Toy Horse Conjoint Experiment
Case Assignment

MSBA Cohort 2 Group 22

Pin Li, Jiawen Liang, Ruiling Shen, Chenxi Tao, Khanh Tran

Benefit Segmentation

Decision-oriented Analysis

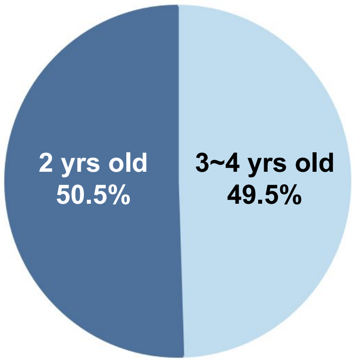
- **Context**
  - Underperforming product set
  - Fierce competition
  - Undefined consumers
- **Tool**
  - Conjoint Analysis
- **Analytical Goals**
  - Benefit segmentation
  - Product revitalization
- **Data**
  - Survey data
    - 200 respondents
- **Method**
  - A priori segmentation
    - Demographic information
  - Post-hoc segmentation
    - Individual part-utilities

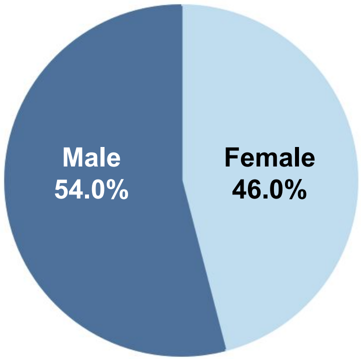
Product Profiles

Price	Height	Motion	Style
$139.99	18"	Rocking	Glamorous
$119.99	26"	Bouncing	Racing

4 attributes
16 profiles

Demographic Information of Respondents





· A **quota** sample
· Equal weight on the **age range**
· Can **represent the population** of buyers

Source: respondentData

Market Segmentation

Priori Segmentation

Age & Gender

Coefficient	P-Value
price:factor(seg)2	0.008761 **
price:factor(seg)3	0.866651
price:factor(seg)4	0.334917
size:factor(seg)2	0.047452 *
size:factor(seg)3	0.025311 *
size:factor(seg)4	4.04e-05 ***
motion:factor(seg)2	0.001331 **
motion:factor(seg)3	0.271504
motion:factor(seg)4	0.835281
style:factor(seg)2	0.000225 ***
style:factor(seg)3	0.888958
style:factor(seg)4	0.005487 **

Gender

Coefficient	P-Value
price:factor(seg)2	0.01454 *
size:factor(seg)2	0.00293 **
motion:factor(seg)2	0.00520 **
style:factor(seg)2	1.91e-05 ***

Age

Coefficient	P-Value
price:factor(seg)2	0.37977
size:factor(seg)2	0.00239 **
motion:factor(seg)2	0.02310 *
style:factor(seg)2	0.9502

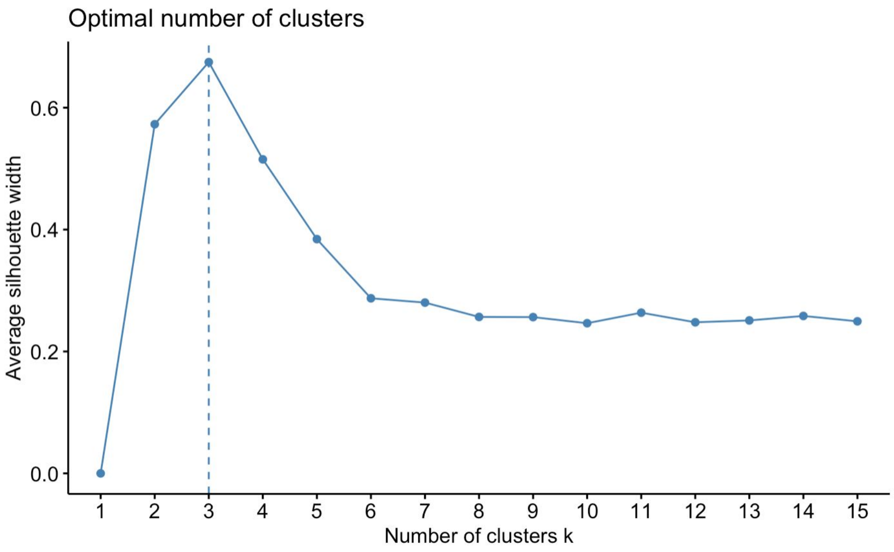
Findings:

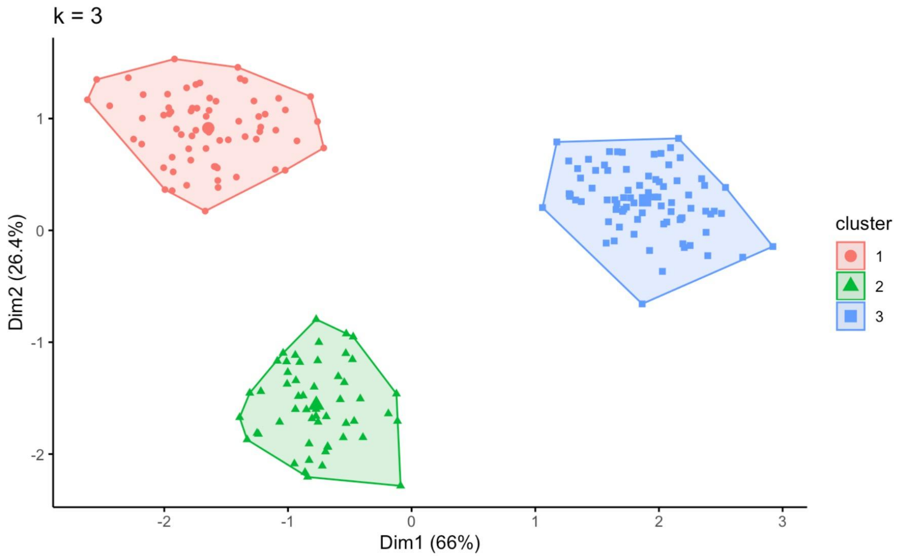
1. Age & Gender Segmentation **insignificant**.
2. Age Segmentation **insignificant**.
3. Gender Segmentation is **significant**.

