1^{st} Assignment

We'll look at Conjoint Analysis in this study and see how it's used in two examples. The first example is about attributes that can be utilized to buy a car, while the second is about attribution analysis and conclusions about tea. But what exactly is Conjoint analysis and how does it work?

Conjoint analysis is a form of statistical analysis that firms use in market research to understand how customers value different components or features of their products or services. It’s based on the principle that any product can be broken down into a set of attributes that ultimately impact users’ perceived value of an item or service. Conjoint analysis is typically conducted via a specialized survey that asks consumers to rank the importance of the specific features in question. Analyzing the results allows the firm to then assign a value to each one. The insights a company gleans from conjoint analysis of its product features can be leveraged in several ways. Most often, conjoint analysis impacts pricing strategy, sales and marketing efforts, and research and development plans. The following are some examples of possible approaches: Conjoint Analysis in **Pricing**, **Sales & Marketing**, **Research & Development.**

. What will be examined in this case are what criteria one would use to purchase a car and how crucial each feature would be in making this decision. The 5 attributes used are:

1. Price (1.000-4.000€,5.000-10.000€,11.000-20.000€,20.000€+)
2. Cubic Capacity (800-1.200cc,1.250-1.600cc,1.650-2.000cc,2.000+cc)
3. Kilometers that have been done(0-50.000,50.000-100.000,100.000-200.000,200.000+)
4. Used=(yes, no)
5. Crashed=(yes, no)

Because too many attributes would place responders under cognitive strain, only five were used.

Price	*Cubic Capacity*	*Kilometers*	*Used*	*Crashed*
1.000-4.000	800-1.200	0-50.000	Yes	Yes
5.000-10.000	800-1.200	0-50.000	Yes	Yes
11.000-20.000	800-1.200	0-50.000	Yes	Yes
20.000+	800-1.200	0-50.000	Yes	Yes
1.000-4.000	1.250-1.600	0-50.000	Yes	Yes
5.000-10.000	1.250-1.600	0-50.000	Yes	Yes

Table 1.Full Factorial

The results of the full factorial design are shown in table 1. Full factorial design creates experimental points using all the possible combinations of the levels of the factors in each complete trial or replication of the experiments. Using this design, all the possible combinations of factor levels can be investigated in each replication.

Price	***Cubic Capacity***	***Kilometers***	***Used***	***Crashed***
11.000-20.000	2.000+	0-50.000	Yes	Yes
20.000+	1.650-2.000	50.000-100.000	Yes	Yes
5.000-10.000	1.250-1.600	100.000-200.000	Yes	Yes
5.000-10.000	800-1.200	200.000+	Yes	Yes
20.000+	800-1.200	0-50.000	No	Yes
11.000-20.000	1.250-1.600	50.000-100.000	No	Yes
1.000-4.000	1.650-2.000	100.000-200.000	No	Yes
1.000-4.000	2.000+	200.000+	No	Yes
1.000-4.000	1.250-1.600	0-50.000	Yes	No
1.000-4.000	800-1.200	50.000-100,000	Yes	No
20.000+	2.000+	100.000-200.000	Yes	No
11.000-20.000	1.650-2.000	200.000+	Yes	No
5.000-10.000	1.650-2.000	0-50.000	No	No
5.000-10.000	2.000+	50.000-100.000	No	No
11.000-20.000	800-1.200	100.000-200.000	No	No
20.000+	1.250-1.600	200.000+	No	No

Table 2.Near Orthogonal Design

The results of the Fractional Factorial Near-Orthogonal Design are shown in Table 2. In Near-Orthogonal design the attributes are uncorrelated and attribute levels are unbalanced, which means that each level does not appear an equal number of times for each attribute. Generally, a fractional factorial design looks like a full factorial design for fewer factors, with extra factor columns added (but no extra rows). Using fractional factorial design makes experiments cheaper and faster to run, but can also obfuscate interactions between factors.

