HW 1

```r
# load package
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)

## Load the data
df <- read.csv('Conjoint Data.csv')
head(df)
```

```
##   Preference.Rank Screen.75.inch Screen.85.inch Resolution.4K...1 Sony...1
## 1              15              1              0                 0        1
## 2              24              0              1                 0        1
## 3               5              0              0                 0        1
## 4              20              1              0                 1        1
## 5              23              0              1                 1        1
## 6               2              0              0                 1        1
##   Price...low...0..high..1.
## 1                         0
## 2                         0
## 3                         0
## 4                         0
## 5                         0
## 6                         0
```

```r
# Getting the preference vector and design matrix
pre_all <- df$Preference.Rank
design_matrix <- df[,2:6]
```

```r
# Conjoint Function
my_conjoint <- function(preferences, plotit=FALSE){
```

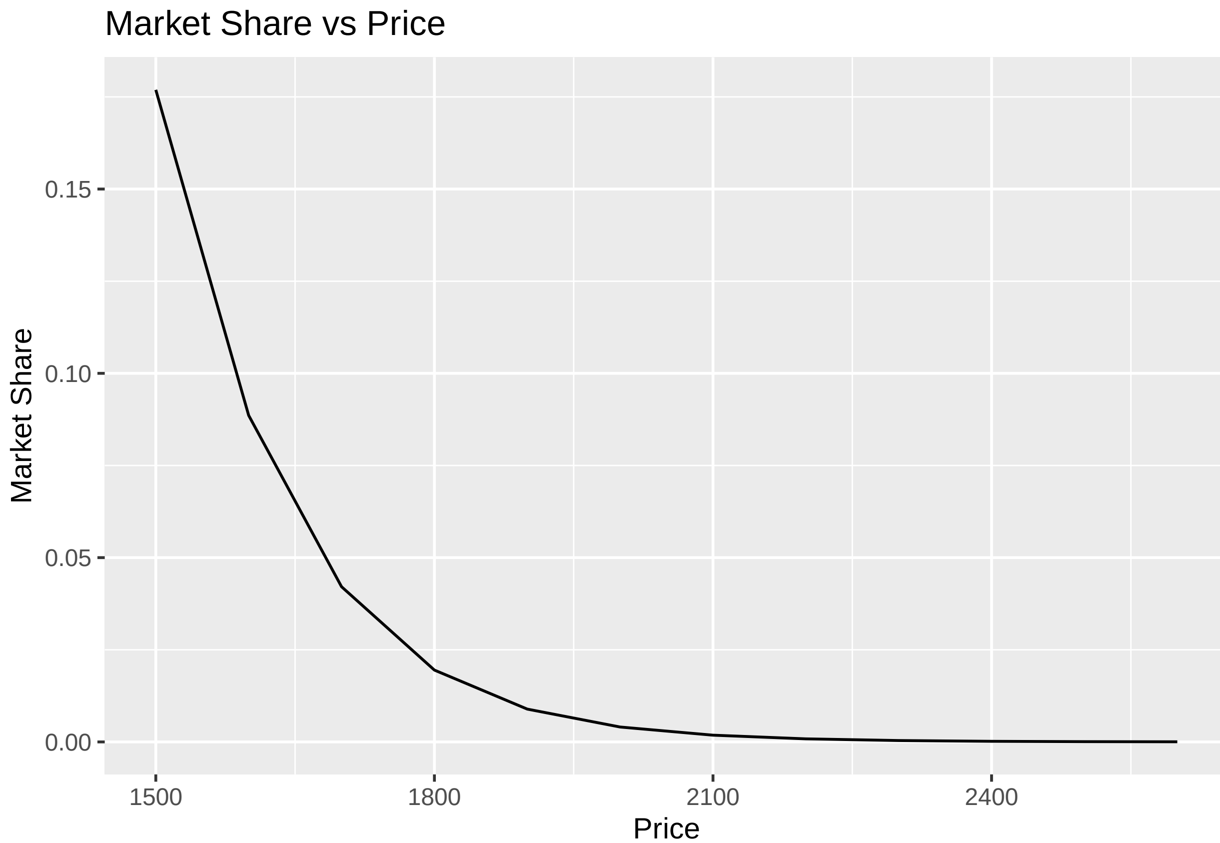
```r
# transform data types and store variables
screen_75 <- factor(design_matrix$Screen.75.inch)
screen_85 <- factor(design_matrix$Screen.85.inch)
resolution_4k <- factor(design_matrix$Resolution.4K...1)
sony <- factor(design_matrix$Sony...1)
high_price <- factor(design_matrix$Price...low...0..high..1.)

# run regression model and extract the coefficients
model_reg <- lm(preferences~screen_75+screen_85+resolution_4k+sony+high_price)
partworth <- coef(summary(model_reg))[,1]

#############################################################
# output 1: regression model summary for partworth, se, tval#
#############################################################
mod_summary <- coef(summary(model_reg))[,1:3]

#################################
# output 2: attribute importance#
#################################
attr_importance <- data.frame(matrix(NA, nrow=4, ncol=2))
colnames(attr_importance) <- c('Range', 'Importance')
rownames(attr_importance) <- c('screen_size', 'resolution', 'brand','price')

# attribute importance - Range
attr_importance[1,1] <- max(mod_summary[2:3,1], 0) - min(mod_summary[2:3,1], 0)
attr_importance[2,1] <- max(mod_summary[4,1], 0) - min(mod_summary[4,1], 0)
attr_importance[3,1] <- max(mod_summary[5,1], 0) - min(mod_summary[5,1], 0)
attr_importance[4,1] <- max(mod_summary[6,1], 0) - min(mod_summary[6,1], 0)

# attribute importance - Importance
attr_importance[1,2] <- attr_importance[1,1]/sum(attr_importance$Range)
attr_importance[2,2] <- attr_importance[2,1]/sum(attr_importance$Range)
attr_importance[3,2] <- attr_importance[3,1]/sum(attr_importance$Range)
attr_importance[4,2] <- attr_importance[4,1]/sum(attr_importance$Range)

##############################
# output 3: Willingess to Pay#
##############################
Sony_design <- c(1,1,0,1,1,2500)
Sharp_design <- c(1,0,1,1,0,2000)
