

Toy Horse Main File

MSBA In_Person Team D

2020/12/7

```
library(cluster)
library(fpc)
library(foreign)
library(factoextra)

## Loading required package: ggplot2
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
library(gridExtra)
library(data.table)
library(reshape)

##
## Attaching package: 'reshape'

## The following object is masked from 'package:data.table':
##
##      melt

#setwd("C:/Users/USER/Desktop")
load("GBA424 Fall 2020 - Toy Horse Case Data.Rdata")
source("Functions.R")
df = conjointData[!is.na(conjointData$ratings),]
```

Part A: Regressions

```
#Do the regression on individual basis

coefficient = matrix(nrow=200,ncol=5)
for (i in 1:200){

  coefficient[i,] =lm(ratings~price+size+motion+style,data =df[df$ID==i,])$coefficients

}
colname = c("intercept",names(conjointData)[c(4:7)])
coefficient = data.table(coefficient)
names(coefficient) = colname
coefficient = as.matrix(coefficient)

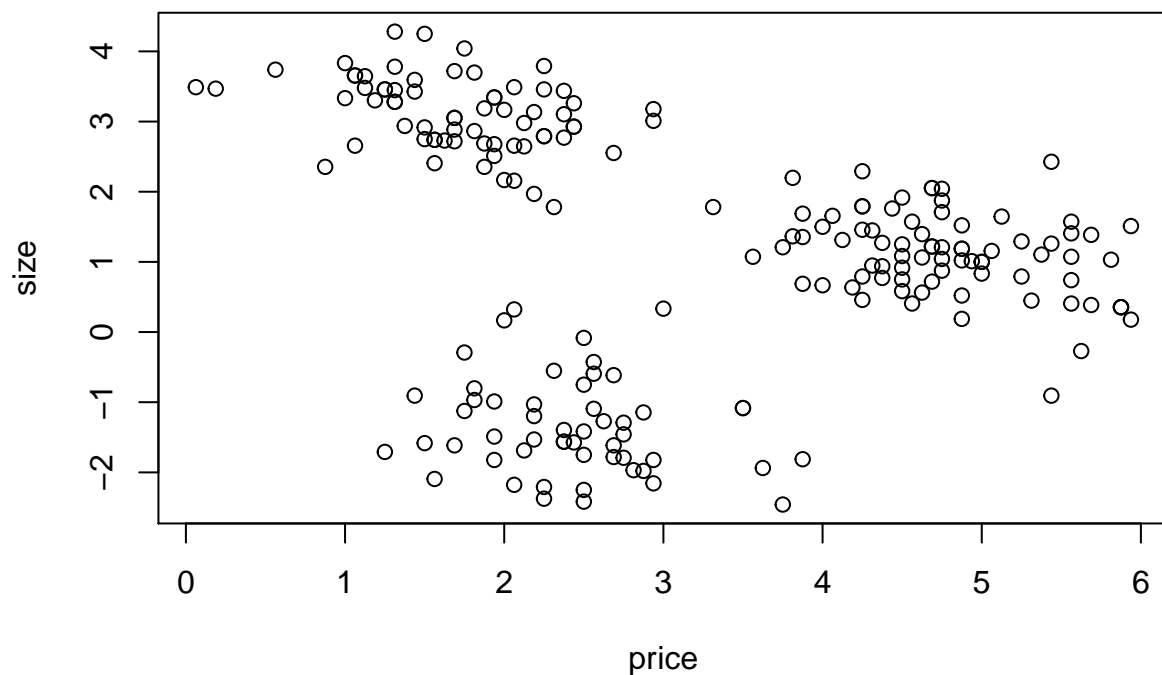
#predict the rating by coefficient
predict = apply(cbind(1,conjointData[,c("price","size","motion","style")])*matrix(rep(coefficient,each =
```

```
#Input the Missing rating
conjointData$ratings = ifelse(is.na(conjointData$ratings),predict,conjointData$rating)
```

Part B: Post-hoc Segmentation

```
# Use plot function to see whether any pattern inside the coefficients

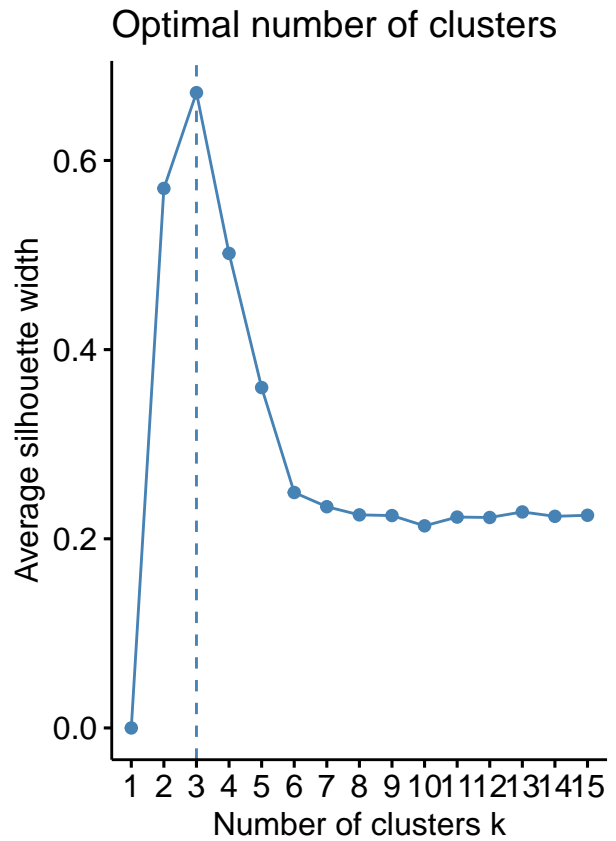
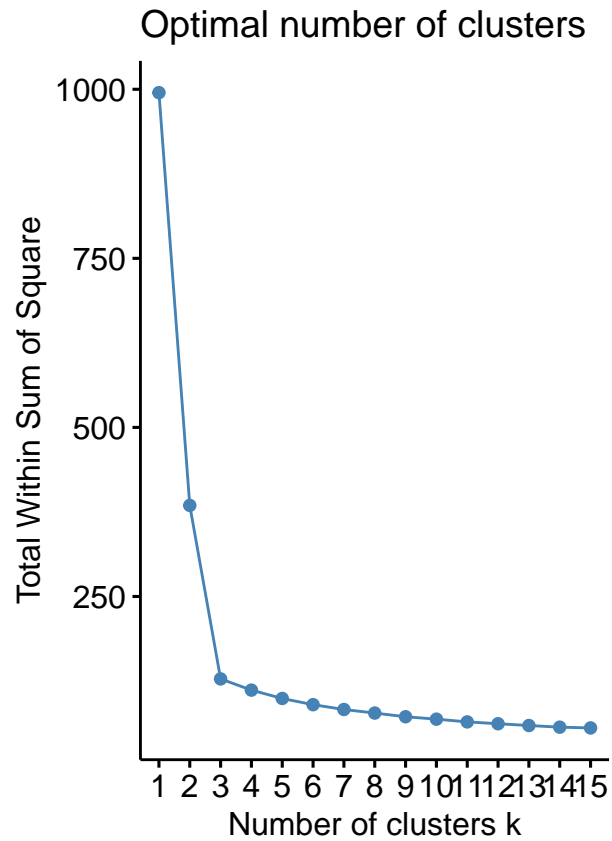
plot(coefficient[, -1])
```



