

SYMSON

# Adaptive Conjoint Analysis

*Internship*

Amadeo Villar Guardia

December 15, 2021



**Submission date:** December 16, 2021  
**Course:** WI5118 Internship 18ECTS  
**Project:** Developing a prototype for Conjoint analysis experiments  
in an e-commerce environment.  
**Supervisors:** Felix Jansen - Raluca Lichiardopol

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Choice Based Conjoint</b>	<b>5</b>
<b>3</b>	<b>Questionnaire Building</b>	<b>7</b>
3.1	Identification of Attributes and Attribute Levels . . . . .	7
3.2	Behavioural Questions . . . . .	8
3.3	Creating the Experimental Design . . . . .	9
3.4	Implementation . . . . .	11
<b>4</b>	<b>Models</b>	<b>14</b>
4.1	Decision Model . . . . .	14
4.1.1	Utility Model: . . . . .	14
4.1.2	Choice Model: . . . . .	17
4.2	Estimation techniques . . . . .	18
4.2.1	Multinomial Logit Model . . . . .	18
4.2.2	Latent Class Model . . . . .	19
4.2.3	Hierarchical Bayes Mixed Multinomial Model . . . . .	19
<b>5</b>	<b>Data Analysis</b>	<b>21</b>
5.1	Multinomial Logit Model . . . . .	21
5.2	Latent Class Model . . . . .	24
5.3	Hierarchical Bayes Mixed Multinomial Logit Model . . . . .	26
5.4	Model Validation . . . . .	31
<b>6</b>	<b>Applications</b>	<b>33</b>
6.1	Customer Clustering . . . . .	33
6.2	Market Simulator . . . . .	36
6.2.1	Price Elasticities of New Products . . . . .	36
6.2.2	Willingness To Pay per Feature . . . . .	39
6.2.3	Designing Products for Market Segments . . . . .	41
6.3	Remarks . . . . .	45
<b>7</b>	<b>Future Research</b>	<b>46</b>

# 1 Introduction

Conjoint Analysis is a survey-based statistical technique used in market research that helps determine how people value different attributes that make up an individual product or service [1]. Its main objective is then to find what characteristics of the product lead consumers to make purchasing decisions.

Conjoint Analysis, as one of the most popular methods within preference measurement [2], [3], assumes that products are attribute bundles. These attributes can be categorical, having different levels, or they can be continuous. The method then tries to estimate the utilities that each attribute level adds to the overall utility of the product. The idea is the following:

- The utility of a product is the sum of the utilities of its attributes levels. This is known as the part-worth utility model.
- The higher the utility of a product the more desirable is for a consumer.

This can be better illustrated showing an example.

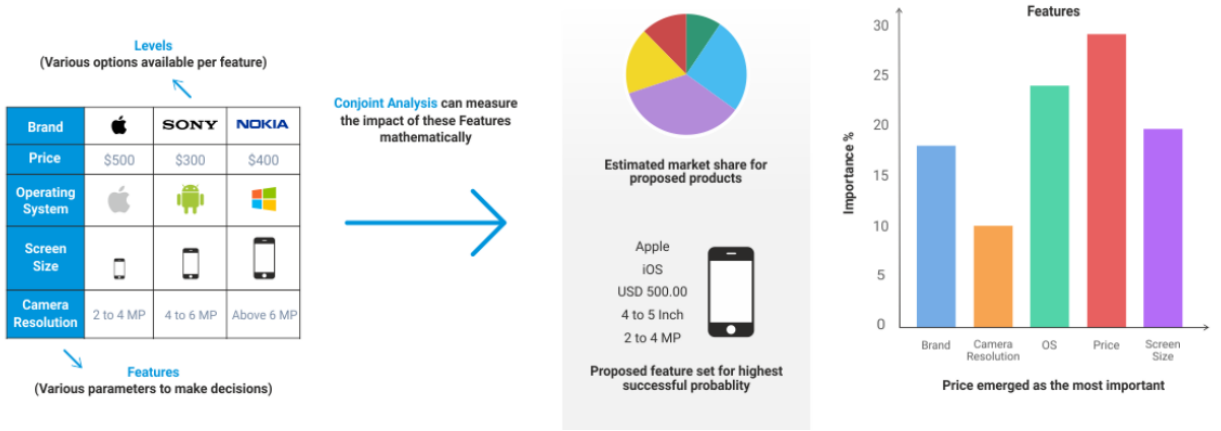


Figure 1: Example of a mobile phone Conjoint Analysis.

In Figure 1, we consider a mobile phone as the sum of its different attributes:

Mobile Phone: Brand + Price + Operating System + Screen Size + Camera Resolution.

These attributes can have different levels, i.e, brand could be either iPhone, Huawei or Samsung, or being continuous, such as the Price, that could range from 100\$ up to 1000\$. The result of performing a Conjoint Analysis would be a series of utilities for the different attribute levels, that would reflect the preference measurement of a consumer. For example, if we obtain:

$$u_{\text{iPhone}} = 0.55 > 0.40 = u_{\text{Samsung}},$$

means that for the consumer is more valuable an I-phone than a Samsung. Moreover if we obtain:

$$u_{\text{Storage:64GB}} = 0.70 > 0.20 = u_{\text{Storage:4GB}},$$

means not only that the consumer prefers an Storage of 64GB than 4GB, but also that the phone Storage is more critical than the Brand when making purchasing desicions. This last statement follows from:

- Mobile 1: Storage 64GB, Brand Samsung.
- Mobile 2: Storage 4GB, Brand iPhone.

$$u_{\text{Mobile1}} = \underbrace{0.70}_{u_{\text{Storage:64GB}}} + \underbrace{0.40}_{u_{\text{Samsung}}} = 1.1 > 0.75 = \underbrace{0.20}_{u_{\text{Storage:4GB}}} + \underbrace{0.55}_{u_{\text{IPhone}}} = u_{\text{Mobile2}}$$

Conjoint Analysis is a decompositional method in the sense that it decomposes the overall product utility into the attributes' utilities via statistical procedures. Once these utilities have been computed, we can explain purchase decisions and predict future consumer choices [4]. In this regard, it is the basis for a multitude of relevant marketing applications:

1. New product development and innovation → Which product attribute will be preferred by consumers? For example, if the Camera Resolution happen to be the most important attribute, then it might be a good idea to release a new version of the product with higher Camera Resolution than the previous versions.
2. Branding → How much value can be attributed to the brand of a product?
3. Market segmentation → Are there different market segments that differ in terms of certain preferred product attributes? and in that case, Do the segment members have common features? For example, it could happen that for students price would be the most important feature when purchasing, whereas for business people Storage could be more relevant. Then we could make tailored offers depending on the cluster where the customer belongs.
4. Pricing → How much are consumers willing to pay and how much are improvements in products attributes allowed to cost? For example, computing the optimal price of a phone if we increase the screen by 1inch, but the rest of the features remain unchanged.
5. Market scenarios → What are the chances that a consumer buys a product among a list of items? Imagine the following scenario.
  - My company: Mobile1, chances of buying 40%.
  - Competitor: Mobile2, chances of buying 60%.

With the market simulators, we could compute how much discount should we offer to the customer in order to reverse the scales.

## 2 Choice Based Conjoint

The next question we ask ourselves is: How can we perform a Conjoint Analysis? There are several methods that differ in terms of how the overall utilities are elicited [1]. All of them show products formed by different attribute levels and they ask the respondents about their preferences. The main differences are mainly in the way of asking. Traditional approaches:

- Rating-based conjoint: use ratings of single product concepts
- Ranking-based conjoint: rankings of a selection of products




Currently the most popular conjoint approach and the one that we are going to put our efforts on is the so-called:

- Choice Based Conjoint.

It is based on showing the respondents a selection of product alternatives in a choice set and ask for the most preferred option [1], [5]. This procedure is repeated across multiple sequential choice sets, each presenting alternatives that are systematically varied by an experimental design. The decision within a choice set often require a trade-off between attributes. This can be illustrated with a camera example in Figure 2:

If these were the options available to you when buying a new compact photo camera, which would you choose?

Task (6 of 12)

Brand			
	Shoot	Kodak	Olympus
Resolution	18 MP	8 MP	5 MP
Optical zoom	10x	4x	8x
Battery life	<200 photos	300-400 photos	>400 photos
Image stabilizer	Yes	Yes	No
Price	€ 299	€ 149	€ 249

Given what you know about the market, would you really buy the compact camera you chose above?

☐ Yes

☒ No

Figure 2: Example of a Choice-Based Conjoint Analysis.

In this case comparing the Shoot and Kodak cameras we have to decide if it is worth spending more money for the Shoot in exchange for better resolution or higher optical zoom. On the other hand, one might be interested in having a great battery life and would not doubt about buying the Kodak camera. Additionally, one might not be interested in changing their current camera by any of the 3 options, and would not buy any of the above items.

Among the features of the Choice-Based Conjoint [1], we highlight:

- It is the most used since it mimics consumer behaviour when purchasing.
- It is more realistic than the ranking/rating aforementioned models.
- It is able to incorporate the NO-choice option. That is a option: I would not buy any of the previous items. This feature it is extremely important to find threshold purchasing prices, and it increases the realism of CBC.
- The only disadvantage is that less information is obtained per question with respect to the previous methods. Therefore CBC requires the collection of multiple choice set, that can lead to respondent fatigue.

Furthermore, for this project we have decided to make an Adaptive Conjoint Analysis [6]. The term adaptive means that the choice situations of the questionnaire are generated based on the previous answers. This fact lead us to more efficient data collection and dynamical procedures to adjust the preferences of the respondents.

Once we have introduced the Conjoint Analysis statistical technique in Section 1 and the Choice-Based type in Section 2, we proceed with the important parts of the project. In Section 3 we propose a method for building an adaptive questionnaire that let us obtain efficient data. In Section 4 we study the mathematical models behind Conjoint Analysis and we present some tools for the estimation procedure. In Section 5 we applied the previous methods to a real ebook data set<sup>1</sup> and we show the obtained results. In Section 6 we show the possible marketing applications that the previous data analysis has. Lastly, in Section 7 we comment on possible further improvements and recommendations. The main objective of this report is to guide the reader through the different stages of the Conjoint Analysis process, in order to show all the concepts and the reasoning behind them. The final purpose is to implement a Conjoint Analysis tool as an additional feature of the Symson Software.

---

<sup>1</sup><http://www.preferencelab.com/data/CBC.R>.

## 3 Questionnaire Building

This section will be mainly dedicated to build the questionnaire that will be shown to the respondents. We can split this section into 4 parts. The first part will deal with the identification of Attributes and Attribute Levels. The second one will be devoted to create a behavioural framework prior to the questionnaire. The third part will deal with creating the experimental design. Finally, in the forth part we will show an example of the implementation carried out. The idea of this section is to provide the client for whom we are developing the Conjoint Analysis, the implementation of the questionnaire. So that they would be the ones to collect data from their surveyed customers on their own website. Sometimes it is advisable to offer some kind of incentive for these respondents, either a discount or a shorter delivery time, to try to obtain better quality data. We wil comment further in [Section 7](#) possible improvements for this section.

### 3.1 Identification of Attributes and Attribute Levels

The prerequisite and most important step in conducting conjoint analysis is to identify the relevant determinants of consumer choices, i.e. product attributes and their levels [1] The selection of attributes and levels should reflect the products in the market and affect consumer preferences. Otherwise, the validity of the model may be questioned. In general, the selection of attributes has to meet the following requirements:

- The attributes must be relevant, i.e., they must influence consumers' utility. At Qualitative studies, such as focus groups or in-depth interviews, can be used to identify relevant attributes.
- Attributes must be discriminatory, i.e., they must be able to differentiate between competitive offerings in the market.
- The number of attributes should be manageable. CBC experiments typically use fewer than seven attributes. The use of more attributes greatly increases the complexity of the experimental design and requires high cognitive abilities from respondents.
- Attributes should not be interrelated, i.e., they should measure product-independent aspects. If attributes are interrelated, some combinations could be very unrealistic and confuse respondents.

Additionally, the selection of attribute levels has to meet the following requirements:

- The levels should span a range that is larger than in reality, but not substantially, in order to be able to cover potential future scenarios.
- Levels that have an ambiguous meaning should be avoided.
- The number of levels should be kept low because the complexity of the experimental design will increase exponentially with more levels.

- The number of levels should be balanced across attributes. Otherwise, the number-of-levels effect can occur, which leads to an artificially higher relevance of attributes that have more levels.
- Attribute levels are assumed to be mutually exclusive.

The idea would be for the customer to decide which attributes and levels to use.

## 3.2 Behavioural Questions

As shoppers process information and act on it, they are not simple stimulus-response robots. Creating a behavioral framework prior to answering choice tasks therefore helps respondents select from choice tasks as if they were in a real purchase situation [7]. If price and assortment changes are the focus of the research, it is particularly important to understand shopper perceptions of prices and values. Again, a behavioral framework is useful for interpreting consumer decisions, as simulated by the results of the choice model, in the appropriate context.