Price	***Cubic Capacity***	***Kilometers***	***Used***	***Crashed***
11.000-20.000	2.000+	0-50.000	yes	yes
20.000+	1.650-2.000	50.000-100.0000	yes	yes
5.000-10.000	1.250-1.600	100.000-200.000	yes	yes
5.000-10.000	800-1.200	200.000+	yes	yes
20.000+	800-1.200	0-50.000	no	yes
11.000-20.000	1.250-1.600	50.000-100.0000	no	yes
1.000-4.000	1.650-2.000	100.000-200.000	no	yes
1.000-4.000	2.000+	200.000+	no	yes
1.000-4.000	1.250-1.600	0-50.000	yes	no
1.000-4.000	800-1.200	50.000-100.0000	yes	no
20.000+	2.000+	100.000-200.000	yes	no
11.000-20.000	1.650-2.000	200.000+	yes	no
5.000-10.000	1.650-2.000	0-50.000	no	no
5.000-10.000	2.000+	50.000-10.0000	no	no
11.000-20.000	800-1.200	100.000-200.000	no	no
20.000+	1.250-1.600	200.000+	no	no

Table 3.Efficient Design

The results of the Efficient Design are shown in table 3. Sometimes near orthogonal and even blocking will not be enough (too many profiles) .As a result, in this case, the approach you prefer is the Efficient Design. This method selects a subset of profiles from the Full fractional design and computes efficiency measures. There were 12 profiles used in this case.

After that, we'll look at the next example. This example, which belongs in the Conjoint library, is about a package of "tea". This package concerns sample data on a rating scale collected in 2007 about preferences of tea consumers. The product described by 4 attributes (with following attributes' levels): price (low, medium, high), variety (black, green, red), kind (bags, granulated, leafy), aroma (yes, no).The content of the package is presented below:

1. tprefm - matrix of preferences (100 respondents and 13 profiles)
2. tpref - vector of preferences (length 1300)
3. tprof - matrix of profiles (4 attributes and 13 profiles)
4. tlevn - vector of names for the attributes' levels (11 levels)
5. tsimp - matrix of simulation profiles (4 attributes and 4 profiles)

The Full Factorial design contains 100 observations of 13 variables and the size of it is 13*100. The Final Design, which comprises of 13 profiles and their 4 selections ,the size of it is 13*4,, is shown in table 4.

Price	*Variety*	*Kind*	*Aroma*
High	Bags	Black	Yes
Low	Granulated	Black	Yes
Medium	Granulated	Green	Yes
Medium	Bags	Red	Yes
High	Leafy	Red	Yes
Medium	Bags	Black	No
High	Granulated	Black	No
Medium	Leafy	Black	No
High	Bags	Green	No
Low	Leafy	Green	No
Low	Bags	Red	No
Medium	Granulated	Red	No
High	Granulated	Red	No

Table 4.Final Design

Levels	*Utilities*
Intercept	3,5534
Low	0,2402
Medium	-0,1431
High	-0,0971
Black	0,6149
Green	0,0349
Red	-0,6498
Bags	0,1369
Granulated	-0,8898
Leafy	0,7529
Yes	0,4108
No	-0,4108