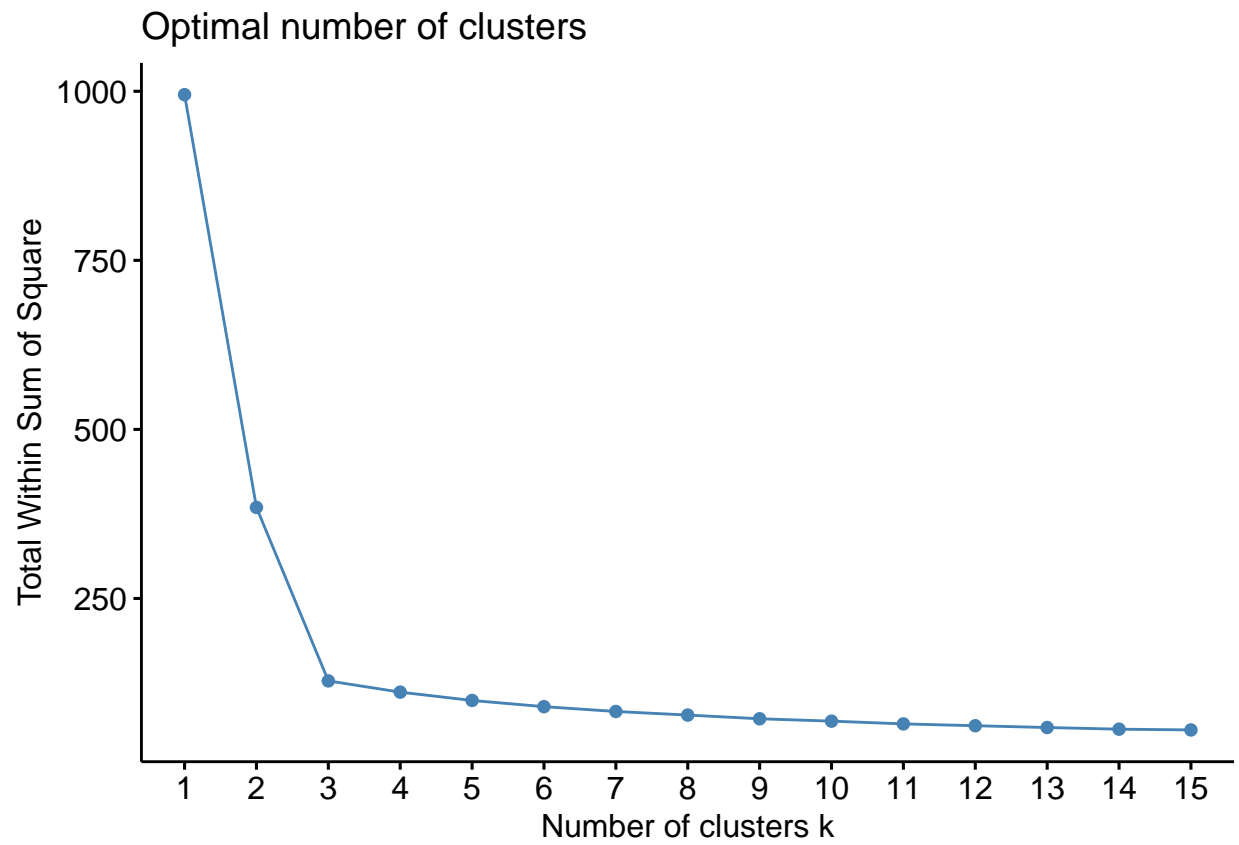
```
#set the seed for k-mean cluster
seed = 12464

#Please refer to the functions R script for the function we used below

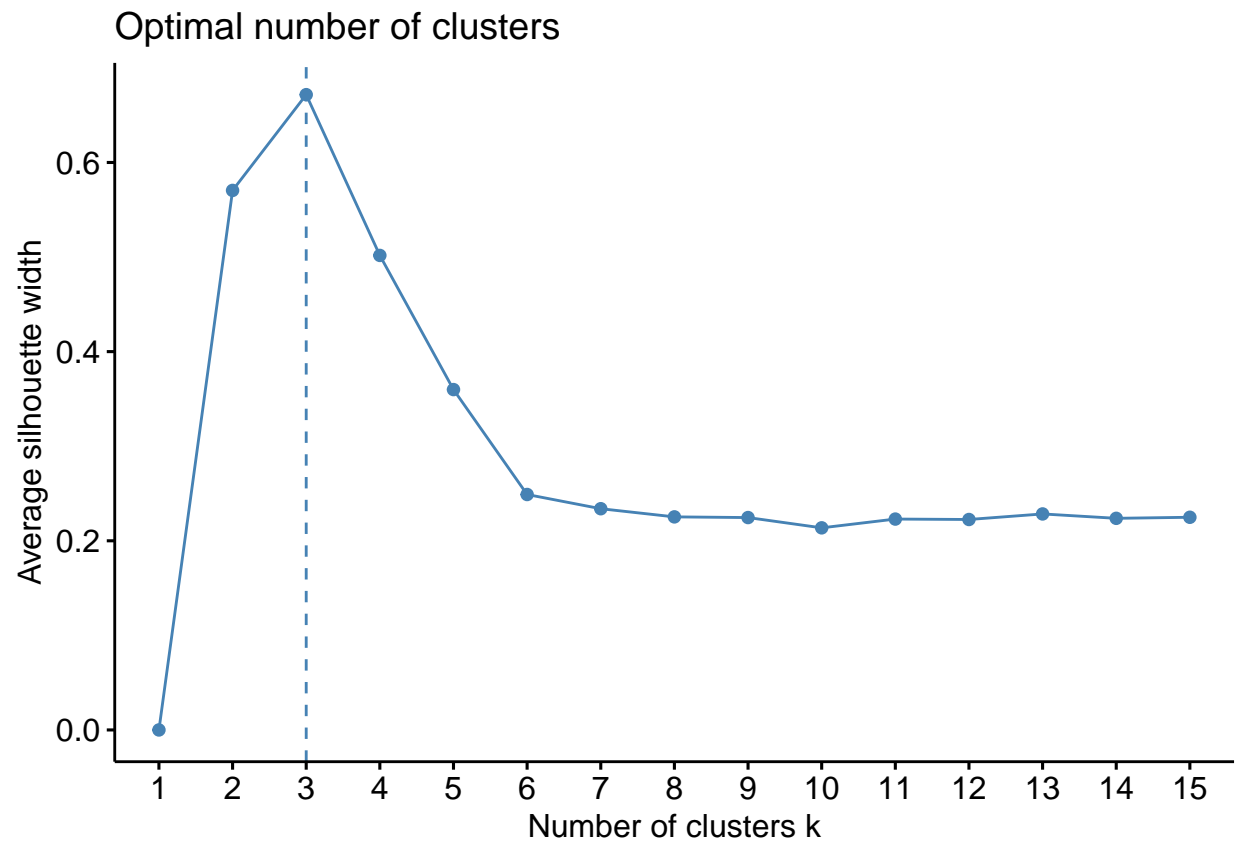
# Test on the cluster number
clustTest (coefficient)
```



```
## $wss
## [1] 995.00000 384.65741 128.13460 111.54654 99.20436 90.02812 82.87987
## [8] 77.43612 72.30745 69.14349 64.98986 61.80101 59.63123 56.83070
## [15] 54.13261
##
## $pm1
## [1] 3
##
## $gpw
```

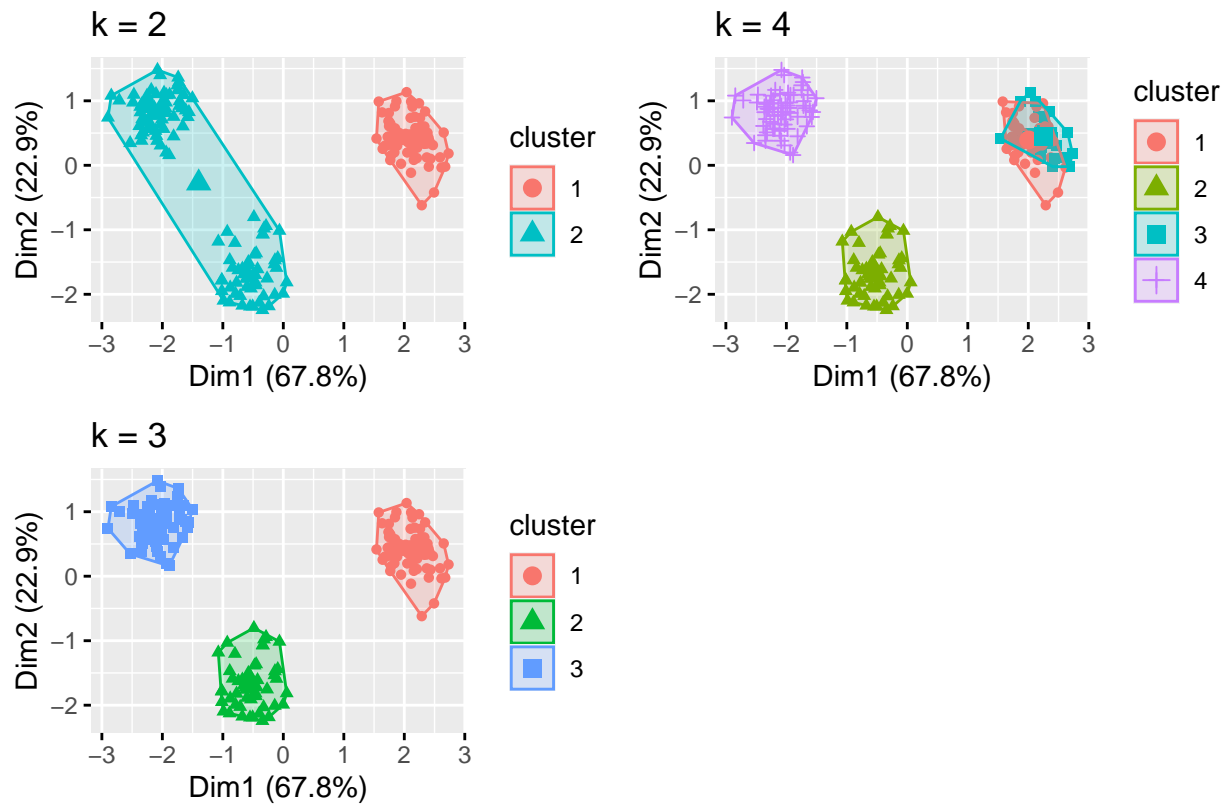


\$gps



The result suggests that the best number of cluster would be 3. We will use 2 to 4 clusters to find t