Gender Segmentation

Market Segmentation

Benefit Segmentation





Optimal Number of Segments: 3

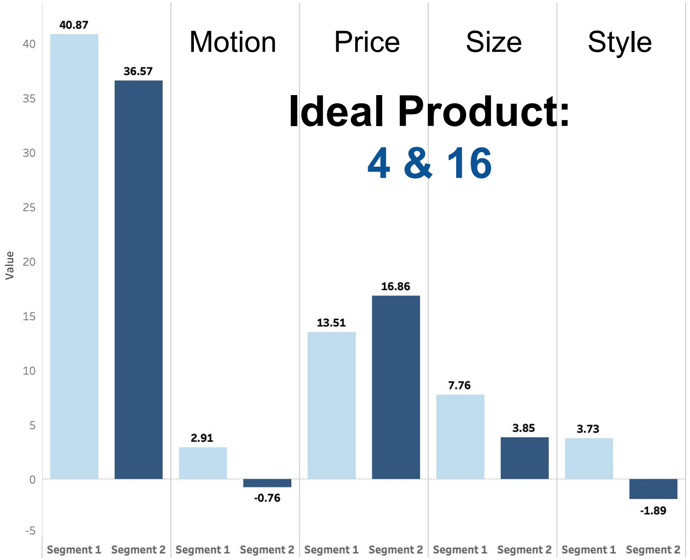
highest average silhouette width

three **clear, discrete** group

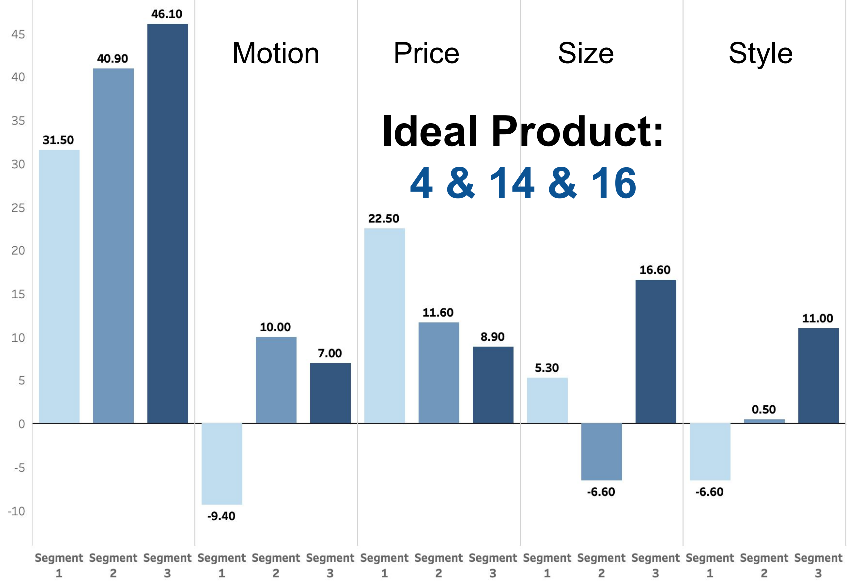
Market Segmentation

Conclusion

Priori Segmentation



Benefit Segmentation



The optimal product set from two segmentation are **mostly consistent**.
These products are **recommended inputs** for the scenarios in **Market Simulation**.

Market Simulation

Methodologies

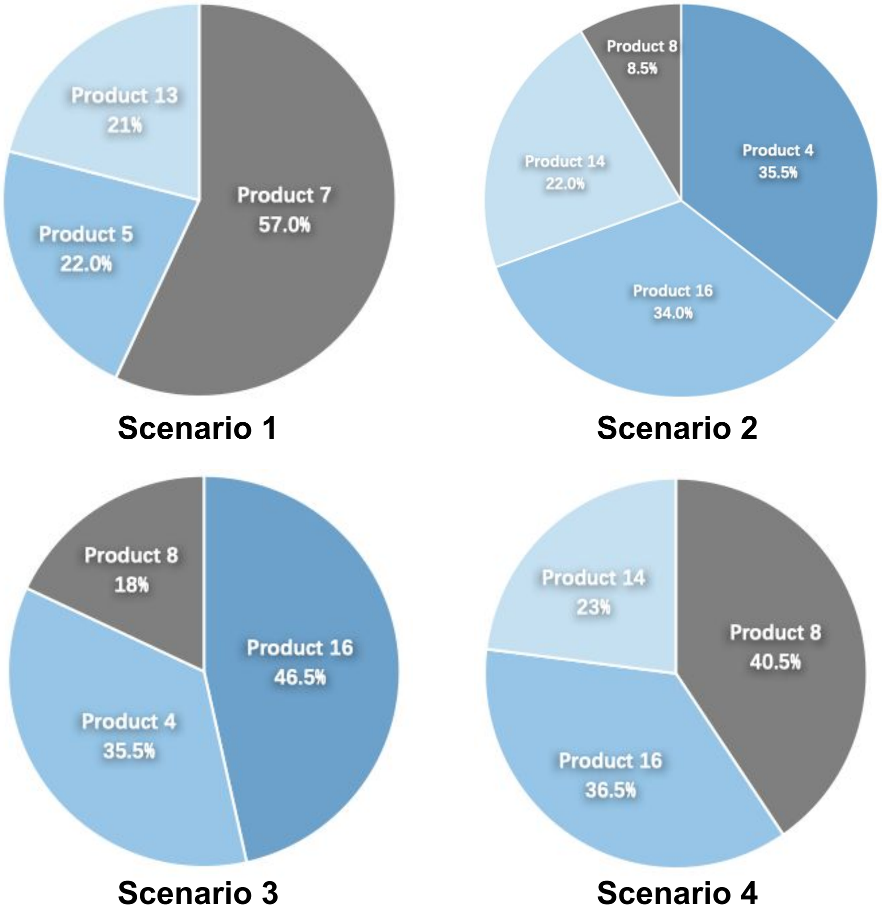
Simulate **market share** and **profitability** in 4 scenarios, in which we launch product:

(1) **5 and 13** (current market)
(2) **4, 14 and 16** (ideal products from post-hoc analysis)
(3) **4 and 16** (ideal products from priori segmentation)
(4) **14 and 16**

Competitor's response: lower their price to $119.99 to keep market share.

Findings:
In **Scenario 2** with product 4, 14 and 16, we yield the largest market share.