To create such a behavioral framework, prior to each conjoint exercise, we apply nine standardized, binary “Behavioral Calibration Questions” regarding each respondent’s individual shopping behavior for the focal category. Based on principles from behavioral economics, these questions help consumers recall their usual buying habits. “Behavioral Calibration Questions” are also used to describe the context of consumer choices, including how purchase decisions are made within a specific category, as they reveal typical patterns of buying habits, purchase repertoires, and brand value perceptions, as well as price knowledge. These questions that we are using are:

We would like to learn a few things about you and your general thoughts, feelings, and opinions when it comes to home upkeep, construction adhesives.  
Please read each pair of statements. For each pair, please indicate whether you agree with the statement on the left or the statement on the right more, and how much more.  
If both statements describe your opinion well, choose the one that best describes you. If neither seems to describe you well, choose the one that comes the closest.  
Select one response for each.

	Agree Left	Agree Right	
I think that brands differ a lot	<input type="radio"/>	<input type="radio"/>	I think that all brands are more or less the same
I always know exactly what brand I'm going to buy before I enter the shop	<input type="radio"/>	<input type="radio"/>	I decide what brand I'm going to buy when I'm standing in front of the shelf
I always buy the brand I bought last time	<input type="radio"/>	<input type="radio"/>	I switch between different brands
I compare prices very carefully before I make a choice	<input type="radio"/>	<input type="radio"/>	To be honest, I compare prices only superficially
I always search for special offers first	<input type="radio"/>	<input type="radio"/>	Special offers are not the first thing I look out for
I always know the price of the products I buy	<input type="radio"/>	<input type="radio"/>	I never really know what products cost
I'm always interested in new products	<input type="radio"/>	<input type="radio"/>	I prefer to stick to what I know
I think that products in this category need to be improved	<input type="radio"/>	<input type="radio"/>	I'm completely satisfied with the products as they are
I find it easy to make the right choice for me	<input type="radio"/>	<input type="radio"/>	I find it very difficult to make the right choice for me

Example from R&D study in US (2020, context: construction adhesives)<sup>1</sup>

Figure 3: Behavioural Questions.



Asking the nine “Behavioral Calibration Questions” before our choice exercise helps respondents to recall their behavior during their last shopping trip in a specific category. Therefore, we assume that the nine questions improve their decision-making process in the subsequent choice exercise, supporting a realistic answering behavior comparable to real shopping situations. Therefore, this approach helps generate more realistic data. Using the derived shopper classifications as segmentation variables in the choice simulator provides deeper insights into respondents’ preference structure. Based on our findings from numerous conjoint exercises, we learned that answering the nine questions results in better “Share of Choice” estimates as compared with conjoint exercises performed without the calibration questions. Furthermore, part-worth estimates, which include the “Behavioral Calibration Questions” as covariates, further improve share predictions against holdout samples (ensembles with the questions and other covariates offer marginal improvement in results).

### 3.3 Creating the Experimental Design

The experimental design determines which combinations of attribute levels are presented to the respondent as stimuli (factorial design) and how these stimuli are allocated to choice sets (choice design). The main objective is to build an Adaptive Questionnaire and to store respondent’s answers to analyse them later using the methods in [Subsection 4.2](#). Among the basic features of this questionnaire building that have been successfully achieved in our code, we have:

- Balance: each attribute level is showed equal number of times.
- Orthogonality: attribute levels are uncorrelated
- Minimal overlap: alternative within a choice are maximally different.

On the other hand, we show more complex features that we have implemented, but that represent a current challenge for improvement.

- Price: we will treat price as a discrete variable, and we will use the piece-wise model for estimating utilities of the different levels. This way of treating the price will allow more flexibility than the linear model that would be obtained by treating the price continuously. Moreover, we use the so-called conditional summed price strategy to determine the price of each item in the questions [\[8\]](#). The price of an item will be given by the following formula:

$$\text{Price} = \text{round} \left[ \left( \sum_{i=1}^n \text{Price}_{\text{Attribute-level } i} + \text{Price Basis} \right) \times u \times v \right]$$

where:

1. We have to assign some price priors to each of the attribute levels based on prior information available. As questions are answered, these priors will become less important. This resembles what happens when using the Bayes theorem.
2. We have to choose a price basis for our product.

3.  $u$  is a random number in the interval  $[0.75, 1.25]$ . It can be understood as a random fluctuation to avoid strong dependencies between price and the rest of the attributes.
  4.  $v$  is a number that depends on how many times the none option, mentioned in [Section 2](#), has been chosen. The idea is to increase this number if respondents continue to choose a purchase option, in order to find a threshold purchase price. On the other hand, if the respondent still chooses the option not to buy, then we would be interested in decreasing  $v$  to find what is the minimum price for which the customer is willing to buy. If we did not include this value, it could happen that a person would choose in the whole questionnaire that he/she does not want to buy anything because the prices are high. In this case, we would not get any information.
  5.  $\text{round}$  is the function that rounds the number obtained to the nearest number among the chosen discrete prices.
- Price: regarding the possible discrete prices, we recommend to choose around 5 different values. If we choose a small number, then we will have almost no information on the non-linearity of the price utility. If we choose a big number, as we mentioned before, the complexity of the estimation grows exponentially. For our experiment we have chosen equidistant values between the extreme values:

$$\text{MinPrice} = \left[ \min \left( \sum_{i=1}^n \text{Price}_{\text{Attribute-level } i} \right) + \text{Price Basis} \right] \times 0.8$$

$$\text{MaxPrice} = \left[ \max \left( \sum_{i=1}^n \text{Price}_{\text{Attribute-level } i} \right) + \text{Price Basis} \right] \times 1.2$$

- Balance Utility: alternatives within a choice set should be equally attractive so that there should not be dominated or dominating alternatives. In the implementation, we have developed a counting system to store the number of times that each attribute-level has been chosen. With this system, we then show alternatives with the smallest difference between the scores. Where the score of a product is the sum of the scores of its attributes-levels. This system is not the optimal. Ideally, after each question, we should re-estimate the utilities of each individual using the methods that we will show in [Subsubsection 4.2.3](#). But we have not managed to find an easy-way to do so.

Additionally we have developed the code such that:

- You can choose the number of questions that you want. Ideally the number would be between 10 and 15. If the number were higher, then we take the risk that respondents may get tired and respond randomly. If the number were lower, we would barely have information about each respondent.
- You can choose the number of alternatives per question that you want. Ideally this number will be around 4 (same reasoning as before). Moreover, we incorporate the No-Choice option. This option will play an important role in our experiment. The way the implementation has been built, we have to introduce the no-choice option as the last alternative.

### 3.4 Implementation

We show here an example of a possible questionnaire, built using our implementation tool. The name of the file is `AdaptiveQuestionnaire.py` and has been implemented in `Python`. As we are going to work with an ebook data set, we choose this product to build the questionnaire. We choose that the attributes, its levels and the price priors are:

- Storage: 4GB: 10€, 8GB: 15€, 16GB: 20€, 32GB: 25€
- Screen Size: 5inch: 10€, 6inch: 15€, 7inch: 10€
- Color: white: 5€, black: 5€, silver: 5€
- Delivery time: 0days: 15€, 1day: 5€, 2days: 0€

Choosing Price Basis = 50€ and choosing 5 equidistant values we have:

- Prices: 60.0€, 78.0€, 96.0€, 114.0€, 132.0€

Choosing 8 questions and 4 alternatives we show below the whole display that will appear on the screen:

Question Number 1

The accumulated utility for option 1 is:0.0

The accumulated utility for option 2 is:0.0

The accumulated utility for option 3 is:0.0

Which option do you prefer:

Option 1: Storage 8GB, Screen 6inch, Color black, DelTime 0, Price 78.0

Option 2: Storage 32GB, Screen 5inch, Color silver, DelTime 1, Price 114.0

Option 3: Storage 4GB, Screen 7inch, Color white, DelTime 2, Price 78.0

Option 4: None

Choose one of the previous options: 3

Question Number 2

The accumulated utility for option 1 is:2.0

The accumulated utility for option 2 is:2.0

The accumulated utility for option 3 is:2.0

Which option do you prefer:

Option 1: Storage 8GB, Screen 5inch, Color white, DelTime 0, Price 78.0

Option 2: Storage 4GB, Screen 6inch, Color silver, DelTime 1, Price 78.0

Option 3: Storage 32GB, Screen 7inch, Color black, DelTime 2, Price 96.0

Option 4: None

Choose one of the previous options: 1

Question Number 3

The accumulated utility for option 1 is:3.0

The accumulated utility for option 2 is:3.0

The accumulated utility for option 3 is:3.0

Which option do you prefer:

Option 1: Storage 16GB, Screen 6inch, Color white, DelTime 0, Price 114.0

Option 2: Storage 32GB, Screen 5inch, Color black, DelTime 1, Price 78.0

Option 3: Storage 4GB, Screen 7inch, Color silver, DelTime 2, Price 96.0

Option 4: None

Choose one of the previous options: 2

Question Number 4

The accumulated utility for option 1 is:5.0

The accumulated utility for option 2 is:6.0

The accumulated utility for option 3 is:6.0

Which option do you prefer:

Option 1: Storage 16GB, Screen 5inch, Color white, DelTime 1, Price 96.0

Option 2: Storage 8GB, Screen 6inch, Color black, DelTime 0, Price 78.0

Option 3: Storage 4GB, Screen 7inch, Color silver, DelTime 2, Price 78.0

Option 4: None

Choose one of the previous options: 3

Question Number 5

The accumulated utility for option 1 is:5.0

The accumulated utility for option 2 is:5.0

The accumulated utility for option 3 is:5.0

Which option do you prefer:

Option 1: Storage 16GB, Screen 5inch, Color white, DelTime 0, Price 114.0

Option 2: Storage 32GB, Screen 7inch, Color black, DelTime 1, Price 114.0

Option 3: Storage 4GB, Screen 6inch, Color silver, DelTime 2, Price 96.0

Option 4: None

Choose one of the previous options: 4

Question Number 6

The accumulated utility for option 1 is:9.0

The accumulated utility for option 2 is:9.0

The accumulated utility for option 3 is:9.0

Which option do you prefer:

Option 1: Storage 4GB, Screen 6inch, Color black, DelTime 2, Price 78.0

Option 2: Storage 16GB, Screen 5inch, Color white, DelTime 0, Price 78.0

Option 3: Storage 32GB, Screen 7inch, Color silver, DelTime 1, Price 78.0

Option 4: None

Choose one of the previous options: 1

Question Number 7

The accumulated utility for option 1 is:8.0

The accumulated utility for option 2 is:8.0

The accumulated utility for option 3 is:8.0

Which option do you prefer:

Option 1: Storage 16GB, Screen 6inch, Color silver, DelTime 1, Price 78.0

Option 2: Storage 8GB, Screen 7inch, Color white, DelTime 2, Price 60.0

Option 3: Storage 4GB, Screen 5inch, Color black, DelTime 0, Price 114.0

Option 4: None

Choose one of the previous options: 2

Question Number 8

The accumulated utility for option 1 is:8.0

The accumulated utility for option 2 is:8.0

The accumulated utility for option 3 is:7.0

Which option do you prefer:

Option 1: Storage 16GB, Screen 5inch, Color black, DelTime 2, Price 96.0

Option 2: Storage 4GB, Screen 7inch, Color silver, DelTime 0, Price 114.0

Option 3: Storage 8GB, Screen 6inch, Color white, DelTime 1, Price 96.0

Option 4: None

Choose one of the previous options: 1

We can see how the counting system works to obtain balance utility and how products with high level features are in general shown with higher prices than the low level features. Moreover the implementation stored all the answers in a data frame. Below, we show the 4 first previously answered questions:

Id	Question	Alternative	4GB	8GB	16GB	32GB	5inch	6inch	7inch	white	black	silver	0	1	2	60.0	78.0	96.0	114.0	132.0	None	chosen
1	1	1	0	1	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0
1	1	2	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0
1	1	3	1	0	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	0	1
1	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
1	2	1	0	1	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	1
1	2	2	1	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	0	0	0
1	2	3	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	1	0	0	0	0
1	2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
1	3	1	0	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0	0
1	3	2	0	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	1
1	3	3	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0
1	3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	
1	4	1	0	0	1	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0
1	4	2	0	1	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0
1	4	3	1	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0	0	0	0	1
1	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	

Figure 4: Data Frame of stored answers.

Finally, we mention that this is not the same data format as the one that we will use in [Subsection 5.3](#). We will explain how to change it later, but it is not a difficult task.

## 4 Models

Conjoint applications assume a purchase decision model in which consumer preferences, i.e., utilities, are the central element of the choice process. The first part of this section will be devoted to the study of the decision model [1]. The second part of this section will be devoted to the estimation of the utilities given the chosen decision model.

### 4.1 Decision Model

The assumption of Conjoint Analysis on the decision model is that specific product attributes determine the individual utility evaluations and these, in turn, form the basis for the observed choice behavior. This requires 2 interdependent models: a utility model and a choice model, which translates utilities into multinomial choices. The structure can be seen in Figure 5:

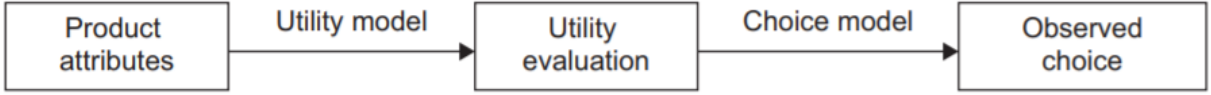


Figure 5: Structure of the decision model.