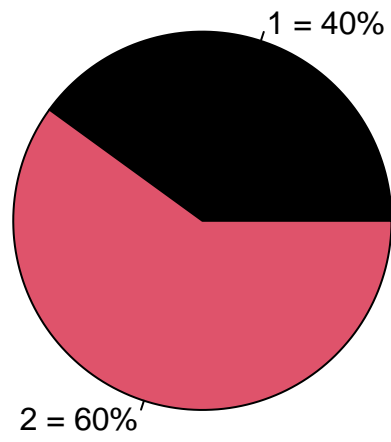
```
clusters = runClusts(coefficient, 2:4)
```

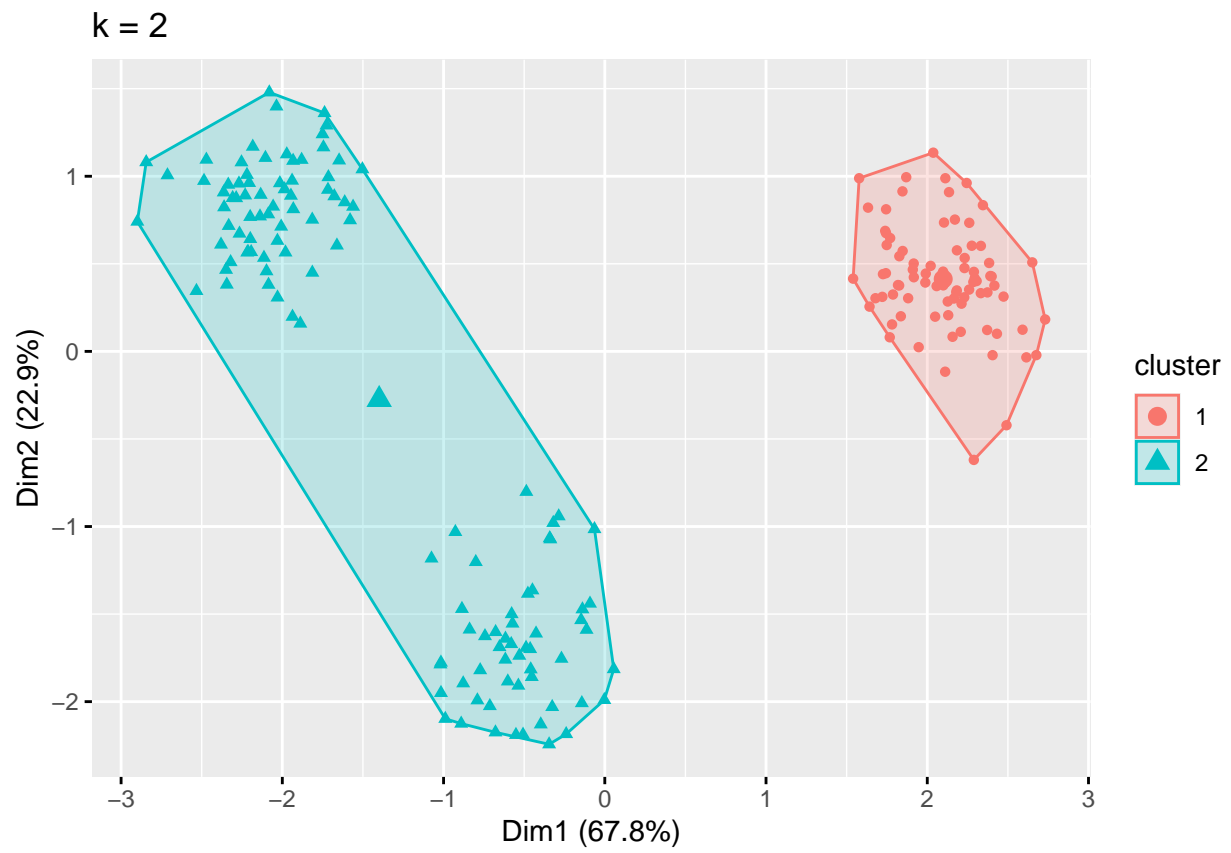


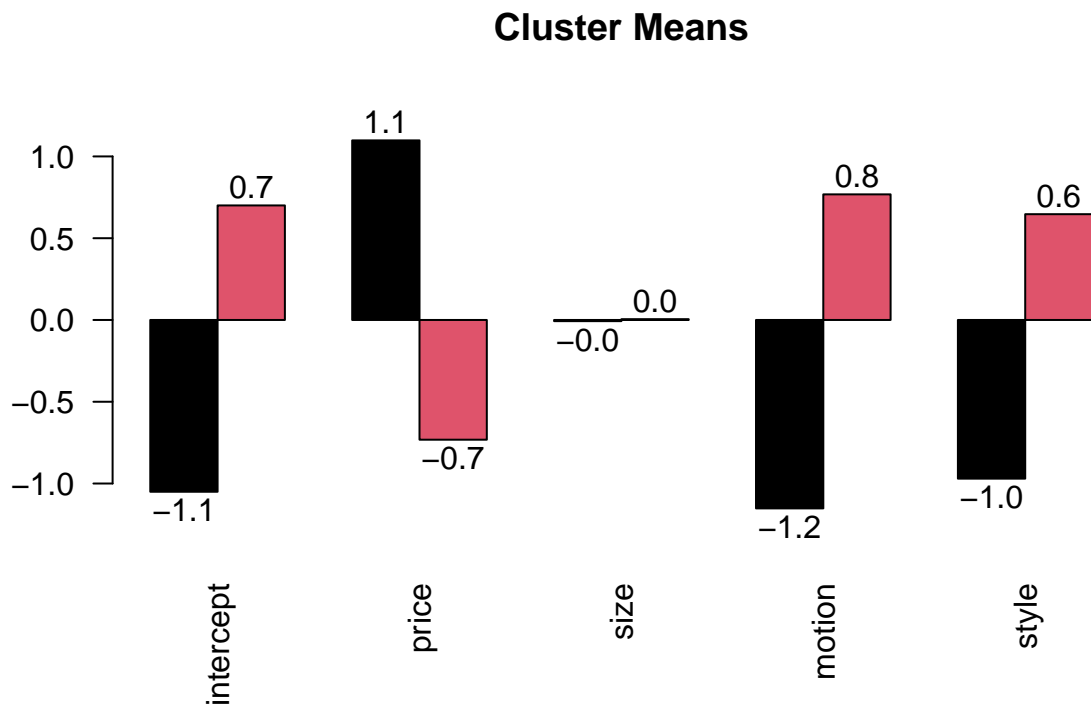
#Trying 2 to 4 clusters

From the graph, we can see that the 2 cluster model has one group with very large space. In the 4 cluster model, the groups are more compact.

```
plotClust(clusters$kms[[1]], coefficient)#2 clusters
```