Market Share



Market Simulation

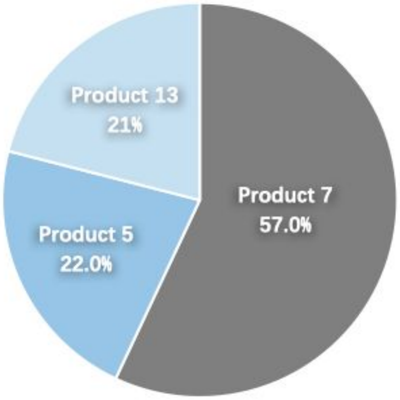
Scenario 1 (current market)

Our Products:

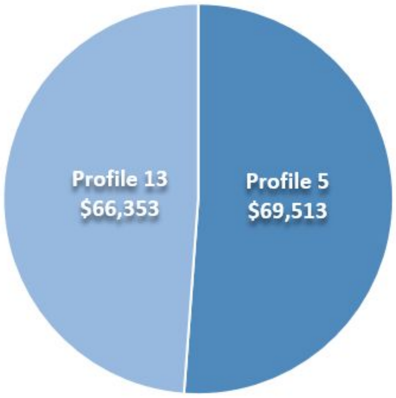
profile 5: 18” Glamorous Rocking Horse at $139.99
profile 13: 18” Racing Rocking Horse at $139.99

Competitor’s Product:

profile 7: 26” Racing Rocking Horse at $139.99

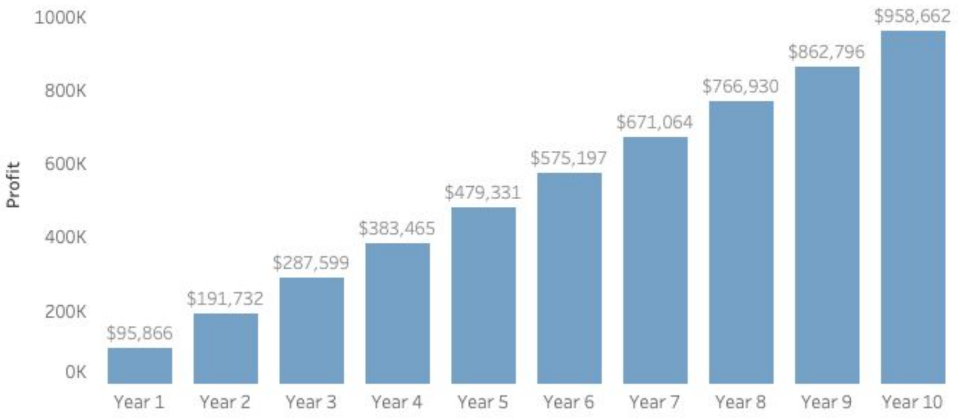


Market Share



Annual Margin

Profitability over 10 years



Findings:

1. Competitor currently takes up the largest proportion of market share (57%).
2. Profile 5 and profile 13 has an annual margin of $69,512 and $66,353 respectively.

Market Simulation

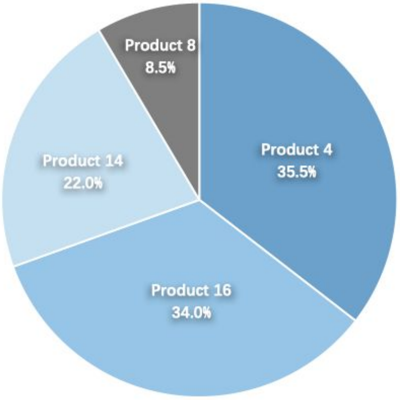
Scenario 2 (ideal products from post-hoc analysis)

Our Products:

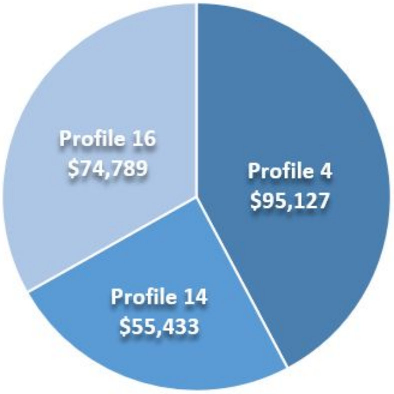
profile 4: 26” Racing Bouncing Horse at $119.99
profile 14: 18” Racing Bouncing Horse at $119.99
profile 16: 18” Glamorous Rocking Horse at $119.99

Competitor’s Product:

profile 8: 26” Racing Rocking Horse at $119.99

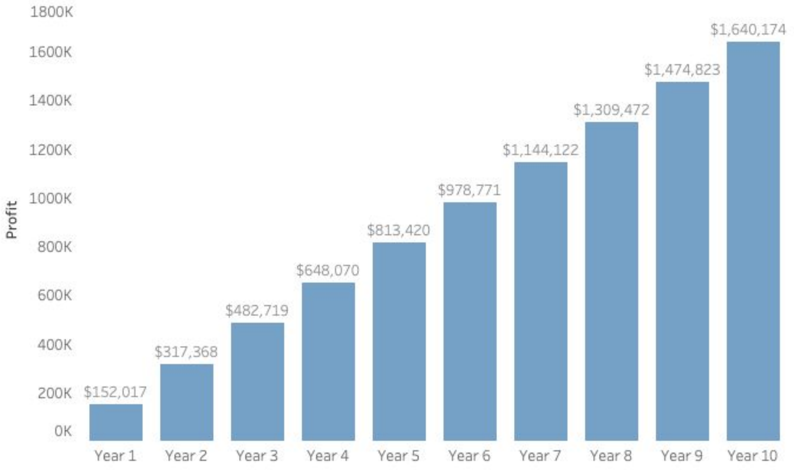


Market Share



Annual Margin

Profitability over 10 years



Findings:

Profile 4 accounts for the largest market share (35.5%), and the greatest proportion of profit annually, with a total margin of $95,127.

Market Simulation

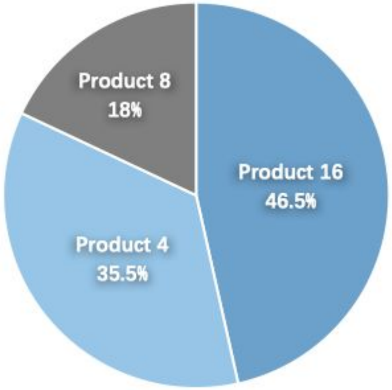
Scenario 3 (ideal products from priori segmentation)

Our Products:

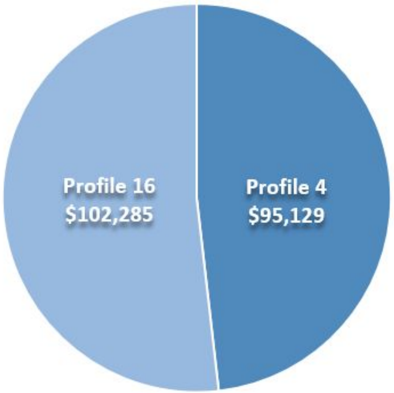
profile 4: 26” Racing Bouncing Horse at $119.99
profile 16: 18” Glamorous Rocking Horse at $119.99

Competitor’s Product:

profile 8: 26” Racing Rocking Horse at $119.99



Market Share



Annual Margin

Profitability over 10 years

Findings:

Profile 16 accounts for the largest market share(46.5%), and the greatest proportion of profit annually, with a total margin of $102,285.

