resp[i,], X = df_X)
}
```

1 How many 1, 2, and 3 responses are there in ydata? Tabulate and indicate how many 1's, 2's and 3's there are.

The below table shows the count of responses for each of the three possible choice alternatives:

```
table(df_y.resp)
```

```
## df_y.resp
##    1    2    3
## 3551 3908 5501
```

2 Calculate the sum of each column in X.matrix, and report what each column sum is.

The 'X-matrix' is a merge of the effects coded version of the 108 possible combinations of attributes from the original dataset, as well as the matrix of interactions between brand and price. Since this matrix is already effects coded and is to be representative of all possible combinations of attributes, we would expect the sum of each column to be zero. The sum of columns for the X-matrix is shown below:

```
colSums(df_X)
```

```
##         screen_2         screen_3            RAM_2            RAM_3 
##                0                0                0                0 
##      processor_2      processor_3          price_2          price_3 
##                0                0                0                0 
##          brand_2          brand_3          brand_4 brand_2_by_Price 
##                0                0                0                0 
## brand_3_by_Price brand_4_by_Price 
##                0                0
```

3 Indicate whether you think that the regression coefficients for your models will be better estimates or worse estimates if the correlations between the columns are large, and explain why.

The 'X-matrix' represents the predictor variables to be used for regression model estimation. Each combination of the original choice set attributes should be unique, and as such, we would hope that there is no correlation between variables of different measures within this matrix. We do note a reported correlation coefficient of 0.5 between attributes of the same type however, since the effects coding generates a value of negative one for all levels of the attribute when the reference value of that attribute is true. Correlations for the X-matrix variables are shown below:

```
df_X.corr <- cor(df_X)
kable(df_X.corr)
```

	screen_2	screen_3	RAM_2	RAM_3	processor_2	processor_3	price_2	price_3	brand_2
screen_2	1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
screen_3	0.5	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
RAM_2	0.0	0.0	1.0	0.5	0.0	0.0	0.0	0.0	0.0
RAM_3	0.0	0.0	0.5	1.0	0.0	0.0	0.0	0.0	0.0
processor_2	0.0	0.0	0.0	0.0	1.0	0.5	0.0	0.0	0.0
processor_3	0.0	0.0	0.0	0.0	0.5	1.0	0.0	0.0	0.0
price_2	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.5	0.0
price_3	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0	0.0
brand_2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
brand_3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5
brand_4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5
brand_2_by_Price	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
brand_3_by_Price	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
brand_4_by_Price	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

4 What is the result of the command sum(lgtdata[[12]]$y)?, and what is the result of the command table(lgtdata[[35]]$y)?

The 'ls_lgtdata' object is a list of combinations of each respondents selection to the 36 choice questions (y-matrix) and the matrix of predictor variables (X-matrix). As such, 'ls_lgtdata[[12]]$y' will return the 36 choice question responses made by the 12th respondent, while the sum of this vector will return the sum of choices made by that respondent. The sum of the vector for the 12th respondent is shown below:

```
sum(ls_lgtdata[[12]]$y)
```

```
## [1] 78
```

We can also table the amount of 1, 2 and 3 responses made by the 35th respondent by calling 'table(ls_lgtdata[[35]]$y)'. The table of responses for the 35th respondent is shown below:

```
table(ls_lgtdata[[35]]$y)
```

```
##
##  1  2  3
## 10 14 12
```