point_val <- (Sharp_design[6]-Sony_design[6])/partworth[6]
WTP <- as.data.frame(point_val*partworth, col.names='WTP')

###############################
# output 4: Willingness to Pay#
###############################
price_saving <- 2500 - 2000
util <- price_saving / abs(mod_summary[6,1])
wtp <- data.frame(matrix(NA, nrow=4, ncol=1))
colnames(wtp) <- c("Willingness To Pay")
rownames(wtp) <- c('75" screen_size', '85" screen_size', 'resolution', 'brand')
wtp[1,1] <- mod_summary[2,1] * util # WTP for 75" screen_size
wtp[2,1] <- mod_summary[3,1] * util # WTP for 85" screen_size
```

```
wtp[3,1] <- mod_summary[4,1] * util # WTP for 4k resolution
wtp[4,1] <- mod_summary[5,1] * util # WTP for Sony brand name

####################################
# output 5: conjoint analysis table#
####################################

 # create blank result matrix
CA <- data.frame(matrix(NA, nrow=12, ncol=6))
CA[,1] <- seq(1500, 2600, by=100)
colnames(CA) <- c('price', 'utility_mydesign', 'market_share', 'sales', 'margin', 'profit')

# use for loop to get all the results for each price
for (i in 1:12){

  # create matrix for my_design, sony, sharp, and costs
  my_design <- c(1,0,1,0,0,CA[i,1])
  Sony <- c(1,1,0,1,1,2500)
  Sharp <- c(1,0,1,1,0,2000)
  costs <- c(1000,500,1000,250,250, NA)
  design_info <- rbind(partworth,my_design, Sony, Sharp, costs)
  colnames(design_info) <- c('intercept','screen_75','screen_85','resolution_4k','sony','high_price')

  # Utility
  CA[i,2] <- sum(design_info['partworth',1:5] * design_info['my_design',1:5])+ design_info['partworth
  utility_sony <- sum(design_info['partworth',1:5] * design_info['Sony',1:5])+ design_info['partworth
  utility_sharp <- sum(design_info['partworth',1:5] * design_info['Sharp',1:5])+ design_info['partwor

  # Market Share
  CA[i,3] <- exp(CA[i,2])/sum(exp(CA[i,2]),exp(utility_sony),exp(utility_sharp))

  # Sales
  market_size <- 100
  CA[i,4] <- market_size*CA[i,3]

  # Margin
  Net_cost <- sum(design_info['my_design',1:5]*design_info['costs',1:5])
  CA[i,5] <- CA[i,1]-Net_cost

  # Profit
  CA[i,6] <- CA[i,4]*CA[i,5]
}

#############################################
# output 5: optimal price and maximum profit#
############################################
# convert CA to dataframe
df_CA <- as.data.frame(CA)

max_profit_val <- max(df_CA$profit)
max_profit <- sprintf('Maximum Profit: %f', max_profit_val)
opt_price_val <- df_CA$price[CA$profit == max_profit_val]
opt_price <- sprintf('Optimal Price: %.0f', opt_price_val)
```

```r
  ####################
  # output 6: plotting#
  ####################
  if(plotit){

    # Price vs Sales
    plot1 <- ggplot(aes(x=price, y=market_share), data=df_CA) + geom_line() + xlab('Price') + ylab('Marl

    # Price vs Profit
    plot2 <- ggplot(aes(x=price, y=profit), data=df_CA) + geom_line() + xlab('Price') + ylab('Profit') +



    out_graph <- list(plot1, plot2)
    print(out_graph)
  }


  out_table <- list(mod_summary, attr_importance, wtp, CA, opt_price, max_profit)
  return(out_table)

}
```