Market Simulation

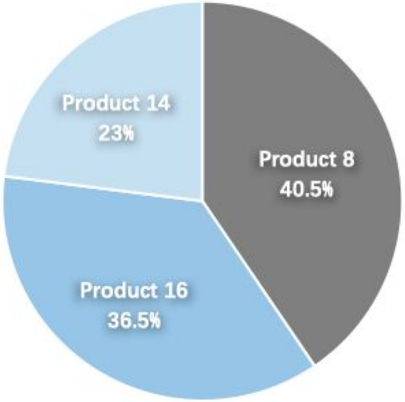
Scenario 4:

Our Products:

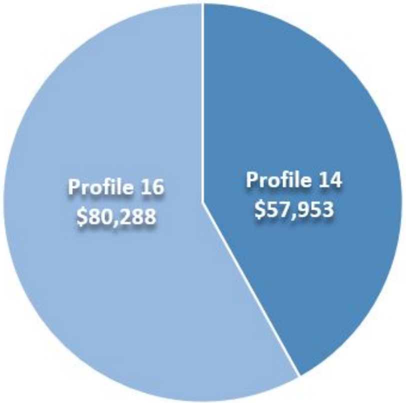
profile 14: 18” Racing Bouncing Horse at $119.99
profile 16: 18” Glamorous Rocking Horse at $119.99

Competitor’s Product:

profile 8: 26” Racing Rocking Horse at $119.99

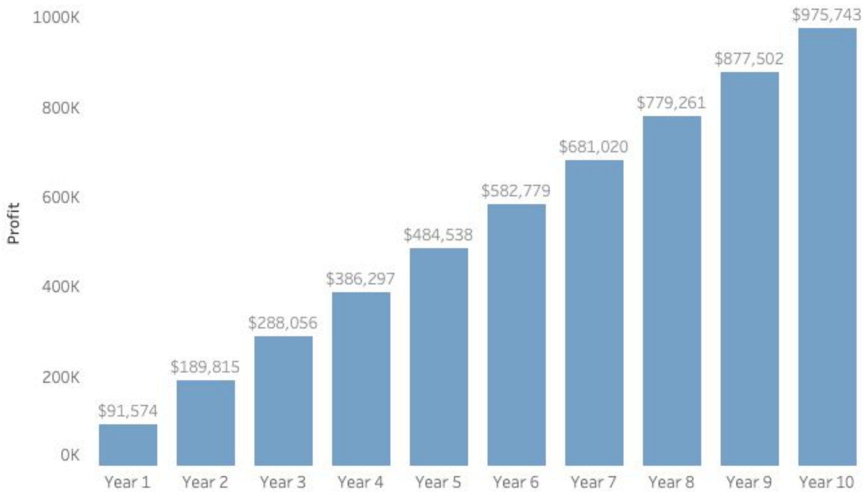


Market Share



Annual Margin

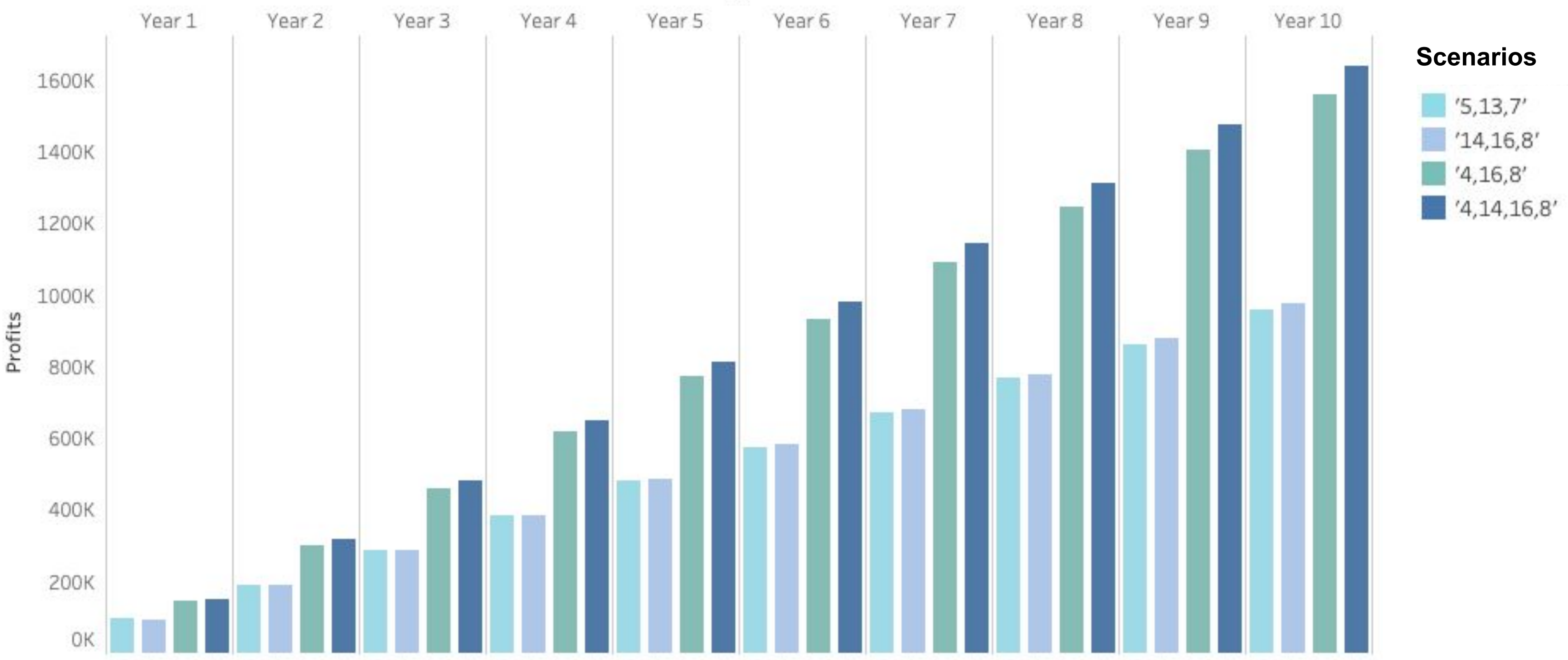
Profitability over 10 years



Findings:

1. Competitor’s product takes up the largest proportion of market share (40.5%).
2. Profile 16 and profile 14 has an annual margin of $80,288 and $57,953 respectively.