#### 4.1.1 Utility Model:

The basis for the utility model in a choice decision setting is the so-called: random utility theory<sup>2</sup>. This theory states that the overall utility  $U$  of consumer  $i$  for a product  $p$ , is the sum of a systematic component and a stochastic error component:

$$U_{i,p} = \underbrace{V_{i,p}}_{Syst.Comp} + \underbrace{\epsilon_{i,p}}_{Stoch.Comp} \quad (1)$$

The stochastic component accounts for some factors that could lead the consumer to choose not the most preferred option. Among these factors we could include the respondent fatigue or biases in the data collection.

The RUT is based on the assumption that a consumer chooses the product from a set of alternatives that has the highest utility. However, as we are dealing with stochastic components, the choices are probabilistic and not deterministic. This means that we can only state that a consumer will choose an alternative with certain probability. If we have product  $p$  and  $q$ , the chances that a respondent  $i$  chooses  $p$  over  $q$  are given by:

$$p = p(U_{i,p} > U_{i,q}) = P(V_{i,p} - V_{i,q} > \epsilon_{i,p} - \epsilon_{i,q}) \quad (2)$$

The greater the difference  $V_{i,p} - V_{i,q}$ , the more likely is that we choose product  $p$ , but this number will never be 1.

---

<sup>2</sup>RUT

The systematic utility  $V$  represents the function that translates the product attributes and their levels into part-worth utilities. The systematic utility  $V_{i,p}$  for a product  $p$  with  $N$  attributes has the general form:

$$V_{i,p} = \Phi[f_1(\nu_{i,p,1}), f_2(\nu_{i,p,2}), \dots, f_N(\nu_{i,p,N})] \quad (3)$$

where:

1.  $\nu_{i,p,n}$  are the part-worth utilities of attribute  $n$  in product  $p$  for respondent  $i$ , with  $n = 1, 2, \dots, N$ .
2.  $f_n$  is the function that represent how the individual part-worth utilities affect the whole systematic utility.
3.  $\Phi$  is the function that represent the interaction between the different part-worth utilities.

The utility model is characterised by the functions  $f_n$  that we choose. We only show here the 2 different models that we are going to work with:

### 1. Vector Model:

The vector model assumes that increasing (decreasing) the attribute level leads to a proportional positive (negative) effect in utility. This model can only be used for continuous or numeric attributes since it uses the actual numeric values of the attributes and just one utility parameter to represent the part-worth utility:

$$\nu_{i,p,n} = \beta_{i,n} \times X_{p,n} \quad (4)$$

where:

- $\nu_{i,p,n}$ : part-worth utility for attribute  $n$  in product  $p$  for respondent  $i$ .
- $\beta_{i,n}$ : utility vector for attribute  $n$  for respondent  $i$ .
- $X_{p,n}$ : numeric value of attribute  $n$  of product  $p$ .

This model can be used when treating price as continuous. However for this project, we treated price as discrete.

### 2. Part-worth Model:

The part-worth model estimates separate part-worth utilities for each level of the attribute, i.e., there is no assumed functional relationship between the attribute levels. This model is required for categorical attribute, but it can also be used for continuous attributes.

$$\nu_{i,p,n} = \sum_{m=1}^{M_i} \beta_{i,n,m} \times X_{p,n,m} \quad (5)$$

where:

- $\nu_{i,p,n}$ : part-worth utility for attribute  $n$  in product  $p$  for respondent  $i$ .

- $\beta_{i,n,m}$  part-worth utility for level  $m$  of attribute  $n$  for respondent  $i$ , with  $m = 1, \dots, M_i$ . Notice that the number of levels  $M_i$  depends on the attribute  $i$ .
- $X_{p,n,m}$  dummy variable: value 1 if product  $p$  features level  $m$  of attribute  $n$ , otherwise 0.

The part-worth model requires more parameters, but provides better model fits. Also known as a trade-off between complexity and fitting. Despite its complexity is the predominant model used in Conjoint Analysis.

These 2 models will be understood better in [Section 5](#) when dealing with actual data examples.

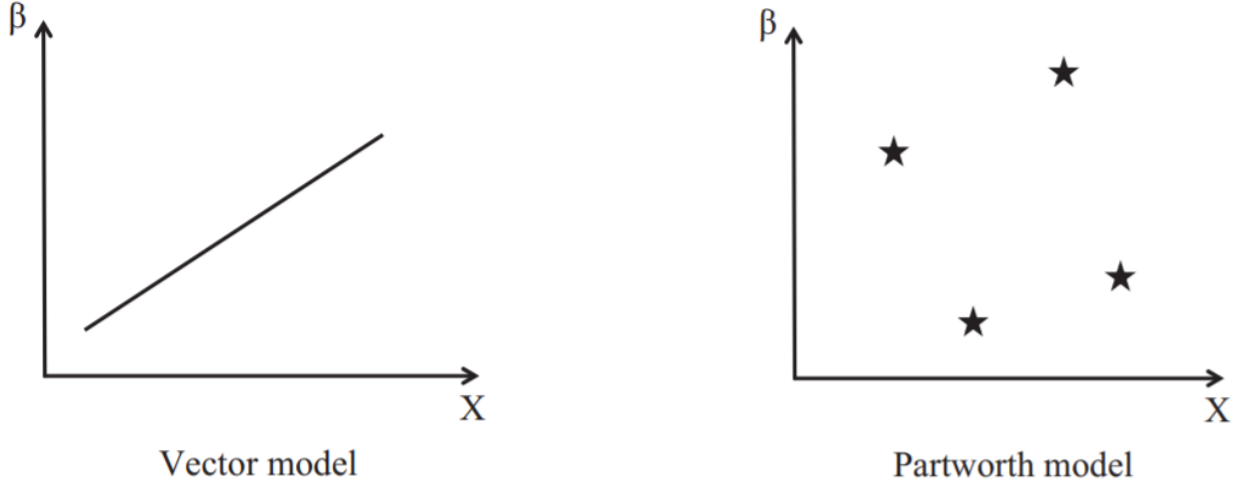


Figure 6: Vector model vs Partworth model.

Additionally, the function  $\Phi$  also plays a roll in the utility model. If the attributes are independent (there is no influence between the levels of different attributes), we use the additive utility model:

$$V_{i,p} = \sum_{n=1}^N \nu_{i,p,n} \quad (6)$$

Quite often this model is too simplistic and we will have to take into account interaction effects between attributes. This can be illustrated with a simple example:

$$\text{Drink Conjoint: } \begin{cases} \text{Type: Coke or Tea} \\ \text{Temperature: cold or hot.} \end{cases}$$

In general, people like cold Coke, cold tea and hot tea, but it is unusual to have preference for hot Coke. In this case, there is a clear interaction between the attributes that can not be neglected.



### 4.1.2 Choice Model:

The Choice Model mainly depends on the assumptions that we make about the stochastic error component. Usually in discrete choice modelling the error is assumed to be i.i.d Gumbell distributed. The reasons behind this choice are: first it works good in general with real-life problems, and second it allow us to have a closed form expression for the probability shown in (2). This assumption leads to the logistic distribution. That is, the probability of choosing an item  $p$  among a choice set  $S$  is given by:

$$p(p|S) = \frac{e^{V_p}}{\sum_{j \in S} e^{V_j}} \quad (7)$$

This models are known as the Multinomial Logistic Regression<sup>3</sup> model. Other popular choices is selecting the error normally distributed. This will lead us to the so-called probit model. But during this project we will only work with the MNL model.

---

<sup>3</sup>MNL

## 4.2 Estimation techniques

Once the decision model has been studied, we proceed with the estimation procedure. The main objective is to find estimators for the part-worth utilities  $\beta_{i,n,m}$ <sup>4</sup> or for the vector utilities  $\beta_{i,n}$ <sup>5</sup> depending on the utility model used. Moreover, depending on the assumptions that we make about these utilities we will have different models. In our case, we will study 3 different models: Multinomial Logit, Latent Class Logit, and Hierarchical Bayes Mixed Multinomial Logit<sup>6</sup>. These methods are ranked from worst to best, being the last one the most used by leading companies of the sector of Conjoint Analysis. On the other hand, there are also machine learning estimation techniques such as Support Vector Machines, but it seems that companies barely used them<sup>7</sup>.

### 4.2.1 Multinomial Logit Model

The main assumption of this model [9] is that the part-worth utilities of the product are the same for all respondents.

$$\beta_{i,n,m} = \beta_{n,m} \rightarrow \text{independent of the respondent } i \quad (8)$$

Notice that this is a quite simplistic assumption since it assumes homogeneous behaviour over the whole population. This model is based on aggregation, but it does not give any insights about the variance of preferences. To find estimates we use the Maximum Likelihood estimator. That is, having:

- $i = 1, \dots, I$  respondents.
- $t = 1, \dots, T$  choice occasions.
- $j = 1, \dots, J$  alternatives.

The Likelihood function is given by:

$$L = \prod_{i=1}^I \prod_{t=1}^T \prod_{j=1}^J \left( \frac{\exp(\mathbf{X}_{i,t,j} \cdot \boldsymbol{\beta})}{\sum_{j=1}^J \exp(\mathbf{X}_{i,t,j} \cdot \boldsymbol{\beta})} \right)^{y_{i,t,j}} \quad (9)$$

where:

- $\mathbf{X}_{i,t,j} = (X_{i,t,j})_{n,m}$  and  $\boldsymbol{\beta} = (\beta)_{n,m}$  are vectors formed by attributes  $n = 1, \dots, N$  and levels  $m = 1, \dots, M_i$
- $y_{i,t,j} = 1$  if respondent  $i$  choose alternative  $j$  in choice occasion  $t$ , and 0 otherwise.

The estimators of the part-worth utilities can be then found by maximizing the previous function with respect to the  $\beta_{n,m}$ .

---

<sup>4</sup>remember, i-th responden, n-th attribute, m-th level.

<sup>5</sup>if treating the n-th attribute as continuous.

<sup>6</sup>MNL, LC and HB-MMNL respectively.

<sup>7</sup>Less accurate results than Hierarchical Bayes model.

### 4.2.2 Latent Class Model

The assumption of aggregate-level analyses that consumers are all identical is usually too restrictive. Considering consumer heterogeneity with advanced estimation techniques is therefore beneficial in reducing the error term [10]. The following method that we propose is the Latent Class model. This model assumes that the heterogeneity of respondents in the sample can be accounted for by a finite number of homogeneous segments. These segments are latent, i.e., each respondent belongs to the segments with a certain probability:

$$\beta_{i,n,m} = \beta_{s,n,m} \text{ with probability } \omega_{i,s} \rightarrow s = 1, \dots, S \text{ number of different segments} \quad (10)$$

In such a way that if a consumer differs in his/her choice behaviour from the part-worth utilities of the respective segment, this is reflected by a lower probability to belong to this segment.

This model therefore aims to estimate different utilities  $\beta_s$  for these segments. Before the estimation starts, we need to specify the number of different segments. Then we use an algorithm to maximize the Likelihood function and obtain the utilities of the different segments. The Likelihood function is given by:

$$L = \prod_{i=1}^I \sum_{s=1}^S \omega_{i,s} \prod_{t=1}^T \prod_{j=1}^J \left( \frac{\exp(\mathbf{X}_{i,t,j} \cdot \beta_s)}{\sum_{j=1}^J \exp(\mathbf{X}_{i,t,j} \cdot \beta_s)} \right)^{y_{i,t,j}} \quad (11)$$

where the only difference with respect to (9) are the probabilities  $\omega_{i,s}$  and the segment utilities  $\beta_s$ .

This model is quite useful since is in line with discovering market segments with distinct preferences that are an attractive target group for a company's market offerings. However, the computation required for the algorithm increases exponentially with the number of segments.

### 4.2.3 Hierarchical Bayes Mixed Multinomial Model

Lastly, we are going to deal with the most powerful of these 3 methods: Hierarchical Bayes Mixed Multinomial model. First we will discuss the Mixed Multinomial Model [10], and later we will discuss Bayesian techniques [11] to compute individual part-worth utilities.

We already saw that the Latent Class model was able to account for heterogeneity. However, the way of doing it was far from complex. There are other methods that are much more realistic and are able to model population heterogeneity in an accurate form. The way of doing so, is to consider the part-worth utilities as random variables that varies across the population. That is:

$$\beta_{i,n,m} \sim f(\beta_{n,m}|\theta) \quad (12)$$

where  $f(\beta_{n,m}|\theta)$  is a continuous density function and  $\theta$  are the parameters of the distribution.