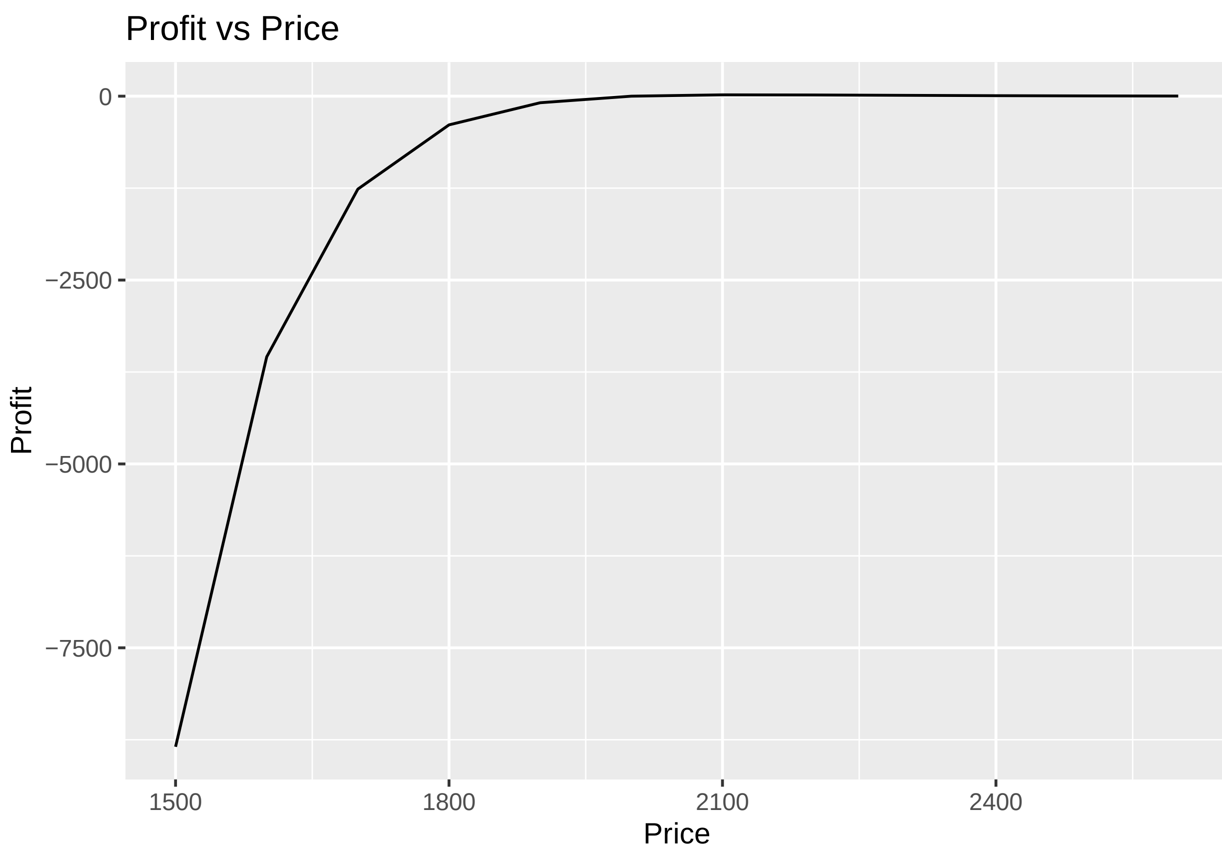
```r
# call function
my_conjoint(pre_all, plotit = TRUE)
```