Comparison for Profitability



Offering profile 4, 14 and 16 yields the highest profitability!

SUMMARY

Market Segmentation

Priori Segmentation

2 Customer Segments

Preferred Product:

4 &16

Benefit Segmentation

3 Customer Segments

Preferred Product:

4 & 14 & 16

Market Simulation

Scenario Combination

Competitor Response

Market Share

Profitability

(short-run, long-run)

Recommended Product Line

Most Profitable Scenario

Competitor: 8

Our Products:

4 & 14 & 16

$119.99
26” Racing Bouncing Horse
$119.99
18” Racing Bouncing Horse
$119.99
18” Glamorous Rocking Horse

GBA424: Analytics Design - Assignment 3

Pin Li, Jiawen Liang, Ruiling Shen, Chenxi Tao, Khanh Tran

2/2/2020

Setup

```
rm(list = ls())

# Set up environment and load datasets
dir="E:/Studying/Simon/Classes/GBA424 - Analytics Design/Assignments/Assignment 3"
setwd(dir)
load("GBA424 - Toy Horse Case Data.Rdata")
require("cluster")
```

```
## Loading required package: cluster
```

```
require("fpc")
```

```
## Loading required package: fpc

## Warning: package 'fpc' was built under R version 3.6.2
```

```
require("factoextra")
```

```
## Loading required package: factoextra

## Warning: package 'factoextra' was built under R version 3.6.2

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.6.2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
require("gridExtra")
```

```
## Loading required package: gridExtra

## Warning: package 'gridExtra' was built under R version 3.6.2
```

```
library(cluster)
library(fpc)
library(factoextra)
library(gridExtra)
library(reshape)
```

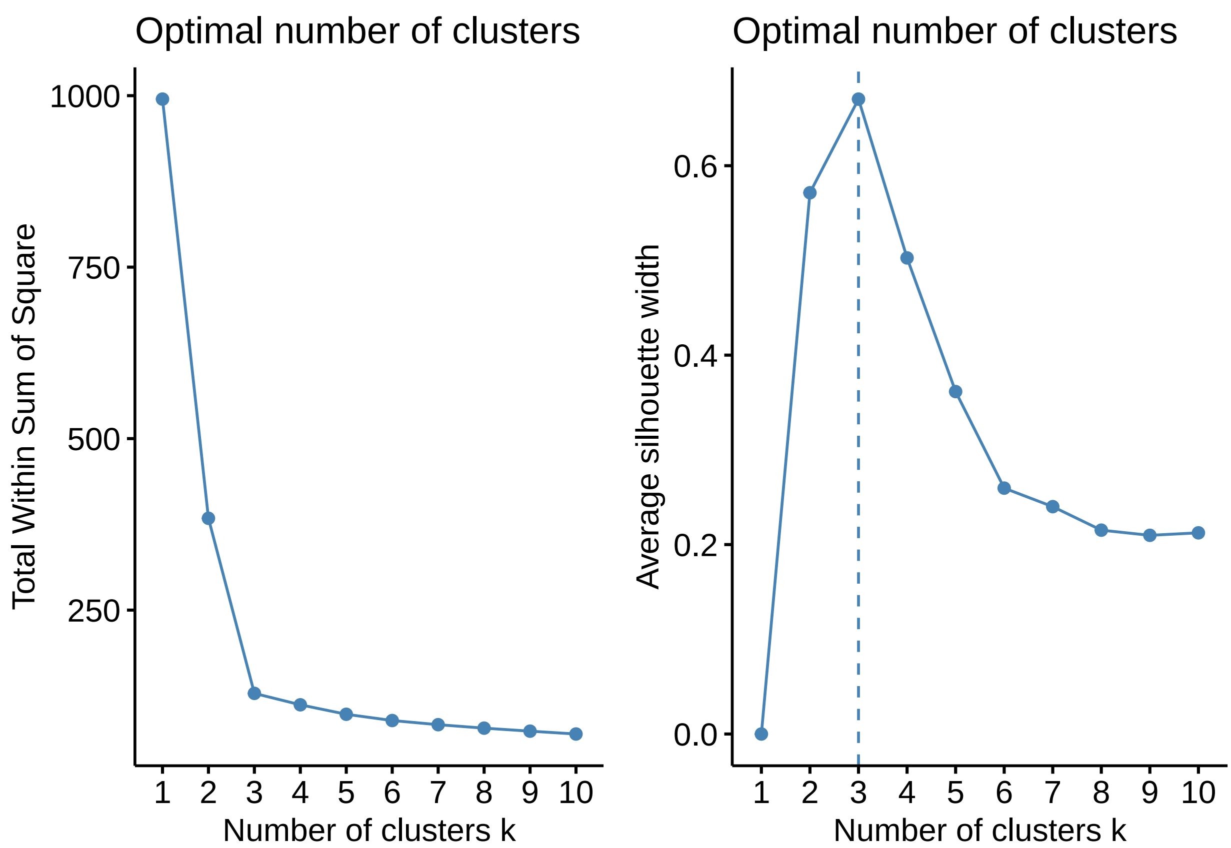
Part A

```r
##################################
#            PART A             #
##################################

## Use regression to estimate the conjoint model at the individual level

# Create new dataset for part A
indi_data = conjointData

# Subset the data to train the model
indi_training = indi_data[!(is.na(indi_data$ratings)),]
indi_missing = indi_data[is.na(indi_data$ratings),]

# Store the coefficients
numIDs = length(unique(conjointData$ID)) # Number of respondents
partworths1 = data.frame(ID = 1:numIDs, intercept = NA, price = NA,
                         size = NA, motion = NA, style = NA)
indi_pred = list() # List that saves predicted ratings of missing profiles
for (num in 1:numIDs){
  data.training.sub = subset(indi_training, ID == num)
  data.missing.sub = subset(indi_missing, ID == num)
  lm = lm(ratings~price+size+motion+style,data = data.training.sub)
  partworths1[num, 2:6] = lm$coefficients
  indi_pred = append(indi_pred,predict(lm,data.missing.sub))
}

# Replace NA ratings with predicted ratings for missing profiles
indi_data$ratings[is.na(indi_data$ratings)] = unlist(indi_pred)
```