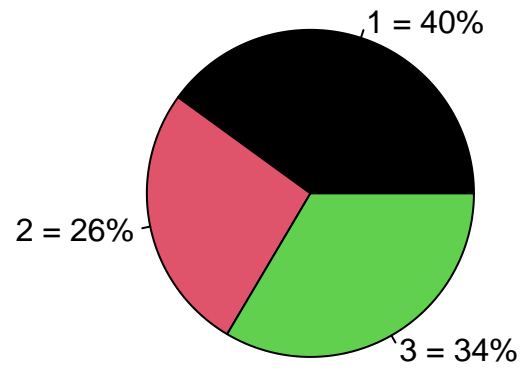


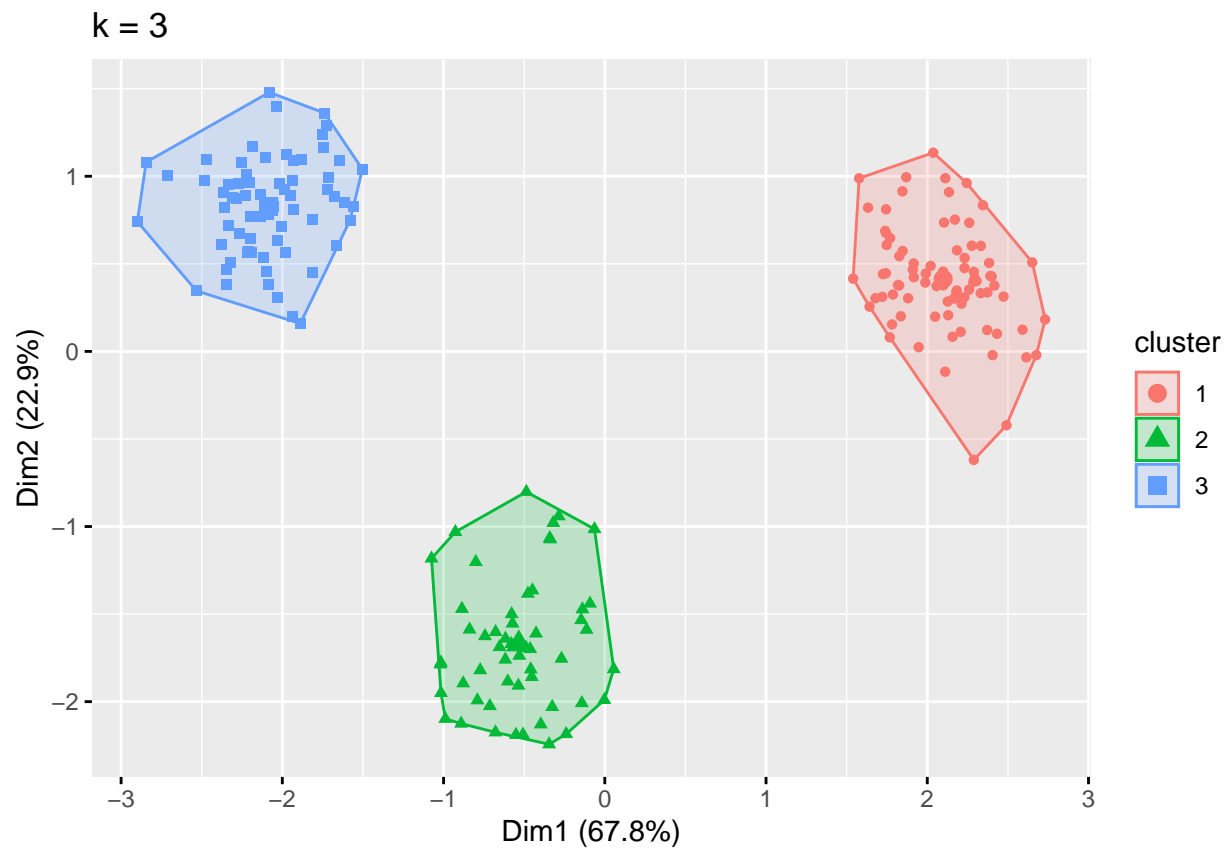


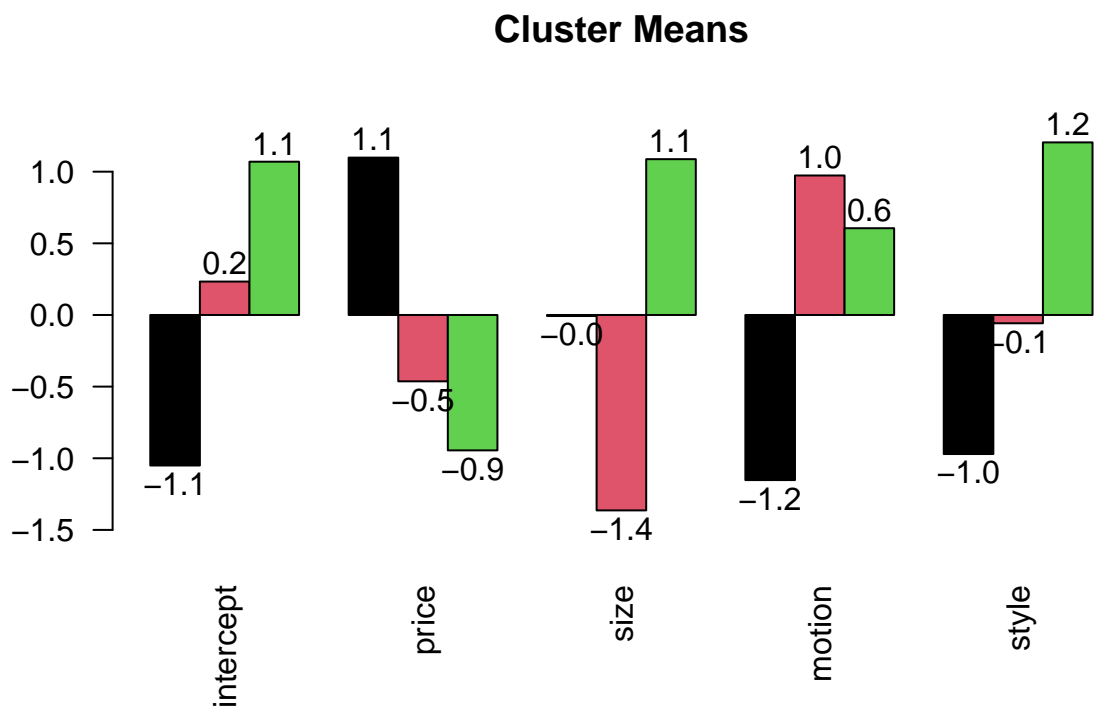


```
##      Segment intercept    price    size    motion    style Size
## 1:      2  7.900174  2.030208  1.140799  1.686979  1.167535  0.6
## 2:      1  5.132031  4.760937  1.119010 -1.969010 -1.360677  0.4
```

```
plotClust(clusters$kms[[2]], coefficient) #3 clusters- preferred
```