In this setting the Likelihood function becomes:

$$L = \prod_{i=1}^I \int \prod_{t=1}^T \prod_{j=1}^J \left( \frac{\exp(\mathbf{X}_{i,t,j} \cdot \beta)}{\sum_{j=1}^J \exp(\mathbf{X}_{i,t,j} \cdot \beta)} \right)^{y_{i,t,j}} f(\beta|\theta) d\beta \quad (13)$$

Notice that we can choose the density function that we want to model the heterogeneity. Maximizing the previous expression with respect to  $\theta$  using Monte Carlo approximation techniques, allow us to obtain maximum likelihood estimators  $\hat{\theta}$ . If we plug these estimators into the density function, we obtain a specific function that shows the spread of the respondents' preferences.

One popular option is to use the multivariate normal distribution as the density function:

$$\beta_i \sim N(\beta, \Sigma), \quad (14)$$

where  $\beta$  is the vector of means and  $\Sigma$  is the covariance matrix. This choice will lead us to the so-called Mixed Multinomial Logit model. Using the previously mentioned procedure we can obtain estimators  $\hat{\beta}$  and  $\hat{\Sigma}$  that characterizes the heterogeneity.

Once we have identify the part-worth distribution, we ask ourselves: How can we obtain the individual part-worth utilities?. One may wonder why we do not directly use the MNL model applied to each individual. The main problem is that we usually do not have enough data to get accurate estimators, leading to overfitting and almost zero predictive ability. The way of solving this problem is incorporating prior knowledge to the task, that allow us to get robust estimates even if we have few observations.

The procedure that we will use to incorporate prior knowledge is known as Hierarchical Bayes. The main idea is to use the Mixed Multinomial Logit as a basis to draw conditional estimates for each individual given respondent's choice data. The HB model therefore consists of two coupled layers:

- First layer describes the choice probabilities given the individual part-worth utilities, i.e., the MNL model described by (7). It is characterised by the probability:  $p(y_{i,h}|\beta_i)$ , where  $y_{i,h}$  set of answers.
- Second layer describes the continuous distribution of the part-worth utilities across the population, i.e, the MMNL model, described by (14). It is characterised by the probability:  $p(\beta_i)$

The objective now is to compute the density function of the individual part-worth utilities given individual choices, that is  $p(\beta_i|y_{i,h})$ . The key step that the HB uses for this computation is the Bayes Theorem, such that:

$$p(\beta_i|y_{i,h}) = \frac{p(y_{i,h}|\beta_i)p(\beta_i)}{p(y_{i,h})} = \frac{p(y_{i,h}|\beta_i)p(\beta_i)}{\int p(y_{i,h}|\beta_i)p(\beta_i)d\beta_i} \propto p(y_{i,h}|\beta_i)p(\beta_i) \quad (15)$$

The main problem is that there is no close form expression for this probability. Instead, we can sample from this distribution. To do so, we need to use the Metropolis-Hasting algorithm, that is a type of Markov-chain Monte Carlo method. Simulating from this distribution and using individual respondent data  $y_i$ , we can therefore obtain individual-specific values for  $\beta_i$ . Once we have these estimators for the whole population, we can generate box plots or we can "rebuild" a more accurate distribution of  $\beta$ .

## 5 Data Analysis

In this section we are going to apply the estimation methods shown in [Section 4](#) to a real data set of E-books preferences<sup>8</sup>. We show the results obtained for each of the estimation models studied above. With the results of the last model, we will show later in [Section 6](#) the possible applications that Conjoint Analysis has. Moreover, we perform a model validation applied to the HB-MMNL. To carry out the data analysis we have used the software R, in particular the libraries: `mlogit` [9], `gmm1` [10] and `ChoiceModelR` [12]. We have also tried to implement it in Python using the libraries: `PyLogit`, `PandasBiogeme` and `bayes_mxl`, but with less success than the former.

### 5.1 Multinomial Logit Model

In this subsection we are going to show the results when applying the model explained in [Subsubsection 4.2.1](#). As we mentioned during [Subsubsection 4.1.1](#), we can obtain different results depending on the models that we choose. In this section we only show the results obtained for the part-worth model, treating price as discrete and considering independent attributes. However we include the code for treating price as continuous and allow interactions between attributes. The file used to carry out this subsection is `ChoiceBasedConjointR.R`. The results that we obtain are:

```
Call: mlogit(formula = Selected ~ Storage_4GB + Storage_8GB + Screen.size_5inch +
Screen.size_6inch + Color_black + Color_white + Price_79 +
Price_99 + Price_119 + None | 0, data = abc, method = "nr")
```

Frequencies of alternatives:

```
      1      2      3      4
0.2830 0.2925 0.2925 0.1320
```

nr method

4 iterations, 0h:0m:0s

$g'(-H)^{-1}g = 2.05E-07$

gradient close to zero

Coefficients :

	Estimate	Std. Error	z-value	Pr(> z )
Storage_4GB	-0.3892543	0.0417537	-9.3226	< 2.2e-16 ***
Storage_8GB	-0.0512003	0.0387297	-1.3220	0.1862
Screen.size_5inch	-0.0493755	0.0387547	-1.2740	0.2026
Screen.size_6inch	0.4467212	0.0361670	12.3516	< 2.2e-16 ***
Color_black	-0.0022636	0.0382276	-0.0592	0.9528
Color_white	0.2400119	0.0366608	6.5468	5.877e-11 ***
Price_79	0.8408738	0.0454489	18.5015	< 2.2e-16 ***
Price_99	0.2857849	0.0468274	6.1029	1.041e-09 ***

---

<sup>8</sup><http://www.preferencelab.com/data/CBC.R>.

```

Price_119      -0.2842908  0.0524903 -5.4161 6.092e-08 ***
None          -0.5318355  0.0686299 -7.7493 9.326e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Log-Likelihood: -2277.8

```

Plotting these part-worth utilities to have a better visual representation:

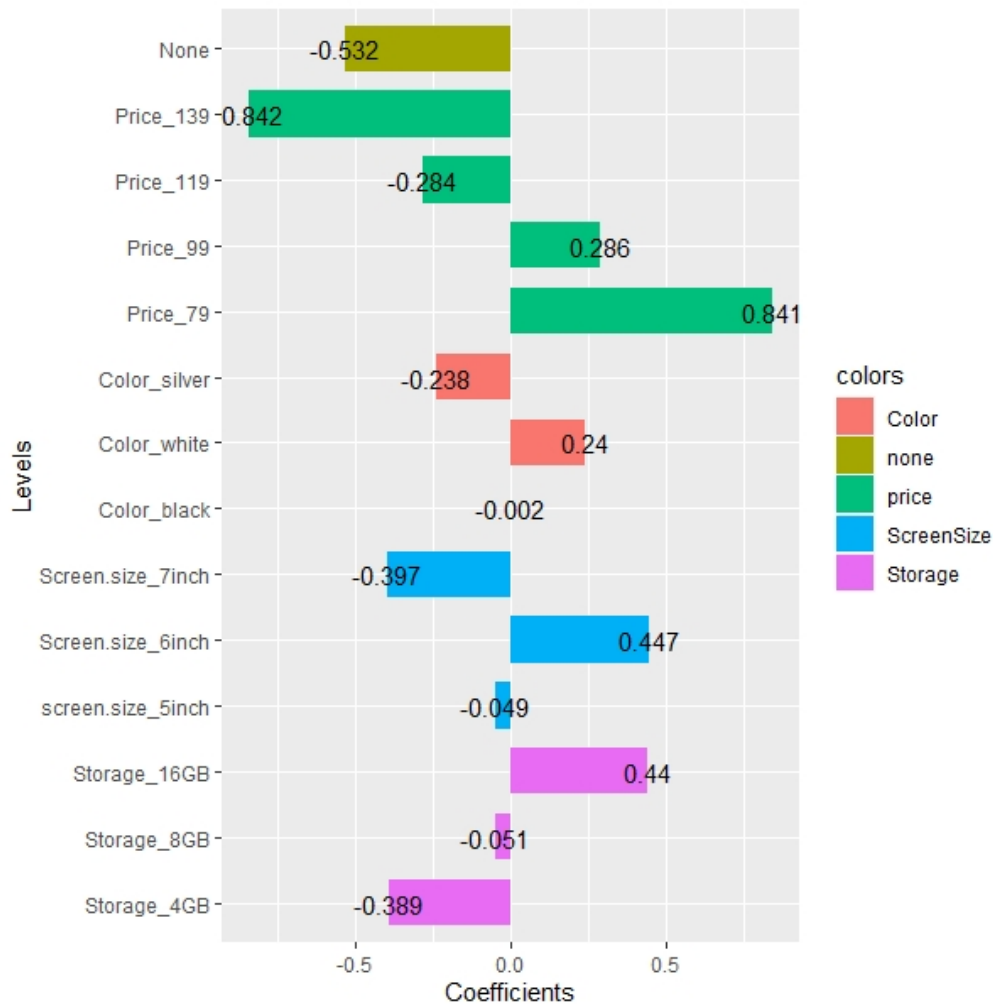


Figure 7: Part-worth utilities for the Multinomial Logit Model.

These values help us to have an overview of the population. But we should never carry out applications with them, since they are based on aggregation and they can lead us to results far away from reality.

Additionally, as we mentioned in application 1 in [Section 1](#), we can compute which product attribute will be the most relevant when making purchases. We show in [Figure 8](#) that the relative attribute importance is given by:

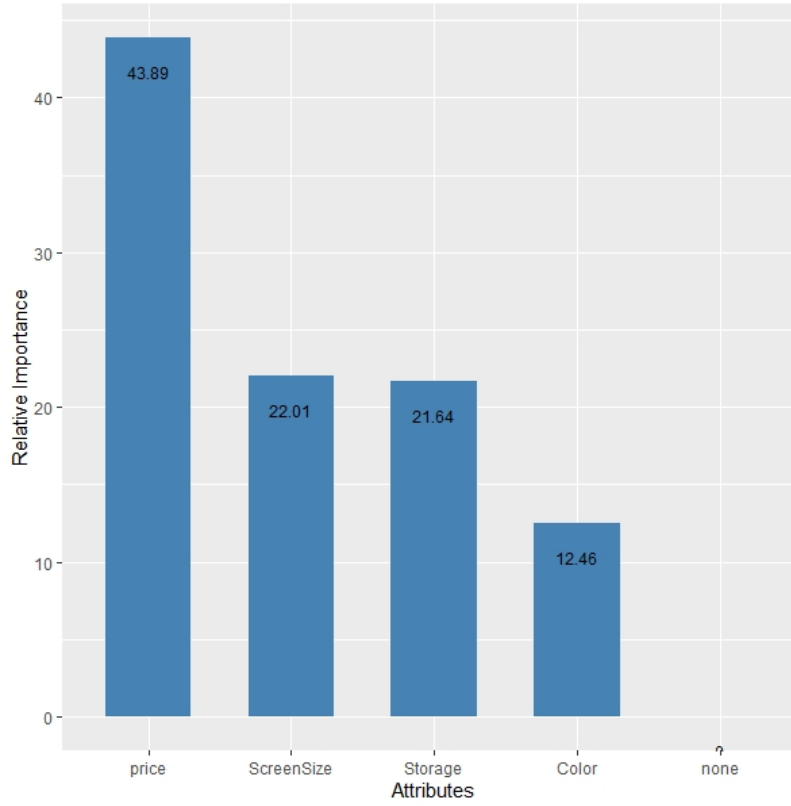


Figure 8: Relative Attribute Importance for the MNL model.

In general, we see that the price of the product is the most important aspect for consumers at the time of purchase. In second place we find Storage and Screen Size. Checking [Figure 7](#), we see that the part-worth utilities grows as Storage grows. Therefore, it could be a good idea to release a new ebook version with 32GB of Storage. On the other hand, the most preferred screen size is 6inch, which is the intermediate value. Hence, 8inch and 4 inch screen would have lower utilities than 7inch and 5inch screen respectively. So it would not make sense to release new products with these features.

## 5.2 Latent Class Model

The second model that we treat is the Latent Class Model [10], previously mentioned in [Subsubsection 4.2.2](#). Ideally we would run this model for different number of segments and we would choose the one with the least AIC (Akaike Information Criterion). The main problem is that the computational times grows exponentially as we increase the number of segments. We run the model for  $S = 2, \dots, 6$  obtaining the lowest AIC for  $S = 3$ . The file used to carry out this subsection is ChoiceBasedConjointR.R. The results can be seen below.