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



Market Share vs Price
Market Share
0.15
0.10
0.05
0.00
1500
1800
2100
2400
Price


```
##
## [[2]]
```




```
## [[1]]
##                   Estimate Std. Error    t value
## (Intercept)       8.444892  0.2962094  28.509875
## screen_751        2.522849  0.2962094   8.517115
## screen_851        4.083333  0.2962094  13.785294
## resolution_4k1    5.471326  0.2418539  22.622439
## sony1             2.157706  0.2418539   8.921525
## high_price1      -3.967742  0.2418539 -16.405529
##
## [[2]]
##                 Range Importance
## screen_size 4.083333  0.2604149
## resolution  5.471326  0.3489342
## brand       2.157706  0.1376079
## price       3.967742  0.2530430
##
## [[3]]
##                   Willingness To Pay
## 75" screen_size             317.9201
## 85" screen_size             514.5664
## resolution                  689.4761
## brand                       271.9061
##
## [[4]]
##    price utility_mydesign market_share        sales margin       profit
## 1   1500        16.495968 1.769167e-01 17.691668904   -500 -8845.834452
```

```
## 2   1600        15.702419 8.859379e-02  8.859379444   -400 -3543.751778
## 3   1700        14.908871 4.210889e-02  4.210888861   -300 -1263.266658
## 4   1800        14.115323 1.949282e-02  1.949282246   -200  -389.856449
## 5   1900        13.321774 8.910522e-03  0.891052166   -100   -89.105217
## 6   2000        12.528226 4.049434e-03  0.404943434      0     0.000000
## 7   2100        11.734677 1.835376e-03  0.183537608    100    18.353761
## 8   2200        10.941129 8.308607e-04  0.083086070    200    16.617214
## 9   2300        10.147581 3.759172e-04  0.037591723    300    11.277517
## 10  2400         9.354032 1.700388e-04  0.017003878    400     6.801551
## 11  2500         8.560484 7.690503e-05  0.007690503    500     3.845252
## 12  2600         7.766935 3.478078e-05  0.003478078    600     2.086847
##
## [[5]]
## [1] "Optimal Price: 2100"
##
## [[6]]
## [1] "Maximum Profit: 18.353761"
```

Interpretation:

1. Partworth: Partworth is numerical score that measures how much each attribute/feature influences the customer's decision to select an alternative. Partworths are the coefficients of the product features in our linear regression model. And tells us how the ranking/preference is related to a certain attribute. In our case case, 4K resolution is the most influential feature as having 4K resolution would make the ranking of the product go up by 5.4, keeping all the other attributes constant.

2. Attribute Importance: As the relative importance of each attribute, it shows which attributes of a product are more or less important when we make a purchase. Based on the value of importance, the most important attribute is resolution (35%) and the least important attribute is brand (14%).

3. Willingness to pay: the conjoint shows that the customers are willing to pay pay $317 more for TV with 75 inch screen, $514 more for TV with 85 inch screen, $689 more for TV with 4K resolution, and $271 more for TV with Sony brand.

4. Optimal Price: The optimal price for "my design" is $2100, the price that would generate the most profit.

5. Maximum Profit: The maximum profit is $18 which can be achieved if we set the price of "my design" at $2100.

6. Market share (as function of Price): As we increase the prices, market share decreases. This is as expected since consumers are generally price-motivated, the higher the price, the lower the demand. The market share is a mere 1.83e-03 at our optimal pricepoint.

7. Profit (as function of Price): The plot shows the profit generated at each price point - we break even at $2000 and can generate the most profit at $2100 (although profit is minimal).