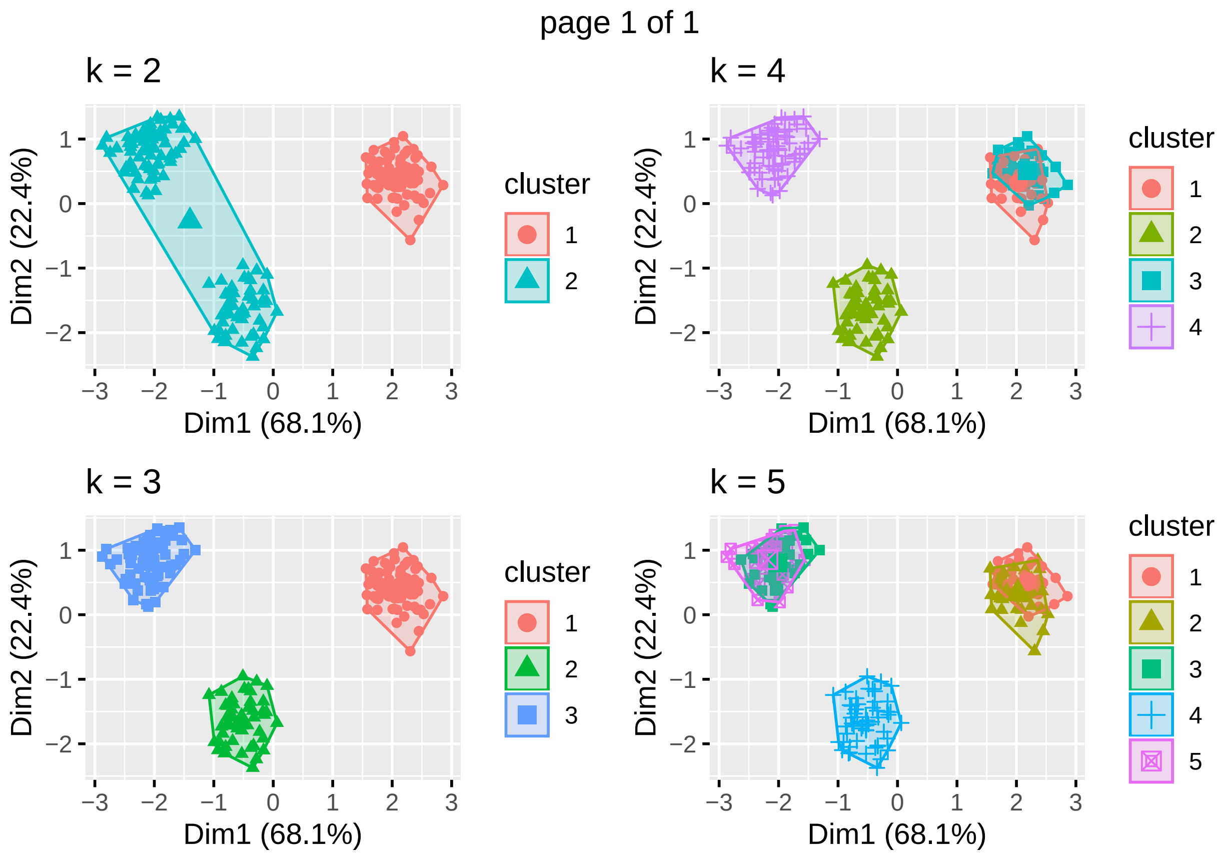
Part-utilities of the conjoint model at the individual level are stored in the 'partworths1'. The NAs in the survey data are replaced by the predictions for missing profiles.

Part B

```r
##################################
#            PART B             #
##################################
source("ConjointCode.R")

## Evaluate number of clusters to use on data with visualizations
checkClust = clustTest(partworths1[,2:6],print=TRUE,scale=TRUE,maxClusts=10,
                       seed=12345,nstart=20,iter.max=100)
```

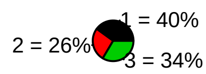


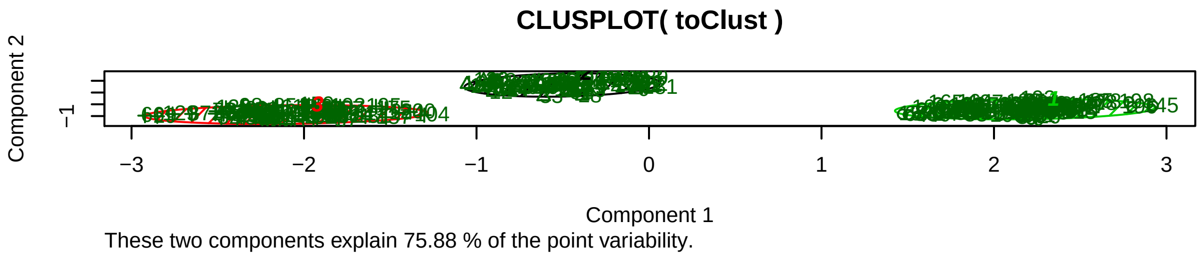

```
clusts = runClusts(partworths1[,2:6],c(2,3,4,5),print=TRUE,maxClusts=4,
                   seed=12345,nstart=20,iter.max=100)
```

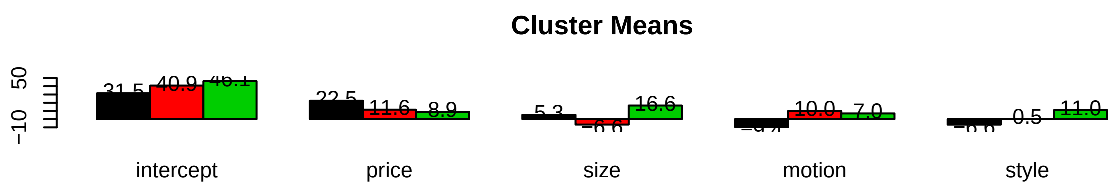


The optimal number of clusters is 3, where the average silhouette width is the highest, and the customer can be separated in to 3 non-overlapped groups of people with different preferences.

```
## Plot clusters with nClusters = 3
plotClust(clusts[[1]][[2]],partworths1)
```








```r
# Cluster means
partworths1_seg = as.data.frame(clusts[[1]][[2]]$centers)
partworths1_seg
```

```
##   intercept     price      size    motion      style
## 1  31.50291 22.474947  5.348628 -9.434120 -6.6285255
## 2  40.88116 11.564861 -6.561319 10.038882  0.4726337
## 3  46.13706  8.938311 16.593109  6.965209 10.9539010
```

In the post-hoc segmentation, we use 3 clusters. The sign and magnitude of attribute coefficients indicate the preference of consumers within certain attributes, with positive sign meaning consumers prefer that attribute. We can use this result to support our product line decision.