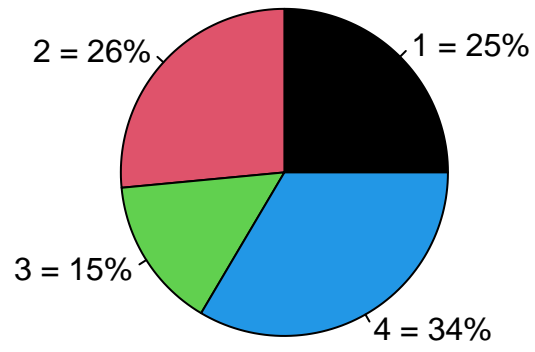


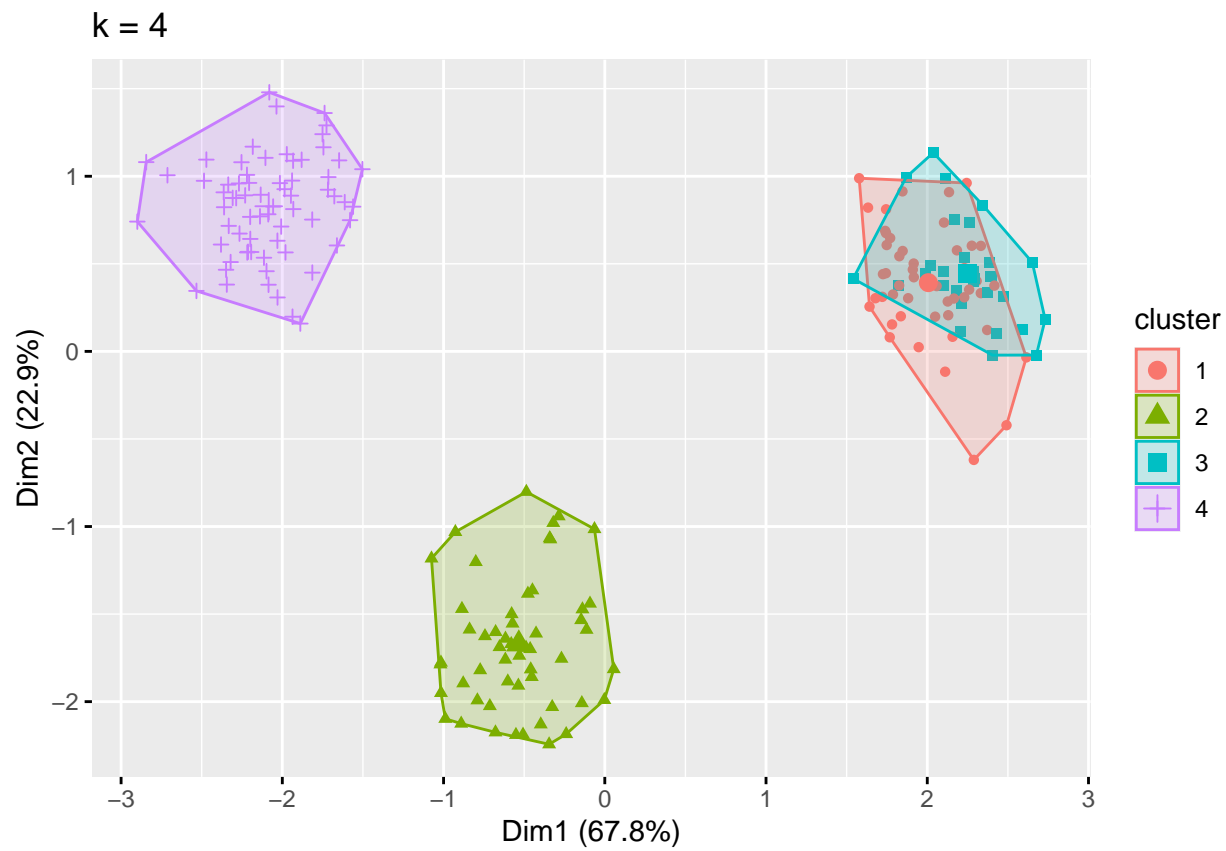




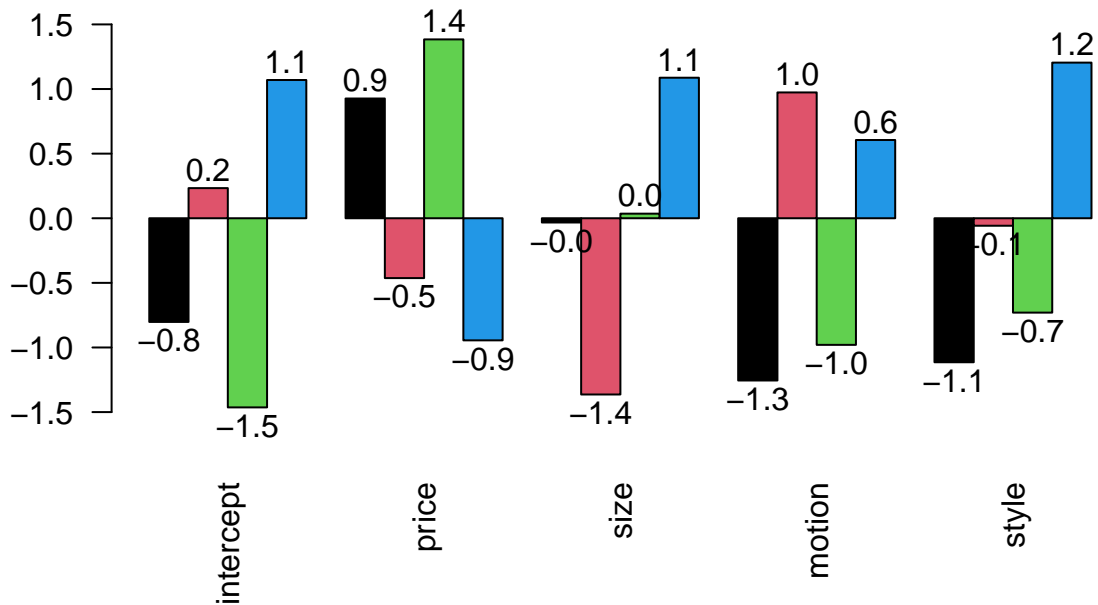
```
##      Segment intercept    price      size    motion      style Size
## 1:         1  5.132031  4.760937  1.119010 -1.969010 -1.36067708 0.400
## 2:         3  8.484453  1.712687  3.110075  1.377488  2.03917910 0.335
## 3:         2  7.161557  2.431604 -1.348664  2.078223  0.06564465 0.265
```

```
plotClust(clusters$kms[[3]], coefficient)#4 clusters
```





Cluster Means



```
##      Segment intercept    price    size    motion    style Size
## 1:      4  8.484453 1.712687  3.110075  1.377488  2.03917910 0.335
## 2:      2  7.161557 2.431604 -1.348664  2.078223  0.06564465 0.265
## 3:      1  5.524167 4.505000  1.072500 -2.165833 -1.58583333 0.250
## 4:      3  4.478472 5.187500  1.196528 -1.640972 -0.98541667 0.150
```

*# According to cluster analysis, the ideal product for segment 1 may be P4
 # The ideal product for segment 2 would be P16
 # The ideal product for Last might be P14*

```
respondentData$group =unlist(clusters$kms[[2]][1])
respondent= data.table(respondentData)
respondent[,sum(age),group]
```

```
##      group V1
## 1:      3 38
## 2:      2 14
## 3:      1 49
```

```
respondent[,sum(gender),group]
```

```
##      group V1
## 1:      3 53
## 2:      2 24
## 3:      1 31
```

```
respondent[,sum(gender==1 &age==1),group]
```

```
##      group V1
## 1:      3 32
## 2:      2  8
## 3:      1 21
```

```
respondent[,sum(gender==1 &age==0),group]
```

```
##      group V1
## 1:      3 21
## 2:      2 16
## 3:      1 10
```

```
respondent[,sum(gender==0 &age==1),group]
```

```
##      group V1
## 1:      3  6
## 2:      2  6
## 3:      1 28
```

```
respondent[,sum(gender==0 &age==0),group]
```

```
##      group V1
## 1:      3  8
## 2:      2 23
## 3:      1 21
```

##Further analysis on the respondentData inside each segment to obtain the insight for the respondent c

Here we use the coefficient dataset generated from Part(A) to cluster the respondent. The K-mean cluster method was used. To identify number of clusters, clustTest function was used to determine the best fit number of cluster number. Here we have the number equal to 3. Thus, the cluster number 2-4 will be tested to form our clusters.

After that, we use runCluster function to do the cluster analysis. From the chart generated, the cluster of 3 provide best segmentation as 2 clusters model would contain a lot of spaces inside one of the cluster and for 4 cluster model, there are overlapping on cluster on the right top corner of the graph. Thus we'll use that one for further analysis.

To further analysis on the result, plotClust function was used to show the mean of cluster's mean on each attribute and also a scale bar plot. From the plot, we could identify that the segment one would be more price sensitive, and relative more interested in Bouncing and Racing type and relatively indifferent to the size of toy horse. For segment 2, they might be relative indifferent on the style, a bit less price sensitive, but strongly interested in small size and rocking ones. For the segment 3, they would be less price sensitive, and strongly prefer the large size, rocking and Glamour type of toys.

Meanwhile, while we look on the coefficient mean in each category, segment 1 have positive coefficient mean in price and size and negative in motion and style. Thus, the rate could be maximize by apply **P4** into this segment. For segment2, except the size, all other coefficients are positive, do the **P14** would be suitable for this segment. For segment 3, all coefficient is positive, so the rate could be maximize by apply **P16**.

Also, we want to know the distribution of respondentData inside each segment. We extract the data and then load into data table. After looking and the 4 condition, we conclude that for segment 1, it contains most of older boys. For segment 3, it contains most of older girls. For the segment 2, even it do not take any major proportion of respondent, most of respondents in that segment is young children.

For those who is interested in rocking and Glamour(mostly in segment 2 and 3), would be less price sensitive relative to other customers(mostly in segment 1). Thus, **for those type of product we could charge a**

bit higher price or not offer the discount. For Bouncing and Racing type of toy horses, their potential buyer is more price sensitive, thus would be better to have lower price or discount to increase buying. Thus, **for those respondent with girl, it might not be a good idea to have price deduction** as segment 2 and 3 were less price sensitive and girls are mostly in this two segment.