Call:

```
gmm1(formula = Selected ~ Storage_4GB + Storage_8GB + Screen.size_5inch +
      Screen.size_6inch + Color_black + Color_white + Price_79 +
      Price_99 + Price_119 + None | 0 | 0 | 0 | 1, data = abc,
      subset = 1:8000, model = "lc", Q = 3, panel = TRUE, method = "NR")
```

Frequencies of categories:

```
      1      2      3      4
0.2830 0.2925 0.2925 0.1320
```

The estimation took: 0h:1m:20s

Coefficients:

	Estimate	Std. Error	z-value	Pr(> z )	
class.1.Storage_4GB	-1.209624	0.232735	-5.1974	2.021e-07	***
class.1.Storage_8GB	-0.089556	0.146031	-0.6133	0.5396980	
class.1.Screen.size_5inch	-0.958944	0.227460	-4.2159	2.488e-05	***
class.1.Screen.size_6inch	0.109197	0.128953	0.8468	0.3971082	
class.1.Color_black	-0.070518	0.145015	-0.4863	0.6267656	
class.1.Color_white	0.260554	0.138150	1.8860	0.0592931	.
class.1.Price_79	1.438918	0.252482	5.6991	1.205e-08	***
class.1.Price_99	-0.200198	0.188363	-1.0628	0.2878571	
class.1.Price_119	-0.544020	0.221928	-2.4513	0.0142325	*
class.1.None	-0.883495	0.394478	-2.2397	0.0251134	*
class.2.Storage_4GB	-0.544254	0.105123	-5.1773	2.251e-07	***
class.2.Storage_8GB	0.121511	0.098643	1.2318	0.2180150	
class.2.Screen.size_5inch	0.818071	0.107385	7.6181	2.576e-14	***
class.2.Screen.size_6inch	0.010053	0.107994	0.0931	0.9258305	
class.2.Color_black	-0.595372	0.117239	-5.0783	3.809e-07	***
class.2.Color_white	1.251318	0.130003	9.6253	< 2.2e-16	***
class.2.Price_79	0.922828	0.125093	7.3771	1.616e-13	***
class.2.Price_99	0.413519	0.130017	3.1805	0.0014702	**
class.2.Price_119	-0.250579	0.126049	-1.9879	0.0468178	*
class.2.None	-2.386755	0.415636	-5.7424	9.334e-09	***
class.3.Storage_4GB	-0.323894	0.061794	-5.2415	1.592e-07	***
class.3.Storage_8GB	-0.101116	0.060305	-1.6767	0.0935923	.



```

class.3.Screen.size_5inch -0.242274 0.065308 -3.7097 0.0002075 ***
class.3.Screen.size_6inch 0.856361 0.066442 12.8887 < 2.2e-16 ***
class.3.Color_black 0.301099 0.060086 5.0111 5.411e-07 ***
class.3.Color_white -0.238094 0.064945 -3.6661 0.0002463 ***
class.3.Price_79 1.007666 0.075498 13.3470 < 2.2e-16 ***
class.3.Price_99 0.423549 0.076511 5.5358 3.098e-08 ***
class.3.Price_119 -0.319175 0.085802 -3.7199 0.0001993 ***
class.3.None 0.115726 0.094287 1.2274 0.2196770
(class)2 0.465173 0.089018 5.2256 1.736e-07 ***
(class)3 1.339121 0.084546 15.8389 < 2.2e-16 ***

```

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by Newton-Raphson maximisation

Log Likelihood: -2056.6

Number of observations: 2000

Number of iterations: 74

Exit of MLE: successive function values within relative tolerance limit (reltol)

Notice that the Log-likelihood is higher than in the previous model as expected. Plotting these results we get:

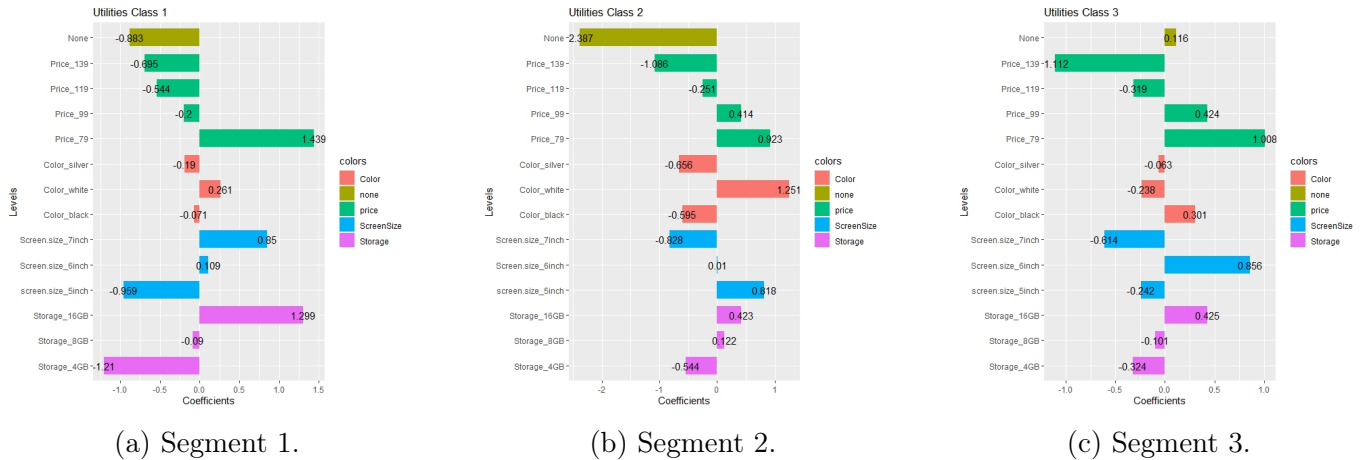


Figure 9: Part-worth utilities for the Latent Class Model with S=3.

This model is much more informative than the previous one since it allow us to identify the behaviour of the different population segments. However, the results with the R software do not inform us which segment each person is associated to. We need to compute individual part-worth utilities to carry out the previous task.

### 5.3 Hierarchical Bayes Mixed Multinomial Logit Model

This section is split into 2 parts as we previously explained in [Subsubsection 4.2.3](#). First, we estimate a Mixed Multinomial Logit Model. These results will be used as priors for the utilities. Finally, we carry out a Metropolis Hasting algorithm to sample from the conditional distribution given by (15) and obtain individual part-worth utilities.

In this section we show the results obtained when treating price as a discrete variable and allowing interaction through the covariance matrix  $\Sigma$ . We also include the code when treating price as continuous and when we neglect correlations. The file used is `ChoiceBasedConjointR.R`. Using the `gmn1` package, we obtain that the estimators for the mean and variance of the normal random variable in (14) are:

Call:

```
gmn1(formula = Selected ~ Storage_4GB + Storage_8GB + Screen.size_5inch +  
      Screen.size_6inch + Color_black + Color_white + Price_79 +  
      Price_99 + Price_119 + None | 0, data = abc, subset = 1:8000,  
      model = "mix1", ranp = c(Storage_4GB = "n", Storage_8GB = "n",  
        Screen.size_5inch = "n", Screen.size_6inch = "n", Color_black = "n",  
        Color_white = "n", Price_79 = "n", Price_99 = "n", Price_119 = "n"),  
      R = 10, correlation = TRUE, panel = TRUE, method = "bfgs")
```

Frequencies of categories:

	1	2	3	4
	0.2830	0.2925	0.2925	0.1320

The estimation took: 0h:3m:26s

Coefficients:

	Estimate	Std. Error	z-value	Pr(> z )
None	-0.027815	0.077807	-0.3575	0.7207244
Storage_4GB	-0.606116	0.060153	-10.0763	< 2.2e-16 ***
Storage_8GB	-0.034707	0.052706	-0.6585	0.5102161
Screen.size_5inch	-0.033415	0.063304	-0.5279	0.5975999
Screen.size_6inch	0.542489	0.060563	8.9574	< 2.2e-16 ***
Color_black	-0.050134	0.059540	-0.8420	0.3997701
Color_white	0.331606	0.065974	5.0263	5.001e-07 ***
Price_79	1.214381	0.069176	17.5550	< 2.2e-16 ***
Price_99	0.450344	0.069457	6.4838	8.945e-11 ***
Price_119	-0.319770	0.076721	-4.1679	3.074e-05 ***

Covariance Matrix:

	Storage_4GB	Storage_8GB	Screen.size_5inch	Screen.size_6inch
Storage_4GB	0.122549085	0.05562346	0.07270592	0.08449033
Storage_8GB	0.055623460	0.07613171	0.15389216	-0.06721862
Screen.size_5inch	0.072705922	0.15389216	0.67391714	-0.23878177
Screen.size_6inch	0.084490334	-0.06721862	-0.23878177	0.45148734
Color_black	0.057785300	-0.05268399	-0.43079964	0.25479500
Color_white	-0.104975031	0.10405096	0.47123600	-0.34506099
Price_79	-0.007240816	-0.06570533	-0.25027582	0.16468137
Price_99	-0.023678856	-0.05654504	0.09762568	0.08094076
Price_119	-0.027273607	0.00197706	0.19220555	-0.04208834

	Color_black	Color_white	Price_79	Price_99	Price_119
Storage_4GB	0.05778530	-0.10497503	-0.007240816	-0.02367886	-0.02727361
Storage_8GB	-0.05268399	0.10405096	-0.065705329	-0.05654504	0.00197706
Screen.size_5inch	-0.43079964	0.47123600	-0.250275820	0.09762568	0.19220555
Screen.size_6inch	0.25479500	-0.34506099	0.164681367	0.08094076	-0.04208834
Color_black	0.44732509	-0.53673590	0.143943458	-0.12140899	-0.10618364
Color_white	-0.53673590	0.86111486	-0.140198063	0.04528233	0.09336042
Price_79	0.14394346	-0.14019806	0.206470455	0.07331416	0.04214536
Price_99	-0.12140899	0.04528233	0.073314155	0.32873980	0.23259520
Price_119	-0.10618364	0.09336042	0.042145364	0.23259520	0.40636827

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Optimization of log-likelihood by BFGS maximization

Log Likelihood: -2045.4

Number of observations: 2000

Number of iterations: 261

Exit of MLE: successful convergence

Simulation based on 10 draws

The log-likelihood is slightly higher than in the previous model. On the other hand, the model complexity is way higher than the Latent Class, and also the computational time. This model alone does not perform good in this case since the normality assumption is quite strict. However, when combined using the Hierarchical Bayes procedure is quite powerful. In this case, the normality is desirable due to its closed form when using Bayes Theorem.

To carry out the Hierarchical Bayes Mixed Multinomial Logit, we are going to use the package `ChoiceModelR` [12]. This package includes a function that implements a Markov Chain Monte Carlo algorithm for the part-worth utilities estimation. In particular, it uses a hybrid Gibbs Sampler with a random walk metropolis step. This package is quite powerful and it greatly outperforms Python packages. Among the features that make it so convenient are:

- It allow us to specify which attributes are continuous and which are discrete. This is done in a simple way through a vector of 0's and 1's. The value 0 indicates that the attribute in that

position is categorical, and the value 1 is associated with continuous attributes. The python package `bayes_mx1` works good with continuous attributes, but it did not perform good with categorical attributes. Since Conjoint Analysis projects mainly use categorical attributes, we decided to go on with the `ChoiceModelR` package.

- It allow us to introduce constraints on the part-worth utilities for levels within the same attribute. This fact let us introduce prior knowledge that enhance the estimation. Applied to our example, we can specify that for every individual we have:

$$u_{\text{Storage:4GB}} < u_{\text{Storage:8GB}} < u_{\text{Storage:16GB}}$$

$$u_{\text{Price:79}} < u_{\text{Price:99}} < u_{\text{Price:119}} < u_{\text{Price:139}}$$

- It allow us to introduce the None option to perform the estimation.
- It allow us to introduce the prior mean vector of (14) already computed in the Mixed Multinomial Logit Model. This fact enables a faster convergence of the algorithm to good estimators, which reduces the computational time required.
- It allow us to introduce a vector of covariates for each individual. Among these covariates we could have demographic data (not recommended to include this data in the estimation). Additionally, as we mentioned in Section 3 we could also introduce the answers to the behavioral framework. According to [7] this introduction enhance the estimation and facilitates the subsequent grouping of customers.

The only inconvenience is that we have to change the format in which the data is stored. When building the questionnaire in Section 3, we stored the data in a long format, where each attribute level had its own column. For using this package we need to have only one column per attribute and code the different levels of the attributes. The columns of this case are:

UnitID, Set, Alternative, Attribute 1, Attribute 2, ..., Attribute n, y

The first column contains the ID of the unit (e.g. customer or survey respondent). The second column contains the choice set number for the unit, where each choice set is an observation for the unit. The third column contains the alternative number within the choice set. The last column contains the choice. This variable takes a nonzero value only in the first row of the choice set data, and takes a value from 1 to the number of alternatives in the choice set.

This procedure might seem a bit confusing, but it is more illustrative to see an example where we carry it out. We include the code for changing the data format applied to our example. Customizing this procedure for the chosen data is then not difficult. The file that we used to carry out this computations is `HierarchicalBayesEbooks.R`.

Below, we show the results we get when running the code.

Attribute	Type	Logit Data Levels
-----------	------	-------------------

```

-----
Attribute 1    Part Worth    3
Attribute 2    Part Worth    3
Attribute 3    Part Worth    3
Attribute 4    Part Worth    4

```

10 parameters to be estimated (including 'None').

200 total units. Average of 4 alternatives in each of 10 sets per unit. 2000 tasks.

Table of choice data pooled across units:

```

Choice Count  Pct.
-----

```

```

  1    566    28.3%
  2    585    29.25%
  3    585    29.25%
  4    264    13.2%

```

#### MCMC Inference for Hierarchical Logit

```

=====

```

```

Total Iterations:      8000    Constraints in effect.
Draws used in estimation: 2000    Draws are to be saved.
Units:                  200    Prior degrees of freedom:  5
Parameters per unit:    10    Prior variance:           2

```

Constraints in effect.