Table 5.Levels Utilities

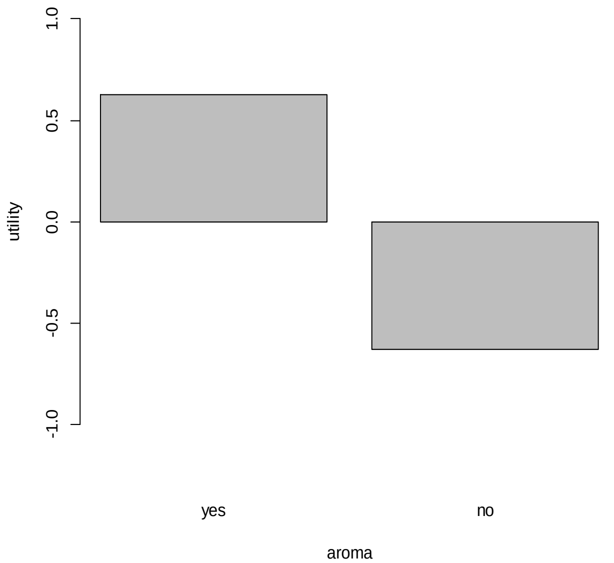

Figure 1.Aroma Utility

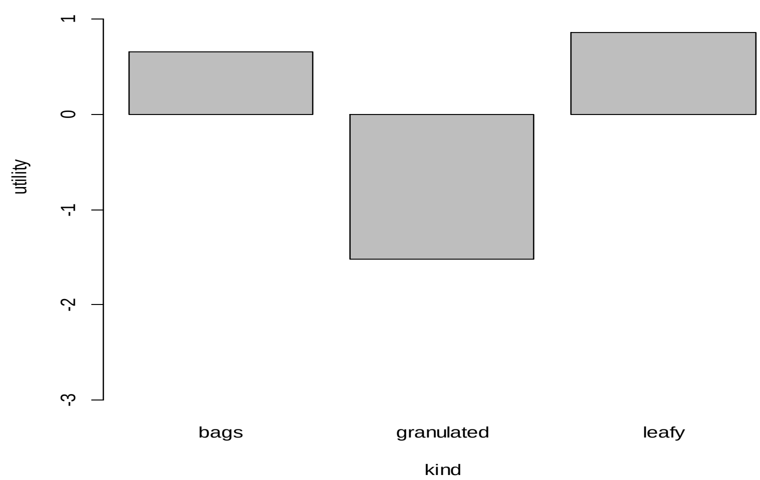

Figure 2. Kind Utility

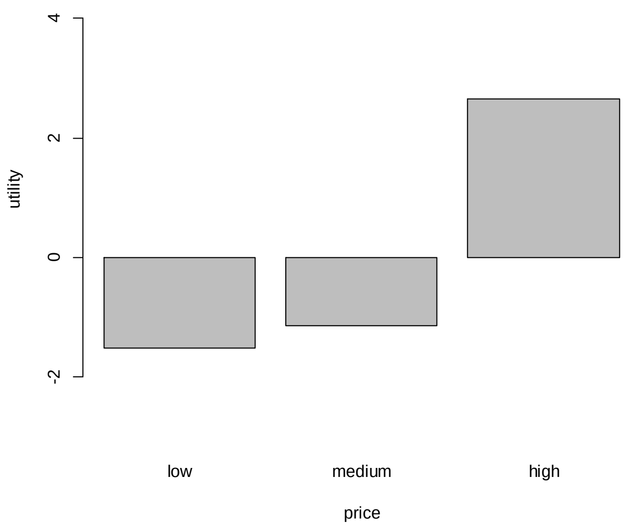

Figure 3.Price Utility

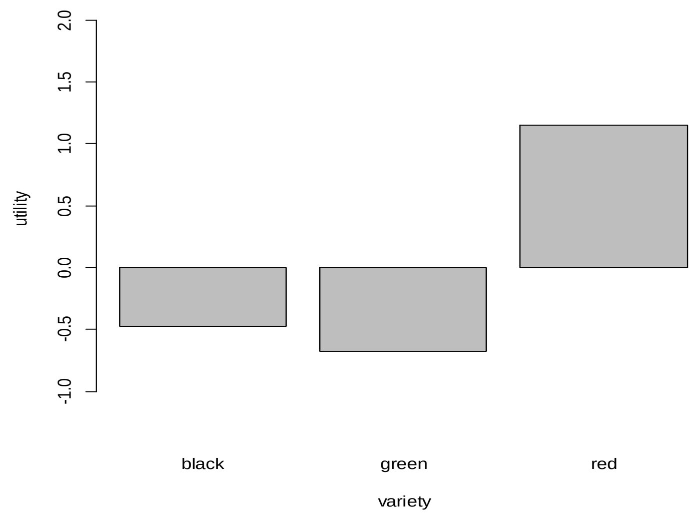

Figure 4.Variety Utility

Price	*Variety*	*Kind*	*Aroma*
24,76	32,22	27.15	15,88

Table 6.Average Importance

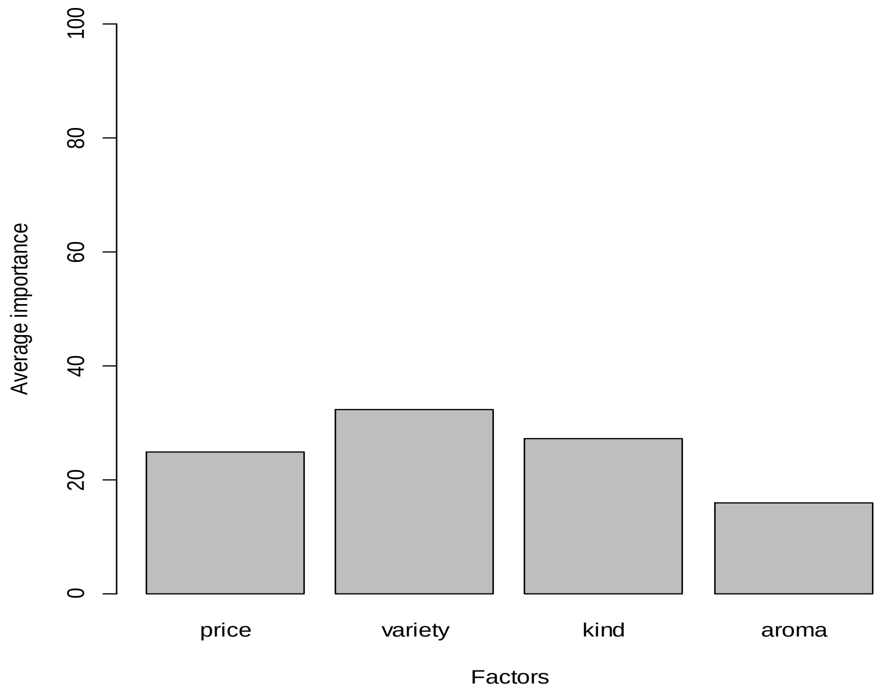

Figure 5. Average Importance

The utility function for each level is shown in table 5. The levels Black, Leafy, and Yes have the biggest positive utility function. The levels of Red, Granulated, and No, on the other hand, have the greatest negative utility function. At a better understanding, see figures 1-4, where we plot the utility function for each level for each attribute. Finally, the average importance for each Attribute is shown in Table 6 and Figure 5. We can observe that the Variety is the most important Attribute, while the Aroma is the least important.

Total Utility	*Max Utility*	*BTL Model*	*Logit Model*
2,19	11	18,85	13,04
3,69	28	28,96	34,84
3,10	26	30,14	35,79
3,59	35	22,06	16,33

Table 7.Market Simulation

The Total Utility, Max Utility, BTL Model, and Logit Model are all shown in Table 7. Market simulation is based on these results. Market simulations provide information on the relative share of respondents who prefer predefined products in a certain competitive environment. They enable managers to test alternative market scenarios.