Part C: A Priori Segmentation

```
#Merge dataset for gender and Age
conjointData.1 = conjointData
conjointData.1$age = NA

for (x in 1:3200) {
  for (y in 1:200){
    if (conjointData.1$ID[x] == respondentData$ID[y]){
      conjointData.1$age[x] = respondentData$age[y]
    }
  }
}

conjointData.1$gender = NA

for (x in 1:3200) {
  for (y in 1:200){
    if (conjointData.1$ID[x] == respondentData$ID[y]){
      conjointData.1$gender[x] = respondentData$gender[y]
    }
  }
}

#-----

lm1 = summary(lm(ratings~ price + size + motion + style, data = conjointData.1))

#interaction with gender
lm2 = summary(lm(ratings~ (price + size + motion + style)*gender, data = conjointData.1))
lm2

##
## Call:
## lm(formula = ratings ~ (price + size + motion + style) * gender,
##     data = conjointData.1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.8897 -2.1970 -0.0482  2.2153  8.5534
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.2695     0.1791  35.001 < 2e-16 ***
## price          3.4973     0.1602  21.829 < 2e-16 ***
## size           0.7468     0.1602   4.661 3.27e-06 ***
```

```
## motion      -0.1526      0.1602    -0.953    0.34084
## style       -0.4171      0.1602    -2.604    0.00927 **
## gender       0.9693      0.2438     3.977 7.14e-05 ***
## price:gender -0.6940      0.2180    -3.183    0.00147 **
## size:gender  0.7134      0.2180     3.272    0.00108 **
## motion:gender 0.6985      0.2180     3.204    0.00137 **
## style:gender  1.0618      0.2180     4.870 1.17e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.073 on 3190 degrees of freedom
## Multiple R-squared:  0.2874, Adjusted R-squared:  0.2854
## F-statistic: 142.9 on 9 and 3190 DF,  p-value: < 2.2e-16
```

```
#interaction with age
lm3 = summary(lm(ratings~ (price + size + motion + style)*age, data = conjointData.1))
lm3
```

```
##
## Call:
## lm(formula = ratings ~ (price + size + motion + style) * age,
##     data = conjointData.1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.375 -2.457 -0.375  2.438  8.310
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.96149    0.18115  38.430 < 2e-16 ***
## price        2.95581    0.16202  18.243 < 2e-16 ***
## size         0.70770    0.16202   4.368 1.29e-05 ***
## motion       0.52125    0.16202   3.217 0.001307 **
## style        0.18455    0.16202   1.139 0.254759
## age        -0.33381    0.25491  -1.310 0.190449
## price:age    0.33008    0.22800   1.448 0.147782
## size:age     0.84036    0.22800   3.686 0.000232 ***
## motion:age  -0.58747    0.22800  -2.577 0.010021 *
## style:age   -0.05605    0.22800  -0.246 0.805833
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.224 on 3190 degrees of freedom
## Multiple R-squared:  0.2157, Adjusted R-squared:  0.2135
## F-statistic:  97.5 on 9 and 3190 DF,  p-value: < 2.2e-16
```

```
#by gender
lm4 = summary(lm(ratings~ price + size + motion + style, data = conjointData.1, subset=conjointData.1$gender == 1))
lm5 = summary(lm(ratings~ price + size + motion + style, data = conjointData.1, subset=conjointData.1$gender == 0))
lm4
```

```
##
## Call:
## lm(formula = ratings ~ price + size + motion + style, data = conjointData.1,
##     subset = conjointData.1$gender == 1)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.8897 -2.5023  0.5787  2.7561  6.5706
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.2388     0.1761  41.099 < 2e-16 ***
## price         2.8032     0.1575  17.794 < 2e-16 ***
## size          1.4603     0.1575   9.269 < 2e-16 ***
## motion        0.5459     0.1575   3.465 0.000543 ***
## style         0.6447     0.1575   4.092 4.47e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.274 on 1723 degrees of freedom
## Multiple R-squared:  0.2002, Adjusted R-squared:  0.1984
## F-statistic: 107.8 on 4 and 1723 DF,  p-value: < 2.2e-16
```

```
lm5
```

```
##
## Call:
## lm(formula = ratings ~ price + size + motion + style, data = conjointData.1,
##     subset = conjointData.1$gender == 0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.4466 -1.9357 -0.5149  1.5534  8.5534
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.2695     0.1643  38.157 < 2e-16 ***
## price         3.4973     0.1470  23.797 < 2e-16 ***
## size          0.7468     0.1470   5.082 4.22e-07 ***
## motion       -0.1526     0.1470  -1.039  0.2992
## style        -0.4171     0.1470  -2.838  0.0046 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.819 on 1467 degrees of freedom
## Multiple R-squared:  0.2907, Adjusted R-squared:  0.2888
## F-statistic: 150.3 on 4 and 1467 DF,  p-value: < 2.2e-16
```

```
#by age
```

```
lm7 = summary(lm(ratings~ price + size + motion + style, data = conjointData.1, subset=conjointData.1$age == 1))
lm8 = summary(lm(ratings~ price + size + motion + style, data = conjointData.1, subset=conjointData.1$age == 2))
lm7
```

```
##
## Call:
## lm(formula = ratings ~ price + size + motion + style, data = conjointData.1,
##     subset = conjointData.1$age == 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.2380 -2.6277 -0.6089  3.0241  8.3100
```

```
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.62768    0.18970  34.938 <2e-16 ***
## price        3.28589    0.16967  19.366 <2e-16 ***
## size         1.54806    0.16967   9.124 <2e-16 ***
## motion      -0.06621    0.16967  -0.390  0.696
## style        0.12851    0.16967   0.757  0.449
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.41 on 1611 degrees of freedom
## Multiple R-squared:  0.2218, Adjusted R-squared:  0.2198
## F-statistic: 114.8 on 4 and 1611 DF,  p-value: < 2.2e-16
```

```
lm8
```

```
##
## Call:
## lm(formula = ratings ~ price + size + motion + style, data = conjointData.1,
##     subset = conjointData.1$age == 0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.3750 -2.1460 -0.1463  2.0385  7.1463
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.9615     0.1698  40.995 < 2e-16 ***
## price         2.9558     0.1519  19.461 < 2e-16 ***
## size          0.7077     0.1519   4.659 3.44e-06 ***
## motion        0.5213     0.1519   3.432 0.000615 ***
## style         0.1846     0.1519   1.215 0.224516
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.022 on 1579 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.2056
## F-statistic: 103.4 on 4 and 1579 DF,  p-value: < 2.2e-16
```

```
#interaction with gender and age
```

```
lm6 = summary(lm(ratings~ (price + size + motion + style)*gender*age, data = conjointData.1))
lm6
```

```
##
## Call:
## lm(formula = ratings ~ (price + size + motion + style) * gender *
##     age, data = conjointData.1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.9624 -2.1417 -0.1299  2.2088  9.3917
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.5569     0.2367  27.698 < 2e-16 ***
```

```
## price          3.1683      0.2117  14.963 < 2e-16 ***
## size           0.4399      0.2117   2.078 0.037824 *
## motion         0.3101      0.2117   1.465 0.143142
## style          -0.2155      0.2117  -1.018 0.308757
## gender          0.8522      0.3436   2.481 0.013170 *
## age            -0.6611      0.3590  -1.841 0.065668 .
## price:gender    -0.4475      0.3073  -1.456 0.145403
## size:gender      0.5641      0.3073   1.836 0.066506 .
## motion:gender    0.4448      0.3073   1.447 0.147887
## style:gender     0.8428      0.3073   2.742 0.006132 **
## price:age        0.7567      0.3211   2.357 0.018503 *
## size:age         0.7059      0.3211   2.198 0.027993 *
## motion:age       -1.0643      0.3211  -3.314 0.000929 ***
## style:age        -0.4636      0.3211  -1.444 0.148893
## gender:age       0.3595      0.4885   0.736 0.461848
## price:gender:age -0.6107      0.4370  -1.398 0.162343
## size:gender:age  0.1019      0.4370   0.233 0.815621
## motion:gender:age 0.6943      0.4370   1.589 0.112176
## style:gender:age 0.4945      0.4370   1.132 0.257816
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.054 on 3180 degrees of freedom
## Multiple R-squared:  0.2987, Adjusted R-squared:  0.2945
## F-statistic: 71.28 on 19 and 3180 DF, p-value: < 2.2e-16
```