Total Time Elapsed: 0:30

Log Likelihood: -1243.811

Writing estimated unit-level betas to Rbetas.csv in the working directory

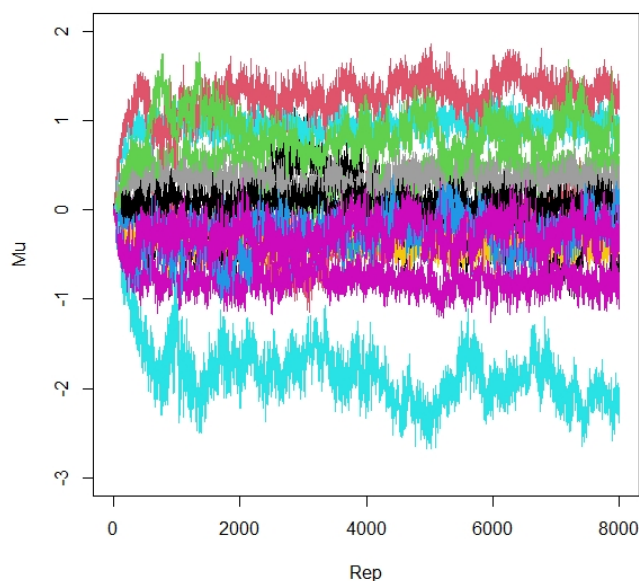


Figure 10: Markov Chain Monte Carlo Algorithm

We see how powerful this method is, since the Log-likelihood is far greater than the previous 2 models and the computational time is way less. In addition, this program stores a data frame of the individual part worth utilities in the working directory as `Rbetas.csv`. So despite having an R implementation, we can import this data frame into Python and carry out all the applications in [Section 6](#) in this software. Finally, we show in [Figure 11](#) the boxplots of the individuals utilities and in [Figure 12](#) the 'rebuilt' more accurate distribution of  $\beta$  for the different attributes.

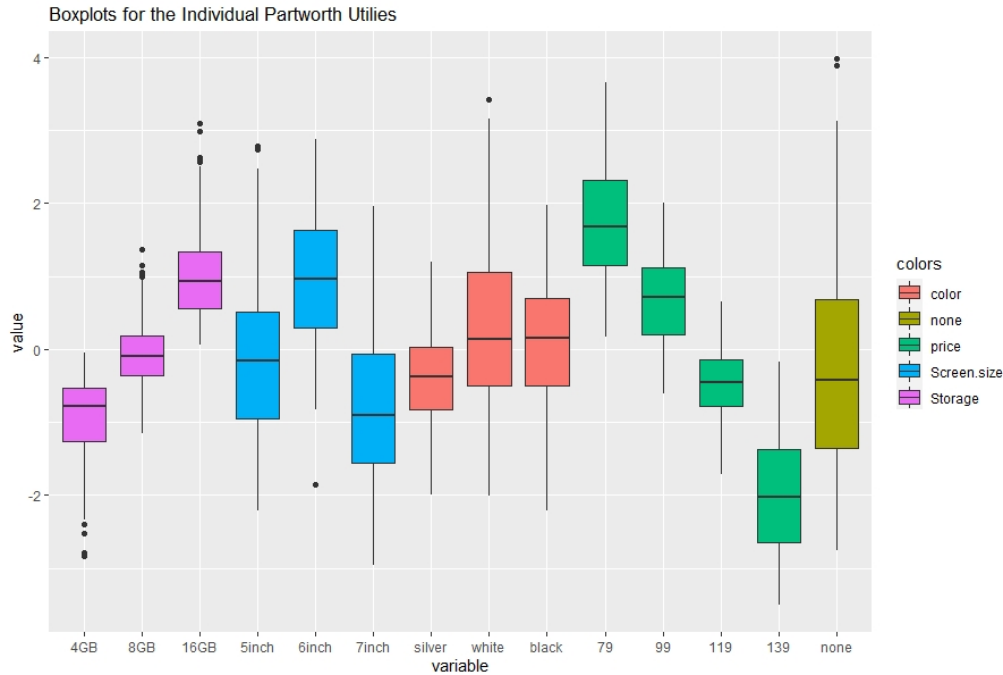


Figure 11: Boxplots of the individual utilities obtained using a HB-MMNL model.

These 2 figures might not be not really informative. However, the fact of having some estimators for the individual part-worth utilities will allow us to carry out interesting applications in [Section 6](#).

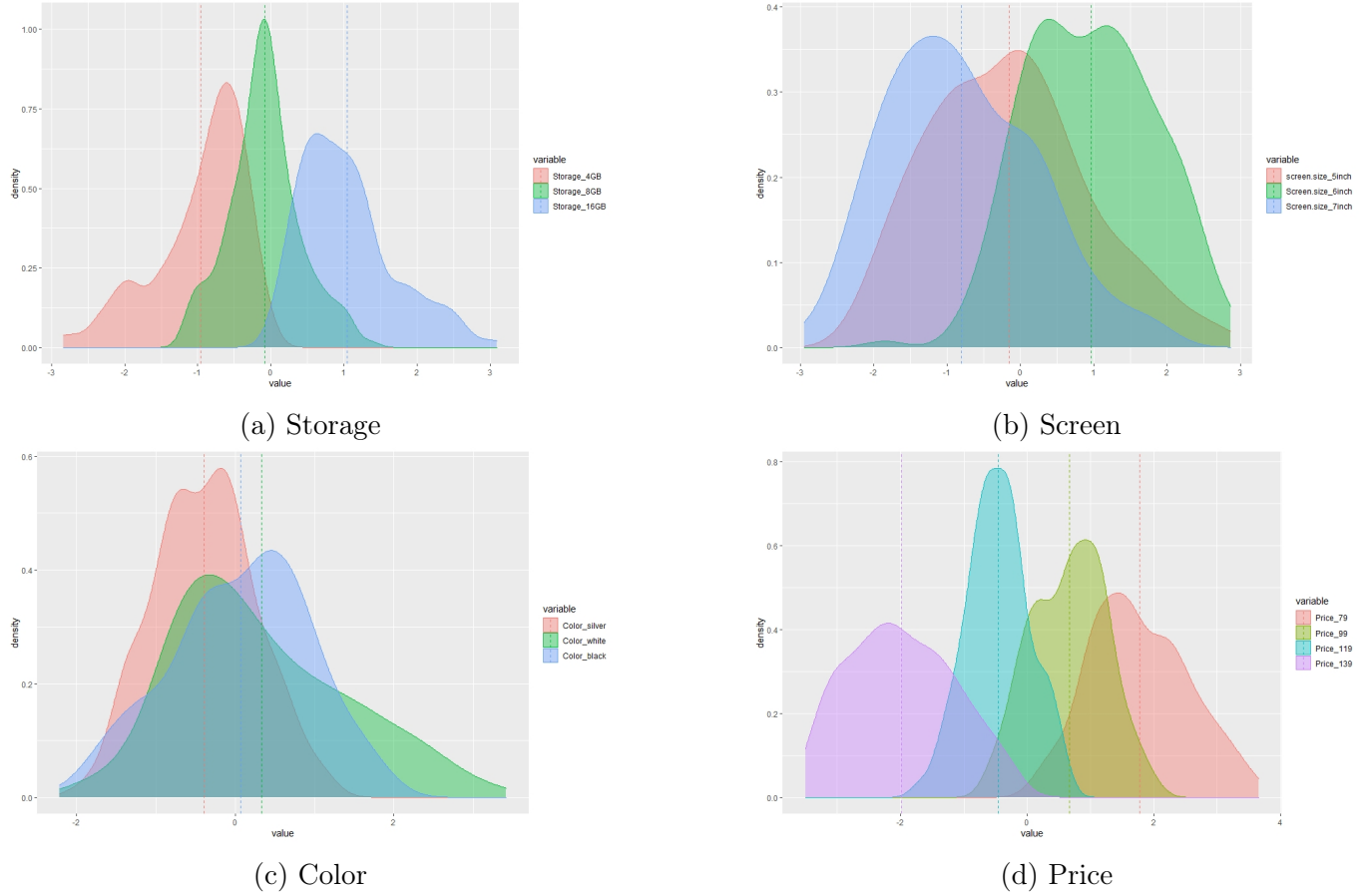


Figure 12: Rebuilt densities for the utilities of the different attribute levels.

## 5.4 Model Validation

Once the HB-MMNL model is built and we have got some estimators, we want to know how good the model is [13]. We have already obtained the Log-likelihood as a measure of goodness of fit. However, we also want to compute the predictive accuracy. To do so we split the data into 2 groups:

- Training data set → We use it to train the model and get some estimators.
- Testing data set → We use it to asses performance of the trained model.

The chosen ratio between these data sets is 80%-20%. As in the E-books data set we have 10 answered questions per respondent, we choose the first 8 questions for training and the last 2 questions for testing. We evaluate 2 different models<sup>9</sup>

- First-Choice Model → We assume that the respondent will choose the option with highest utility. This model is suitable for high-risk or seldom purchased products: education, life insurance, pension plans, etc, where the stakes are high, life-or-death situation such as chronic

<sup>9</sup><https://conjointly.com/guides/preference-share-models/>

medication options, and it will use up a large portion of respondent's disposable income. In high-risk simulations, people tend to select the single most preferred product even when it is only marginally more preferable than the next best product.

- Choice Model → We assume that the respondent will choose the option with certain probability given by (7). We simulate this procedure 20 times and we store the median and the variance. This model is appropriate for low-risk or frequently purchased products: FMCG, software, etc, where there is a strong impulse buying behaviour in play. This model is applicable in the vast majority of applications.

The file that we have used to carry out this computations is `PerformPredAccurHB.R`. The results that we obtain are given by:

Table 1: Predictive accuracy for the HB-MMNL.

HB-MMNL	Maximum Utility	Choice Model
Mean Predictive Accuracy	0.68	0.5887
St.Dev Predictive Accuracy	0	0.02338

The value for the First-Choice model is high in the sense that as we have 4 alternatives per choice set, the predictive accuracy of answering random would be 0.25. Comparing this number with our results, the HB-MMNL model is quite powerful. Even more taking into account that we are modelling human purchasing behaviour, that is in general difficult to predict.

Additionally, it can also be interesting to carry out the same procedure using the Multinomial Logit model applied to each individual. We tried to use the R packages but they could not perform the estimation due to singularity problems, as the sample size was too small. Instead, we use the Python `pylogit` package. The results that we obtain for this case are:

Table 2: Predictive accuracy for the MNL applied for each respondent

MNL	Maximum Utility
Mean Predictive Accuracy	0.2825

These results are extremely poor, since they are slightly better than picking random choices. For the estimation we obtained utilities of the order -18 or +14, so we could foresee how low the predictive ability would be. These results showed what we stated in [Subsubsection 4.2.3](#):

*"One may wonder why we do not directly use the MNL model applied to each individual. The main problem is that we usually do not have enough data to get accurate estimators, leading to overfitting and almost zero predictive ability."*

Therefore we can see how important is to carry out the Hierarchical Bayes procedure for the estimation of the part-worth utilities.



## 6 Applications

Finally, we study the possible applications of Conjoint Analysis [4]. This section is possibly the most important from a practical point of view and the final aim of the project. However, to obtain good results we need to bear in mind the other steps of the process. Building a good questionnaire in [Section 3](#) and properly analysing the data in [Section 5](#) are the fundamental pillars to obtain accurate application results. We split this section into 2 parts. In [Subsection 6.1](#) we study how can we cluster different customers based on their preferences. In [Subsection 6.2](#) we develop a Market Simulator and we carry out relevant simulations from a company marketing perspective. This whole implementation has been carried out in Python, using the `RBetas.csv` file previously obtained.

### 6.1 Customer Clustering

Once the individual utilities have been found, we can cluster the customer into different segments based on their purchasing behaviour. This procedure is similar to the one performed in the Latent Class model, but it is computationally less expensive and the results are more accurate. This subsection is devoted to answer 2 questions:

- Are there different market segments that differ in terms of certain preferred product attributes?
- Do segment members have common features?

With the data that we have, we can only answer the first question. But we also include some remarks, guidelines to overview the second one.

To cluster the customers we use the K-Means algorithm. The reason behind this choice is how powerful this algorithm is implemented in the `sklearn` library. We could have used other methods such as BIRCH or DBSCAN. In fact, we also implement the Agglomerative Clustering method to see if there are any significant differences. The file that we have used to carry out these computations is `ClusteringCustomers.py`.