Ideal product for each segment
Segment 1: prefer lower price, bigger size, bouncing motion and racing style. → Profile 4
Segment 2: preder lower price, smaller size, rocking motion and glamour style. → Profile 14
Segment 3: preder lower price, bigger size, rocking motion and glamour style. → Profile 16

Part C

```r
################################
#           PART C             #
################################
```

```r
# Create new dataset for part C
seg_data = conjointData

## Conduct a priori segmentation using the variabkes gender and age
demo = as.data.frame(lapply(respondentData,as.factor)) # demographic info

# Create 3 segmentations by age, by gender, and by age & gender
demo_seg1 = kmeans(x=demo[,2:3], centers = 4, nstart = 1000) # age & gender (4 clusters)
demo_seg2 = kmeans(x=demo[,2], centers = 2, nstart = 1000) # age (2 clusters)
demo_seg3 = kmeans(x=demo[,3], centers = 2, nstart = 1000) # gender (2 clusters)

# Merge cluster id with the original conjoint data
cluster_id = data.frame(ID = demo$ID,
                        seg1=factor(demo_seg1$cluster),
                        seg2=factor(demo_seg2$cluster),
                        seg3=factor(demo_seg3$cluster))
seg_data = merge(seg_data, cluster_id,by = "ID", all.x = T)

# Subset training and missing data
seg_training = seg_data[!(is.na(seg_data$ratings)),]
seg_missing = seg_data[is.na(seg_data$ratings),]
```

To test whether priori segmentations affect part-utilities, we run regressions with interactions of the segment dummies with each attribute.

Segmentation 1: By age and gender

```r
# Segmentation 1 by age and gender
summary(lm(ratings~price+size+motion+style+
             price*seg1+size*seg1+
             motion*seg1+style*seg1,
           data=seg_training))[[4]]
```

```
##                 Estimate Std. Error    t value      Pr(>|t|)
## (Intercept)  40.3828391   1.310593 30.8126462 9.855947e-176
## price        13.6446118   1.228081 11.1105104  5.374151e-28
## size          9.4914412   1.175798  8.0723367  1.083173e-15
## motion        2.0536536   1.175798  1.7466034  8.083515e-02
## style         3.8353065   1.175798  3.2618740  1.122419e-03
## seg12        -5.3905360   2.082565 -2.5884120  9.700640e-03
## seg13        -2.6048124   1.931992 -1.3482519  1.777057e-01
## seg14         1.1196003   1.986693  0.5635497  5.731137e-01
## price:seg12   5.1194739   1.951452  2.6234178  8.760826e-03
## price:seg13   1.7460060   1.810359  0.9644529  3.349169e-01
## price:seg14  -0.3126248   1.861616 -0.1679319  8.666511e-01
## size:seg12   -3.7054922   1.868373 -1.9832722  4.745179e-02
## size:seg13   -7.1289830   1.733287 -4.1129851  4.038368e-05
## size:seg14   -3.9889956   1.782362 -2.2380394  2.531065e-02
## motion:seg12 -6.0030956   1.868373 -3.2130070  1.331076e-03
## motion:seg13 -0.3604476   1.733287 -0.2079561  8.352810e-01
## motion:seg14  1.9603507   1.782362  1.0998613  2.715038e-01
## style:seg12  -6.9041865   1.868373 -3.6952935  2.245875e-04
## style:seg13  -4.8176418   1.733287 -2.7794833  5.487330e-03
## style:seg14  -0.2488841   1.782362 -0.1396373  8.889584e-01
```

The interaction coefficients between segmentations and attributes are not entirely significant, so we consider testing whether gender or age is meaningful for business segmentation.

Segmentation 2: By age

```r
# Segmentation 2 by age
summary(lm(ratings~price+size+motion+style+
             price*seg2+size*seg2+
             motion*seg2+style*seg2,
           data=seg_training))[[4]]
```

```
##                 Estimate Std. Error     t value      Pr(>|t|)
## (Intercept)  39.54618223  1.0867243 36.39026357 4.223074e-231
## price        14.41328802  1.0183069 14.15416914  9.781338e-44
## size          3.85315922  0.9749546  3.95214210  7.970172e-05
## motion        2.79499918  0.9749546  2.86679924  4.182755e-03
## style         1.18667087  0.9749546  1.21715497  2.236654e-01
## seg22        -1.29820883  1.5292330 -0.84892805  3.960063e-01
## price:seg22   1.25883815  1.4329565  0.87849016  3.797661e-01
## size:seg22    4.17076027  1.3719514  3.04002051  2.391276e-03
## motion:seg22 -3.11880912  1.3719514 -2.27326508  2.309859e-02
## style:seg22  -0.08569567  1.3719514 -0.06246261  9.501997e-01
```

The segmentation here only affects part-utilities of size attribute.

Segmentation 3: By gender

```r
# Segmentation 3 by gender
summary(lm(ratings~price+size+motion+style+
             price*seg3+size*seg3+
             motion*seg3+style*seg3,
           data=seg_training))[[4]]
```

```
##                Estimate Std. Error    t value      Pr(>|t|)
## (Intercept)  36.5668426  1.0739178 34.0499449 1.606196e-207
## price        16.8573429  1.0063067 16.7516948  1.215027e-59
## size          3.8509324  0.9634653  3.9969600  6.610460e-05
## motion       -0.7601191  0.9634653 -0.7889429  4.302236e-01
## style        -1.8895287  0.9634653 -1.9611797  4.997401e-02
## seg32         4.3032300  1.4614170  2.9445599  3.265352e-03
## price:seg32  -3.3487809  1.3694100 -2.4454188  1.454011e-02
## size:seg32    3.9045569  1.3111102  2.9780539  2.930023e-03
## motion:seg32  3.6668883  1.3111102  2.7967811  5.202755e-03
## style:seg32   5.6165245  1.3111102  4.2837927  1.910028e-05
```

The interation coefficients are significant in segmentations by gender.

Conclusion
From the significant effect of gender segmentation to all the four attributes, we can safely conclude that gender is the most meaningful factor to use for a priori segmentation. Meanwhile, age does not play such an important role as gender with insignificant effects to price and style.

We will use only **gender** to do priori demographic segmentation.

```r
## Segment-level regressions
partworths2_seg = data.frame(cluster = 1:2, intercept = NA, price = NA,
                             size = NA, motion = NA, style = NA)
for (seg in 1:2){
  data.sub = subset(seg_training, seg3 == seg)
  lm = lm(ratings~price+size+motion+style, data=data.sub)
  partworths2_seg[seg, 2:6] = lm$coefficients
}
partworths2_seg
```

```
##   cluster intercept    price     size     motion     style
## 1       1  36.56684 16.85734 3.850932 -0.7601191 -1.889529
## 2       2  40.87007 13.50856 7.755489  2.9067692  3.726996
```

We'll only get 2 sets of part-utilities instead of 200. But at least one set of part-utilities for attributes varies significantly across segments, and can be used for target different optimal products.