#by gender and age

```
lm9 = summary(lm(ratings~ price + size + motion + style, data = conjointData.1, subset=conjointData.1$age == 1))
lm10 = summary(lm(ratings~ price + size + motion + style, data = conjointData.1, subset=conjointData.1$gender == 1))
lm11 = summary(lm(ratings~ price + size + motion + style, data = conjointData.1, subset=conjointData.1$age == 1 & conjointData.1$gender == 1))
lm12 = summary(lm(ratings~ price + size + motion + style, data = conjointData.1, subset=conjointData.1$age == 1 & conjointData.1$gender == 1))
lm9
```

```
##
## Call:
## lm(formula = ratings ~ price + size + motion + style, data = conjointData.1,
##     subset = conjointData.1$age == 1 & conjointData.1$gender ==
##         1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.9624 -2.7657  0.8707  2.8040  6.8494
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.1076     0.2457  28.926 < 2e-16 ***
## price         2.8668     0.2198  13.044 < 2e-16 ***
## size          1.8118     0.2198   8.244 5.37e-16 ***
## motion        0.3849     0.2198   1.751 0.08020 .
## style         0.6581     0.2198   2.995 0.00282 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.433 on 971 degrees of freedom
## Multiple R-squared:  0.2048, Adjusted R-squared:  0.2016
```

```
## F-statistic: 62.54 on 4 and 971 DF, p-value: < 2.2e-16
```

```
lm10
```

```
##
## Call:
## lm(formula = ratings ~ price + size + motion + style, data = conjointData.1,
##     subset = conjointData.1$age == 1 & conjointData.1$gender ==
##         0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5.2875 -1.6167 -0.8500  0.9333  9.3917
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.8958     0.2450  24.062 < 2e-16 ***
## price         3.9250     0.2192  17.910 < 2e-16 ***
## size          1.1458     0.2192   5.228 2.32e-07 ***
## motion        -0.7542     0.2192  -3.441 0.000617 ***
## style         -0.6792     0.2192  -3.099 0.002028 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.772 on 635 degrees of freedom
## Multiple R-squared:  0.3679, Adjusted R-squared:  0.3639
## F-statistic: 92.38 on 4 and 635 DF, p-value: < 2.2e-16
```

```
lm11
```

```
##
## Call:
## lm(formula = ratings ~ price + size + motion + style, data = conjointData.1,
##     subset = conjointData.1$age == 0 & conjointData.1$gender ==
##         0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.0913 -1.8790 -0.1651  1.5585  8.2188
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.5569     0.2154  30.436 <2e-16 ***
## price         3.1683     0.1927  16.442 <2e-16 ***
## size          0.4399     0.1927   2.283 0.0227 *
## motion        0.3101     0.1927   1.609 0.1079
## style        -0.2155     0.1927  -1.119 0.2636
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.779 on 827 degrees of freedom
## Multiple R-squared:  0.2525, Adjusted R-squared:  0.2489
## F-statistic: 69.85 on 4 and 827 DF, p-value: < 2.2e-16
```

```
lm12
```

```
##
```

```
## Call:
## lm(formula = ratings ~ price + size + motion + style, data = conjointData.1,
##     subset = conjointData.1$age == 0 & conjointData.1$gender ==
##         1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.7952 -2.1680  0.2088  2.3887  5.9597
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.4091     0.2486  29.805 < 2e-16 ***
## price         2.7207     0.2223  12.237 < 2e-16 ***
## size          1.0040     0.2223   4.515 7.34e-06 ***
## motion        0.7549     0.2223   3.395 0.000722 ***
## style         0.6272     0.2223   2.821 0.004916 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.049 on 747 degrees of freedom
## Multiple R-squared:  0.2024, Adjusted R-squared:  0.1982
## F-statistic:  47.4 on 4 and 747 DF,  p-value: < 2.2e-16
```

#by individual coefficient: Here we use the coefficient data set to calculate the mean of coefficient f

#individual by age

```
individual.age1 = coefficient[ (respondentData$age == 1),]
individual.age0 = coefficient[ (respondentData$age == 0),]
```

```
individual.agecompare = matrix(nrow = 5, ncol = 2)
for (x in 1:5) {
  individual.agecompare[x,1] = mean(individual.age0[,x])
  individual.agecompare[x,2] = mean(individual.age1[,x])
}
```

```
row.names(individual.agecompare) = c('intercept',names(conjointData.1)[4:7])
colnames(individual.agecompare) = c('age0', 'age1')
individual.agecompare
```

```
##           age0      age1
## intercept 6.9614899 6.62768152
## price     2.9558081 3.28589109
## size      0.7077020 1.54806106
## motion    0.5212542 -0.06621287
## style     0.1845539 0.12850660
```

#individual by gender

```
individual.gender1 = coefficient[ (respondentData$gender == 1),]
individual.gender0 = coefficient[ (respondentData$gender == 0),]
```

```
individual.gendercompare = matrix(nrow = 5, ncol = 2)
for (x in 1:5) {
  individual.gendercompare[x,1] = mean(individual.gender0[,x])
  individual.gendercompare[x,2] = mean(individual.gender1[,x])
}
```

```
}

row.names(individual.gendercompare) = c('intercept',names(conjointData.1)[4:7])
colnames(individual.gendercompare) = c('gender0', 'gender1')
individual.gendercompare
```

```
##           gender0  gender1
## intercept 6.2694746 7.2388117
## price     3.4972826 2.8032407
## size      0.7468297 1.4602623
## motion    -0.1526268 0.5459105
## style     -0.4171196 0.6446759
```

Here, to segment the respondentData(Age,Gender), regression analysis were used. To analysis whether there's different effect on attributes by gender and age, the interaction term were separately added into lm2 and lm3.

From lm2, if she is a girl, their parents will be less price sensitive. And girls prefer bigger sizes toy, Rocking, and Glamour style of toy horse. This is the same as post-hoc result as girls mostly stay in segment2 and 3 which are more prefer Rocking, and Glamour style and less price sensitive.

After knowing the difference between gender, we want to do the separate regression for different gender. As mention in previous, girls prefer more Rocking and Glamour toys than boys comparing to lm4 and lm5,

From lm3, in this regression, there is one weak marginal interaction effect (motion:age) and one one moderate interaction effect(size:age). Age not a important segmentation effect to consider compared to gender.

Even the interaction effect might be weak,we still separate the data by age. From lm7 and lm8, by separate by age, younger kids likes more Rocking toys, which is correspond to the preference on segment2(which also contain a lot young people). Meanwhile, even though the coefficient for size is both positive, the younger children would have less rate increase(coefficient for young children is close to 0) for the size become bigger. Thus, we could still find the similar effect in previous segment 2 tat the younger people would have more chance to prefer smaller size of toy hourse.

After segment the data by age and gender separately, we'll these two to separate the data together. In lm6, by analyzing the gender and age interaction effect, the result shows that older kids prefer less Rocking toys. To have deeper insight about the effect, we'll run the regression lm9-lm12 separately.

From lm9 and lm10, they shows older boys do not prefer Rocking and Glamour type toys.

From lm12, it shows younger girls like more Rocking and Glamour.

The result in this part also verify the post-hoc analysis as segment 1 do not prefer Rocking and Glamour and young children prefer those character.

Meanwhile, we try to obtain the mean of coefficient from Part(A) by the two factors to see the difference. While group by gender, which store in 'individual.gendercompare', the data also shows that girls like more Rocking and Glamour than boys. While group by age,the data also shows that older kids don't like Rocking

In conclusion, to find the ideal product line, we look at the coefficient in each of 4 situation. For older girls, since the coefficient on motion is not significant, the ideal type might be **p 16** or **p 12**. For younger girls, we suggest **p 16** as all the coefficient is positive. For older boys, we suggest **p 4** as it would have max rating under P4 criteria. For younger boys, we suggest **p 4** or **p12** the coefficient on motion and style is not significant for young boys, so we account the uncertainty in motion to make the decision. Comparing our recomendationton Part(B), the result is quit similar to post-hoc result for segment1 and 3.For segment 2 in previous cluster analysis, the ideal type is P14, but for young children, the ideal type is P16 mostly.This may result from the coefficient differences for young children and segment2 for priori and post-hoc segmentation in this case.

The coefficients in lm9-12 do not very significant mostly(especially for young children). Thus, we believed that the result might not correctly represent the real condition. Thus, we would mainly use post-hoc analysis

result for the following part.