We use the elbow method to determine the number of clusters in a data set. This method consists of plotting the explained variation as a function of the number of clusters, and picking the elbow of the curve as the number of clusters to use. We can observe in [Figure 13](#) that according to this method, we should choose  $n = 3$  segments for our data-set

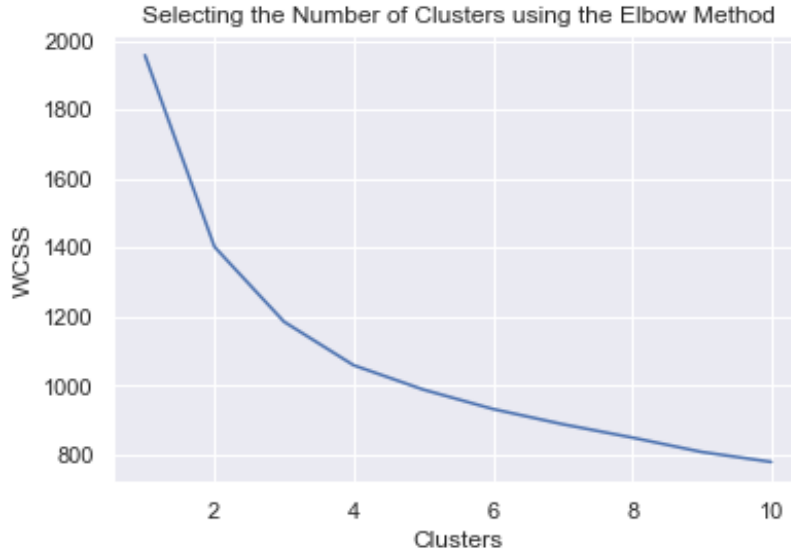
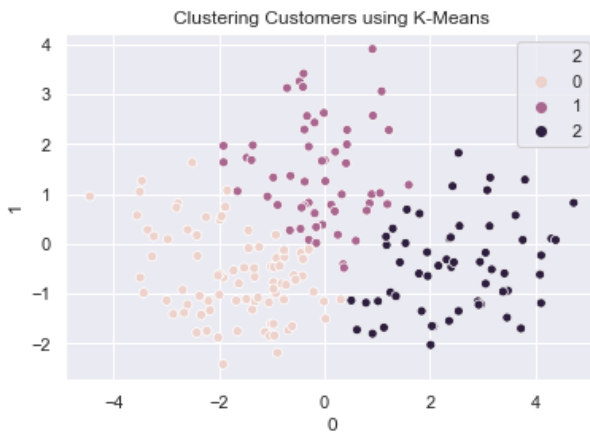
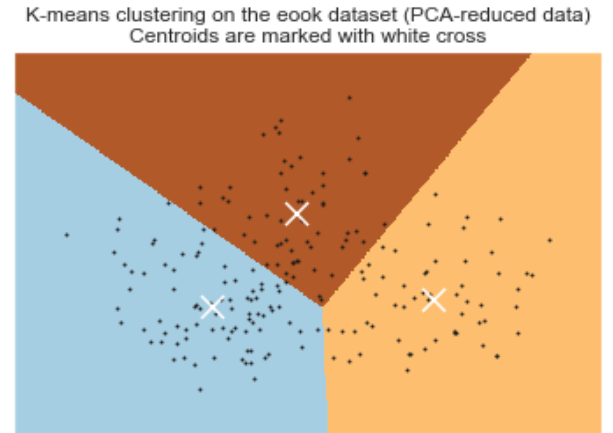


Figure 13: Elbow method

Choosing  $n = 3$  segments, we fit a K-Means model to our data. We include in the implementation the commands to find the center of the clusters and the inertia. Moreover, we use Principal Component Analysis to reduce the space dimension to 2, and check how well the clusters are split.



(a) PCA cluster K-Means



(b) PCA cluster K-Means centroids

Figure 14: Principal Component Analysis  $n = 3$ , using K-Means Clustering.

We can see that there is no clear separation between the 3 segments. On the other hand, we observe points that are very distant from each other, so the behaviors will be very different. Therefore treating these individuals differently is imperative to the company's performance.

Below, we show the boxplots of the part-worth utilities for the different clusters. [Figure 15](#) is much more informative than the previous one and allows us to draw some conclusions about the different cluster preferences.

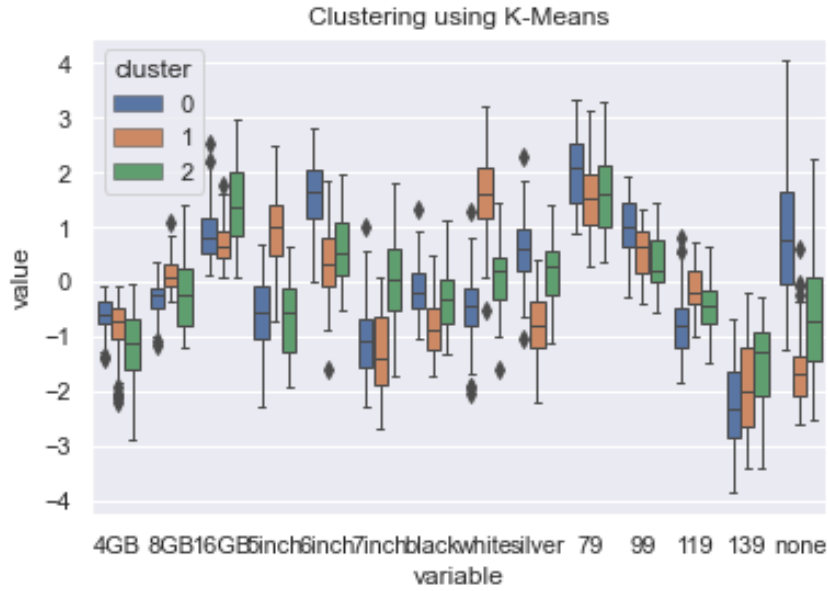


Figure 15: Boxplots of the individual part-worth utilities for the different cluster using K-Means.

For instance, cluster 0 give more importance to price than the other 2 clusters. Moreover, this cluster also prefers 6-inch screens by far, and are the most reluctant to buy ebooks (highest none utility). We will elaborate further in [Subsubsection 6.2.3](#), where we will see the great implications of this clustering technique. Finally, we check in [Figure 16](#) that the results obtained using the Agglomerative Hierarchical Clustering are approximately the same as the K-Means results.

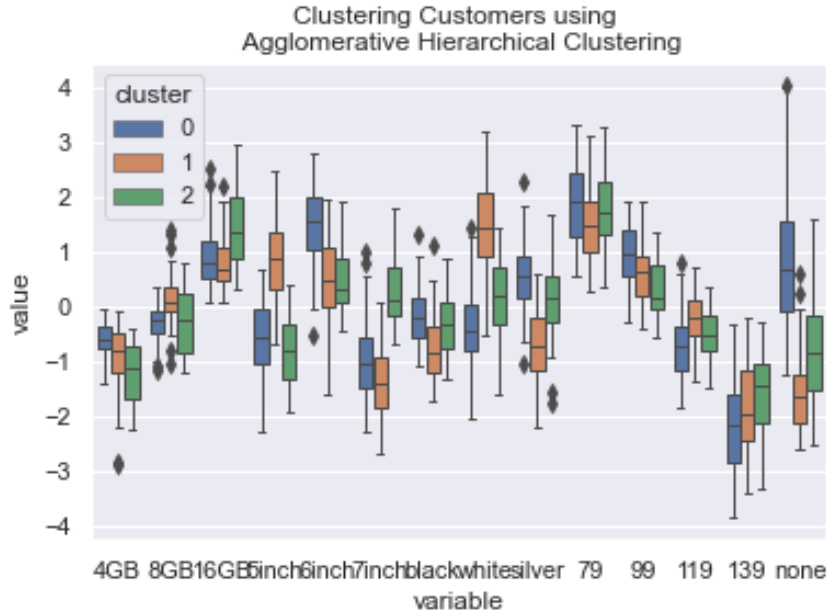


Figure 16: Boxplots of the individual part-worth utilities for the different cluster using Agglomerative Hierarchical Clustering.

Lastly, we comment on the second question. As we mentioned in [Subsection 5.3](#), the package `ChoiceModelR` allow us to introduce some features of the respondents to perform the estimation. Among the features we could find:

- Behavioural features, such as the ones mentioned in [Subsection 3.2](#).
- Demographic features, such as socio-economic status, education, gender, age, monthly income or place of residence.

According to leading companies of the sector such as Sawtooth or Conjointly, the introduction of the former features can enhance the Conjoint Analysis. On the other hand, including demographic features in the estimation procedure, downgrades the results in general. Instead, it is recommended to perform data analysis and customer clustering without using demographic data. For later, see if there are any patterns or common characteristics among the members of each of the clusters. If we find common characteristics for a cluster, we could launch personalized offers for this type of people or discounts to attract their attention.

## 6.2 Market Simulator

The market simulator is usually considered the most important tool resulting from a conjoint analysis project. The simulator converts raw conjoint (individual part-worth utilities) data into something much more managerially useful: simulated market choices. Products can be introduced within a simulated market scenario and the simulator reports the percentage of respondents projected to choose each product. This is like having all of our respondents gathered in one room for the sole purpose of voting on product concepts within competitive scenarios. In this way, we can compare our products with those of our competitors, based on the differences between the attributes of these products. A market simulator serves in general as a tool that helps the marketing team of a company to make decisions, such as new product design, product positioning, and pricing strategy. We refer to the Conjointly company website [\[3\]](#) where we can see how all these applications could be implemented in the Symson Software. All the results shown in this section have been implemented in `Python` and the file that we used is `MarketSimulatorConjoint.py`.

For this market simulator we assume that the Choice model given by (7) holds. Moreover as it is a probabilistic we will simulate 200 times to get more robust estimators, and confidence intervals. Additionally, we have dealt with pricing in a discrete manner in [Section 3](#) and in [Section 5](#). In this section we will use interpolation to treat the price in a continuous way to obtain more realistic results. Below, we show different examples of the conclusions that can be inferred using these market simulators.

### 6.2.1 Price Elasticities of New Products

Let us assume that our company is interested in entering a market that currently consists of 3 competitors + the not-buy option. We have developed a new product and we want to investigate its

potential with respect to the existing products. The first application we can get from this simulator is to compute the price we should put on the product to maximize the total revenue of the company.<sup>10</sup>

For instance:

Table 3: Market Situation for computing price elasticities.

Products	Storage	Screen Size	Color	Price	None (€)
Competitor 1	4GB	5inch	white	100	-
Competitor 2	8GB	7inch	silver	120	-
Competitor 3	16GB	5inch	black	135	-
Competitor 4	-	-	-	-	x
My product	16GB	5inch	white	?	-

The outputs of the market simulator for the Introduction of New Products are:

- Elasticity Curve:

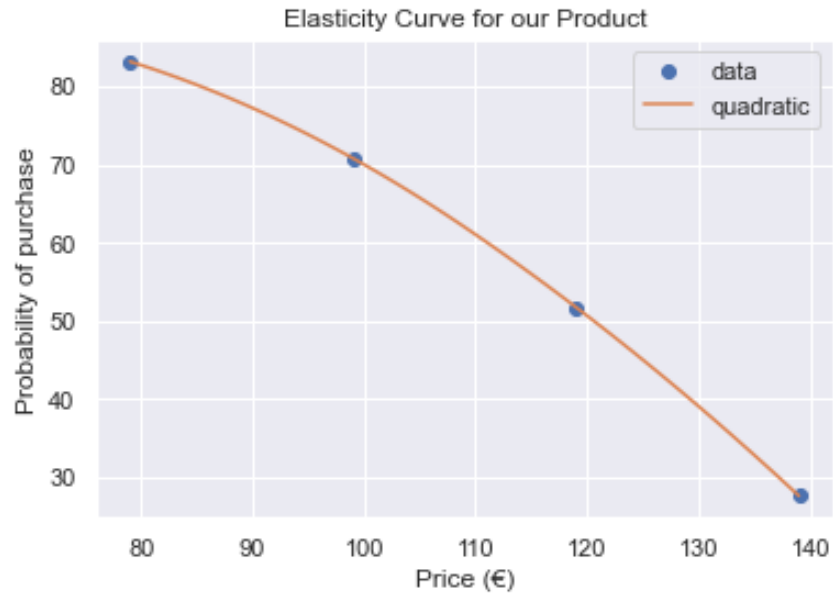


Figure 17: Elasticity Curve for the example of price elasticities.

- Market Share of Preference for each product.

<sup>10</sup>we could also maximize the total margin if we knew the cost behind our product

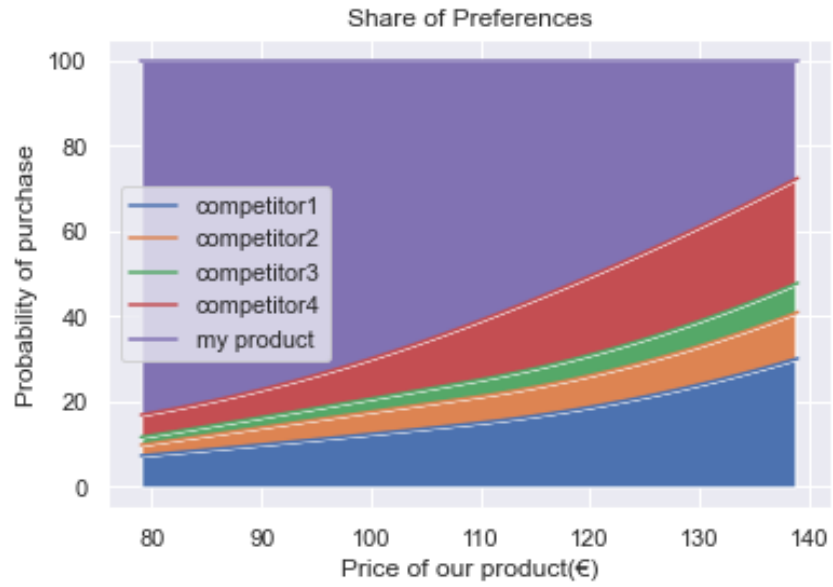


Figure 18: Market Share of Preference for the example of price elasticities.