Ideal product for each segment
Segment 1: prefer lower price, bigger size, bouncing motion and racing style. → Profile 4
Segment 2: preder lower price, bigger size, rocking motion and glamour style. → Profile 16

Part D

```r
#################################
#           PART D              #
#################################

# Prepare data for analysis
ratingData = cast(indi_data, ID ~ profile, value="ratings")
ratingData = ratingData[, -1] # Remove the ID column

# Function to calculate market share and deal with tie decisions
simFCSharesTie = function(scen,data,ascend=FALSE){
    inmkt = data[,scen]
    if(ascend){
    bestOpts = apply(inmkt,1,min)
    } else {
    bestOpts = apply(inmkt,1,max)
    }

    decisions = inmkt == bestOpts
    decisionsTie = decisions / rowSums(decisions)
    mkShare = colSums(decisionsTie)/sum(decisionsTie)
    mkShare
}
```

Set up scenarios

Our current products' profile IDs are 5 and 13, and the competitor's profile ID is 7. We will simulate the scenarios in which we launch ideal products from part B and part C, considering the competitor's reponse by reducing his price.

Senarios:

Scenario	Our Products	Competitor's Product
1 (Original)	5, 13	7
2 (Part B)	4, 14, 16	7
3 (Part B)	4, 14, 16	8
4 (Part C)	4, 16	7
5 (Part C)	4, 16	8
6	14, 16	7
7	14, 16	8

```r
## Set up scenarios
scens = list()
scens[[1]]=c(5,13,7)
scens[[2]]=c(4,14,16,7)
scens[[3]]=c(4,14,16,8)
scens[[4]]=c(4,16,7)
scens[[5]]=c(4,16,8)
scens[[6]]=c(14,16,7)
scens[[7]]=c(14,16,8)

## Market Share
sapply(scens,simFCSharesTie,data=ratingData, ascend=FALSE)
```

```
## [[1]]
##    5   13    7
## 0.22 0.21 0.57
##
## [[2]]
##    4   14   16    7
## 0.40 0.25 0.35 0.00
##
## [[3]]
##     4    14    16     8
## 0.355 0.220 0.340 0.085
##
## [[4]]
##     4    16     7
## 0.405 0.595 0.000
##
## [[5]]
##     4    16     8
## 0.355 0.465 0.180
##
## [[6]]
##    14    16     7
## 0.300 0.695 0.005
##
## [[7]]
##    14    16     8
## 0.230 0.365 0.405
```

In scenario 2, 4, 6, the competitor's share decreases tremendously so we assume that he will decrease his price in response (i.e., changing from profile 7 to profile 8). Hence we remove these scenarios and move forward with scenario 1, 3, 5, 7. After simulating the market share, we will simulate short-term and long-term profitability.

```
## Simulate profitability
# Variable cost
variableCost = profilesData
variableCost$varCost[variableCost$size==0 & variableCost$motion==1] = 33 # 18" Rocking
variableCost$varCost[variableCost$size==1 & variableCost$motion==1] = 41 # 26" Rocking
variableCost$varCost[variableCost$size==0 & variableCost$motion==0] = 21 # 18" Bouncing
variableCost$varCost[variableCost$size==1 & variableCost$motion==0] = 29 # 26" Bouncing

# Function to calculate profitability over years
profitFunc = function(scen, data, year=1) {
    marketShares = simFCSharesTie(scen, data, ascend=FALSE)

    ourProducts = scen[-length(scen)] # exclude competitor's share
    ourMarketShare = marketShares[1:length(ourProducts)]

    quantity = ourMarketShare*4000
    price = profilesData$priceLabel[profilesData$profile %in% ourProducts]*100/125
    varCost = variableCost$varCost[variableCost$profile %in% ourProducts]
    fixCost = 20000*length(ourProducts)*year +
      sum(!(ourProducts %in% c(5, 13, 6, 14)))*1/3*20000

    margin = (price-varCost)*quantity
    profit = sum(margin)*year - fixCost
    results = list(profit, margin)
    results
}
```

First we calculate annual margin for each product in each scenario.

```
# Annual margin for each product
productMargin = lapply(scens[c(1, 3, 5, 7)],
                       function (x) profitFunc(x,
                                               data=ratingData,
                                               year=1)[[2]])
productMargin
```

```
## [[1]]
##        5       13
## 69512.96 66353.28
##
## [[2]]
##        4       14       16
## 95128.64 55432.96 74789.12
##
## [[3]]
##         4        16
##  95128.64 102285.12
##
## [[4]]
##       14       16
## 57952.64 80288.32
```

Then we look at overall profitability of the company over years.

```r
# Calculate overall profit
profitData = matrix(nrow=10, ncol=4)
colnames(profitData) = c("'5,13,7'", "'4,14,16,8'", "'4,16,8'", "'14,16,8'")
rownames(profitData) = paste("Year", 1:10)

for (year in 1:10) {
  profitData[year, ] = sapply(scens[c(1, 3, 5, 7)],
                              function (x) profitFunc(x,
                                                      data=ratingData,
                                                      year=year)[[1]])
}

profitData
```

```
##          '5,13,7' '4,14,16,8'  '4,16,8' '14,16,8'
## Year 1   95866.24    152017.4  144080.4  91574.29
## Year 2  191732.48    317368.1  301494.2 189815.25
## Year 3  287598.72    482718.8  458907.9 288056.21
## Year 4  383464.96    648069.5  616321.7 386297.17
## Year 5  479331.20    813420.3  773735.5 484538.13
## Year 6  575197.44    978771.0  931149.2 582779.09
## Year 7  671063.68   1144121.7 1088563.0 681020.05
## Year 8  766929.92   1309472.4 1245976.7 779261.01
## Year 9  862796.16   1474823.1 1403390.5 877501.97
## Year 10 958662.40   1640173.9 1560804.3 975742.93
```

Scenario 3 (2nd column), in which we sell profile 4, 14, 16 and the competitor sell profile 8, yields the highest profit both in short term and long term.

```r
apply(profitData, 1, which.max)
```

```
##  Year 1  Year 2  Year 3  Year 4  Year 5  Year 6  Year 7  Year 8  Year 9 
##       2       2       2       2       2       2       2       2       2 
## Year 10 
##       2 
```