Part D: Market Simulation

```
#Build up the dataframe to suggest the rating for each profile by each customer

market = cast(ID~profile,data= conjointData[,c("ID","profile","ratings")])

## Using ratings as value column. Use the value argument to cast to override this choice
# Calculate the profit margin for each profile as the input for profit calculation function

profit_table=data.table(profilesData)
profit_table$revenue = profit_table$priceLabel/1.25

profit_table[size == 0&motion==0,"cost"]=21
profit_table[size == 1&motion==0,"cost"]=29
profit_table[size == 0&motion==1,"cost"]=33
profit_table[size == 1&motion==1,"cost"]=41
profit_table$margin = profit_table$revenue-profit_table$cost
profit_table = profit_table[,c("profile","margin")]

#Please refer to the functions R script for the function we used below

market_simulation(c(5,7,13),market)

##      1  2  3  4          5  6          7  8  9 10 11 12          13 14 15 16
## 1 NA NA NA NA 0.2158333 NA 0.5683333 NA NA NA NA NA 0.2158333 NA NA NA

# Calculate the current market condition

market_simulation(1:16,market)

##      1    2 3      4 5    6 7      8 9 10 11          12 13      14      15    16
## 1 0 0.07 0 0.2625 0 0.12 0 0.0425 0 0 0 0.05583333 0 0.1066667 0.0025 0.34

# See which profile would perform better under all profiles are in the market
# P4,P6,P14,P16 seems to be better
#From the previous segmentation, we have P4,P14, and P16 from post-hoc segmentation
#From the prior segmentation, we have P2,P4,P12, and P16 as ideal product

# create the
scen=list()
scen[[1]] = c(5,13,7)
scen[[2]] = c(1:16)
scen[[3]] = c(5,13,6,7)
scen[[4]] = c(5,13,14,7)
scen[[5]] = c(5,6,14,7)
scen[[6]] = c(6,13,14,7)
scen[[7]] = c(5,6,7)
scen[[8]] = c(6,13,7)
scen[[9]] = c(14,13,7)
scen[[10]] = c(3,13,7)
```

```

scen[[11]] = c(4,5,7)
scen[[12]] = c(13,15,7)
scen[[13]] = c(2,4,5,7)
scen[[14]] = c(5,14,15,7)
scen[[15]] = c(13,14,15,7)
scen[[16]] = c(2, 5, 15, 7)
scen[[17]] = c(2, 13, 15, 7)
scen[[18]] = c(13, 15, 7)
scen[[19]] = c(5, 14, 15, 7)
scen[[20]] = c(5, 16, 7)
scen[[21]] = c(5, 13, 16, 7)
scen[[22]] = c(5, 16, 7)
scen[[23]] = c(13, 16, 7)
scen[[24]] = c(13, 14, 7)
scen[[25]] = c(5, 14, 7)
scen[[26]] = c(5, 13, 14, 7)
scen[[27]] = c(5,12,7)
scen[[28]] = c(13,12,7)
scen[[29]] = c(4,5,7)
scen[[30]] = c(12,13,16,7)
scen[[31]] = c(5,12,16,7)
scen[[32]] = c(4,5,12,7)
scen[[33]] = c(5,12,16,7)

```

```

scen_without =list()
scen_without[[1]] = c(6,14,7)
scen_without[[2]] = c(6,7)
scen_without[[3]] = c(14,7)
scen_without[[4]] = c(6,14,7)
scen_without[[5]] = c(16,7)
scen_without[[6]] = c(16,5,7)
scen_without[[7]] = c(16,6,7)
scen_without[[8]] = c(3,14,7)
scen_without[[9]] = c(3,6,14,7)
scen_without[[10]] = c(3,14,7)
scen_without[[11]] = c(2,4,7)
scen_without[[12]] = c(4,15,7)
scen_without[[13]] = c(2,4,15,7)
scen_without[[14]] = c(4,14,15,7)
scen_without[[15]] = c(4,14,7)
scen_without[[16]] = c(14,15,7)
scen_without[[17]] = c(2,4,15,7)
scen_without[[18]] = c(2,15,7)
scen_without[[19]] = c(2, 4, 15, 7)
scen_without[[20]] = c(2, 14, 15, 7)
scen_without[[21]] = c(4, 14, 15, 7)
scen_without[[22]] = c(14, 15, 16, 7)

```

```

scen_without[[23]] = c(16, 7)
scen_without[[24]] = c(4, 16, 7)
scen_without[[25]] = c(9, 16, 7)
scen_without[[26]] = c(14, 16, 7)
scen_without[[27]] = c(4,12, 16, 7)
scen_without[[28]] = c(12, 16, 7)
scen_without[[29]] = c(4, 12, 7)
scen_without[[30]] = c(16, 12, 7)

```

Test the scenario which contain original porfile

```
with_origin = comparison(scen)
```

```
result = profit(with_origin)
```

```
result = data.table(result)
```

```
result[company_profit > 190000,]
```

```

##      company_profit comp_profit
## 1:      211552.5      -9824.48
## 2:      212105.8     -10297.76
## 3:      216759.5     -6511.52
## 4:      219135.9     -9114.56
## 5:      190534.3     -8286.32

```

```
result[,which(company_profit > 190000),]
```

```
## [1] 16 17 27 28 32
```

#There are 5 scenarios which have profit larger than 190,000. Most of them contain 2,12,15 profile.

```
without_origin = comparison(scen_without)
```

```
result_without = profit(without_origin)
```

```
result_without = data.table(result_without)
```

```
result_without[company_profit > 190000,]
```

```

##      company_profit comp_profit
## 1:      197251.5      11236.48
## 2:      191868.2     -18580.16
## 3:      193891.1       7450.24
## 4:      198465.7     -11244.32
## 5:      225083.0     -5328.32
## 6:      193911.9     -14557.28
## 7:      191868.2     -18580.16
## 8:      190744.1       4610.56

```

```
result_without[,which(company_profit > 190000)]
```

```
## [1] 3 5 12 16 18 20 23 29
```

There are 8 scenarios will have profit more than 190,000. They mostly contain 4,14,15. The 16 would p

```
i=1
```

```
j=1
```

```

table=matrix(nrow=4,ncol=3)
for(j in 1:3){
for(i in 1:4){
  t = c(unlist((cbind(1,profilesData[c(2,12,13,15),c(2,3,4,5)])[i,]))) * c(unlist((clusters$kms[[2]][[2]]))
  table[i,j]=sum(t)
}
}
table

##           [,1]      [,2]      [,3]
## [1,] 9.892969 9.593160 10.19714
## [2,] 9.651302 8.310142 15.34639
## [3,] 1.802344 9.305425 11.90112
## [4,] 2.921354 7.956761 15.01119

# calculate the mean rating in each segment to find out the taeget customerscharacters

```

To simulate the market, functions were written to avoid repeat works on calculation. The input would be the rating dataset and the list of scenario to evaluate. The outcome would be the market share distribution of each scenario. After that, the profit function would be used to calculate the profit for competitors and our company. The input would be market share data from previous output.

For market simulation, we generate our candidate profile from previous segmentation analysis and also the simulation of the condition that all profiles are inside the market. From the cluster analysis, we identify **P4 P14 P16** as ideal type for each segment. In the priori segmentation, **P4 P8 P12 P16** would be ideal for the segments. Meanwhile, from the all profile existed condition, **P4 P14 P16** seems to have more market share than others, which may suggest that they have more absolutely high value among 200 respondents. Thus, we will mainly simulate the market with **P2 P4 P12 P14 P16** as they are mention in previous analysis and try to maximize the profit for our company. Other profile including **P3 P9 P15** might also be include randomly to see other potential combination to further increase the profit

For the competitor, we assume that the competitor would offer 26" Racing Rocking Horse, which is P7, all over the period and won't change the type in the analysis period.

Here we separate our analysis into two category: with and without original product line, which is **P5 P13** to form the market scenario. After the calculation for the scenario, we found that for with original group, there are 5 scenarios which have profit larger than 190,000. Most of them contain **P2 P12 P15**. On the other hand, there are 8 scenarios will have profit more than 190,000 in without original profile group. They mostly contain **P4 P14 P15**. Also, we found that even though P16 would have best porpotion in all profile condition, it will perform best while this profile appear alone, which may because the cannibalization effect on P16 and other profiles.

For the two category, we can identify the senario that provide highest profit to the company. For with original product, Earlyriders might want to keep offering P13 and change offering P5 to offering P12. his shift the profit from 96,3930 to 219,136 dollars. For without original product view, the company should offer two new products, one is 18 inches Bouncing Racing Horse with the price of 119.99 dollars and the other one is 26 inches Rocking Glamour with the price of 139.99 dollars. The profit would further shift to 225,083 dollar.

To sum up, the maximize the profit, the company should offer the product correspond to **P2 P15**. Meanwhile, if the company consider the risk to totally change the product line is too high, the company can also consider an alternative plan to keep offering 18" Glamorous Rocking Horse and change offering 18" Racing Rocking Horse to offering 26" Glamorous Bouncing Horse(**P12 P13**). This will only change 1 model, receive much higher profit than current market, and also prevent unexpected condition for changing too much on the product line.

To find potential customers character from the proposed scenario, we look back to the coefficient mean in each segment. Here, we have P2,P12,P15,P13 in our candidate pool. For without origin group, P2 and P12 would have highest rate on **segment3**. For with original product group, both P13 and P15 would perform

best rating on segment 3. Thus, we could conclude that no matter which policy the company choose, the **older girls**.would be the priority targets.