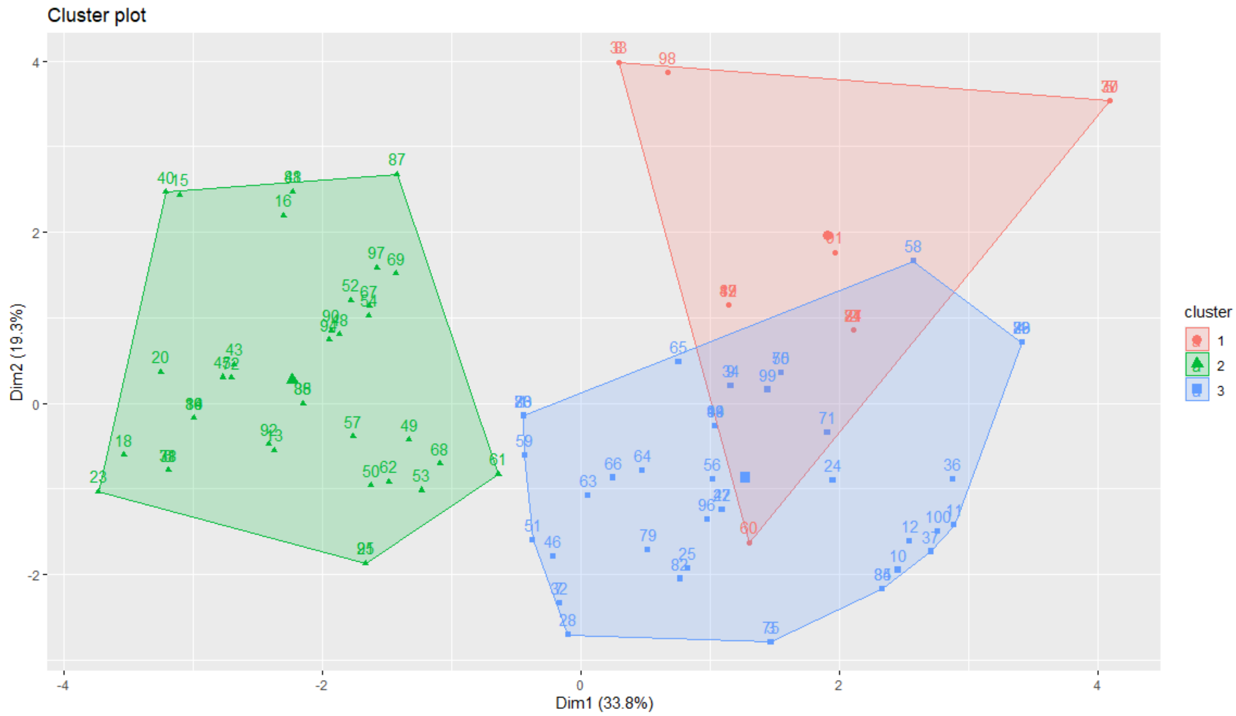

Figure 6.Clustering

Figure 6 shows the Utility function-based grouping of consumers using three different centers.

Conclusions

According to the findings in this study, a Product Manager should begin by focusing on product Variety and Kind, rather than Price and Aroma. More particular, we discover that customers like leafy black tea and that it should be heavily marketed, whereas red granulated tea appears to have a very poor utility function and should even be abolished. Green tea and bags are more neutral, thus with more promotion, the product might get a better response.

CODE

First example

```
library(conjoint)

library(factoextra)

setwd("C:/Users/Bru/Desktop/metaptuxiako/2o/vasilopoulos")

car<- expand.grid(price=c('1,000-4,000','5,000-10,000','11,000-20,000','20,000+'),

                cubic_capacity=c('800-1,200','1,250-1,600','1,650-2,000','2,000'),

                mileage=c('0-50,000','50,000-10,0000','100,000-200,000','200,000'),

                used=c('yes','no'),

                crashed=c('yes','no'))

head(car)

North<-caFactorialDesign(data=car, type="orthogonal",

                     seed=200)

North

nrow(North)

Efficient<-caFactorialDesign(data=car, type="fractional",

                        seed=200,cards=12)

Efficient
```

Second example

```
data(tea)

str(tprof)

str(tlevn)

str(tprefm)

dim(tprefm)

head(tprefm)
```

```
tprof

dim(tprof)

tprefm[1,]

caModel(y=tprefm[1,], x=tprof)

caUtilities(y=tprefm[1,], x=tprof, z=tlevn)

caImportance(y=tpref, x=tprof)

caPartUtilities(y=tprefm[1:10,], x=tprof, z=tlevn)

caPartUtilities

Conjoint(y=tpref, x=tprof, z=tlevn)

ShowAllSimulations(tsimp,tpref,tprof)

totalU <- caTotalUtilities(tpref,tprof)

totalU

ShowAllSimulations(tsimp,tpref,tprof)

fviz_cluster(segments, data = totalU)

segments <- kmeans(totalU, centers = 3,  nstart=25)
```