- Revenue vs Price. Computing the maximum, we obtain the desired optimal price.



Figure 19: Revenue vs Price for the example of price elasticities.

- Histogram of the results for every simulation.



Figure 20: Histogram of optimal prices for the example of price elasticities.

So for this example we can obtain:

The price to maximize revenue is: 96.58

The Maximum Revenue is given by: 7016.1

The 95% confidence interval is: [79.0, 109.3]

In addition, with the market simulator we can see how our product would be affected by the price variance of our competitors, either by discounts or price increases.

### 6.2.2 Willingness To Pay per Feature

Since the inception of conjoint analysis, researchers and their clients have sought intuitive ways to quantify the preference for attribute levels in monetary terms. This is known as: Willingness to Pay per Feature. This subsection then, is mainly devoted to answer the following question:

- How much should we increase the price of a product when we improve one of its attributes?. Or how much we should decrease the price if one of the product's features worsens?.

The common approaches to WTP tend to overstate it, since they do not explicitly consider competition or the ability to opt out (choose the None). They also tend to average across respondents rather than focusing WTP more relevantly on respondents on the cusp of choosing the enhanced product features. To promote better practice and more reasonable results, we show here the procedures outlined in [14] considering competition for estimating WTP via market simulation.

To estimate WTP for features associated with the firm’s base case product, we employ the preference share (indifference) approach. This approach finds the change in price associated with a product enhancement that drives the share of preference for the firm back to its original preference prior to making the enhancement. This is best visualized with an example. For instance:

Table 4: Market Situation for computing WTP.

Products	Storage	Screen Size	Color	Price	None (€)
Competitor 1	4GB	5inch	white	100	-
Competitor 2	8GB	7inch	silver	120	-
Competitor 3	16GB	5inch	black	135	-
Competitor 4	-	-	-	-	x
My old product	4GB	6inch	silver	100	-
My new product	8GB	6inch	silver	?	-

We want to compute how much should we increase the price if we upgrade the Storage of our product from 4GB to 8GB. For the old product we have that the Share of preferences is given by:

	Share of preferences
Competitor1	27.3450
Competitor2	9.5100
Competitor3	8.3875
Competitor4	15.0950
My old product	39.6625

Therefore we need to compute the price such that the share of preference for the new product remains in 0.396625, that would be the same market position. Below we show the obtained results.

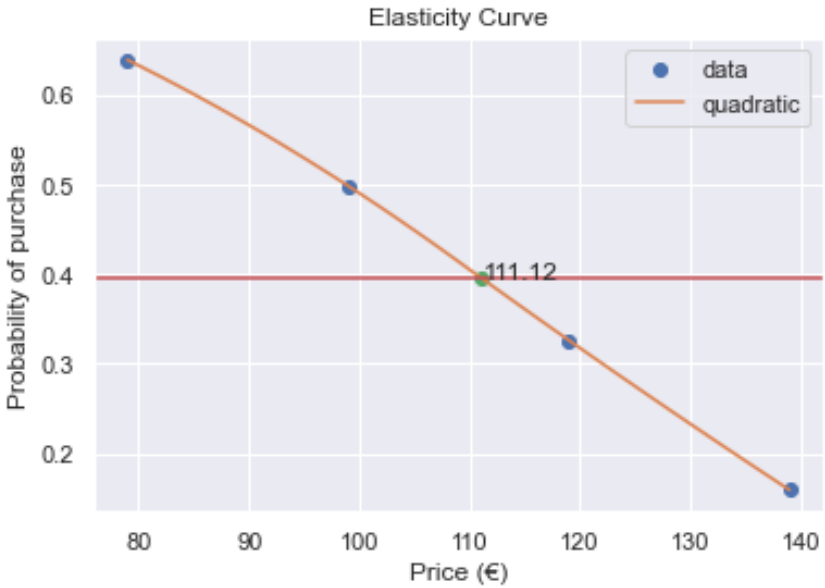


Figure 21: Elasticity curve for the new product





Figure 22: Histogram of the WTP

As we mentioned previously we simulate 200 times to obtain confidence intervals for this estimator. The final results for this example are:

The WTP of this example is: 11.73

The WTP confidence interval is: [3.85, 19.0]

Additionally, WTP is non-additivity in the sense that we can not treat a multilevel change summing up the WTP for each of the individual changes. This is not a problem since we include in our code the option of changing more than one attribute at the same time. We mention that an exhaustive set of competitors doesn't need to be specified; but the important players in the market should be represented. Indeed, it has been shown in [4] that the estimators obtained when using only the most remarkable competitors is quite close to the case where we account for all the competitors.

### 6.2.3 Designing Products for Market Segments

In this subsection we develop the last application for our Conjoint Analysis: Designing Products for Market Segments. This application can be seen as a combination of the Customer Clustering in [Subsection 6.1](#) and the Price Elasticities of new products in [Subsubsection 6.2.1](#).

Customizing products to appeal to target segments or even individuals is a common theme in marketing. Many companies dedicate significant resources to developing a portfolio of products that it hopes will appeal to unique segments. For line extensions, the challenge for any company is to design new products that take share from its competitors without stealing an unacceptable amount of share from products within its existing line. The aim of this section is then to answer the following questions:

- Given the current products that we have in the market, which new products will have the highest impact for our company?.
- In which segments of the population our products have a weak performance and what product can we launch to attract customers belonging to that group?.

To answer these questions we cluster the customers in different segments as we did in [Subsection 6.1](#). The ability to have our market simulator select respondents for segment analysis can further enhance the power of the tool. Below we see an example of this application.

Table 5: Market Situation for designing products for market segments example.

Products	Storage	Screen Size	Color	Price	None (€)
Competitor 1	8GB	5inch	black	106	-
Competitor 2	16GB	7inch	white	110	-
Competitor 3	-	-	-	-	x
My product	4GB	6inch	silver	90	-

The preference share for this market situation for each of the clusters is:

	Cluster0, n = 83	Cluster1, n = 56	Cluster2, n = 61	Total Share
Competitor1	4.819277	30.357143	9.836066	13.25
Competitor2	3.614458	48.214286	54.098361	32.00
Competitor3	16.265060	3.571429	8.196721	10.50
My product	74.698795	16.071429	27.868852	44.50

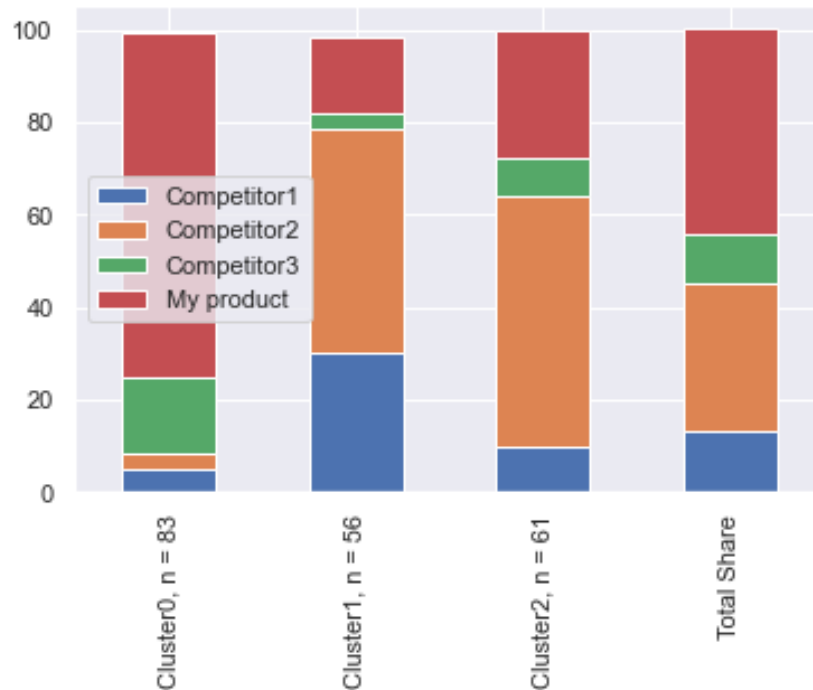


Figure 23: Market Share preferences for the different clusters.

We can see that our product has huge success among cluster 0. In fact, looking back at [Figure 15](#), we observe that the most preferred screen size and color are respectively, 6 inches and silver. Moreover, for this cluster the relative importance given to the Storage is lower than for the other clusters. These 3 facts explain the good performance of this product on cluster 0. On the other hand, respondents from other groups have a preference for other screen sizes and colors, which explain the low market share on these clusters.

After analysing the previous results, we wonder which product might be effective in order to take share from competitors without damaging our own product. We will consider 3 products, each of them trying to attract the attention of each of the clusters. The results that we obtain for each of these products are:

Table 6: Strategic Products for each cluster.

Products	Cluster Aiming	Storage	Screen Size	Color
New product 1	0	4GB	6inch	black
New product 2	1	4GB	5inch	white
New product 3	2	16GB	6inch	silver

- My new product 1:

	Cluster0,n=83	Cluster1,n=56	Cluster2,n=61	Total	Share
Competitor1	2.409639	23.214286	8.196721		10.0
Competitor2	2.409639	41.964286	45.901639		27.0
Competitor3	12.048193	1.785714	6.557377		8.0
Myproduct	50.602410	10.714286	19.672131		30.0
Mynewproduct1	31.325301	21.428571	18.032787		25.0

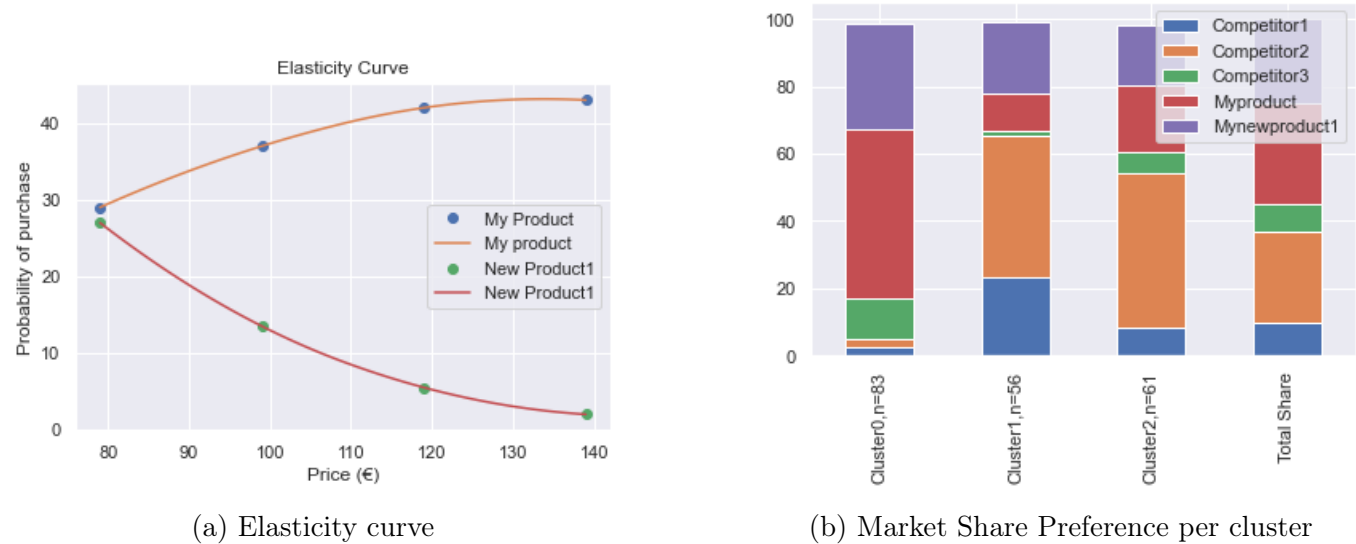
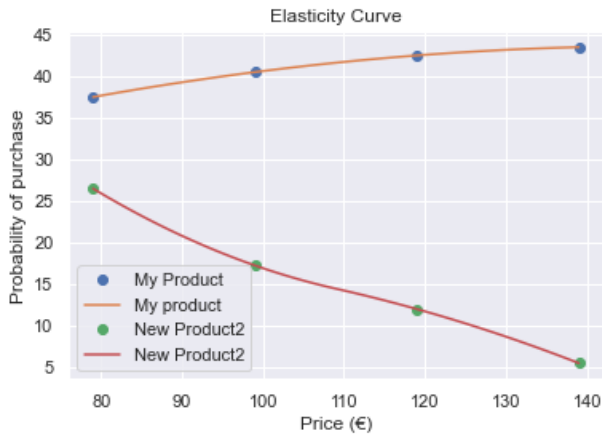


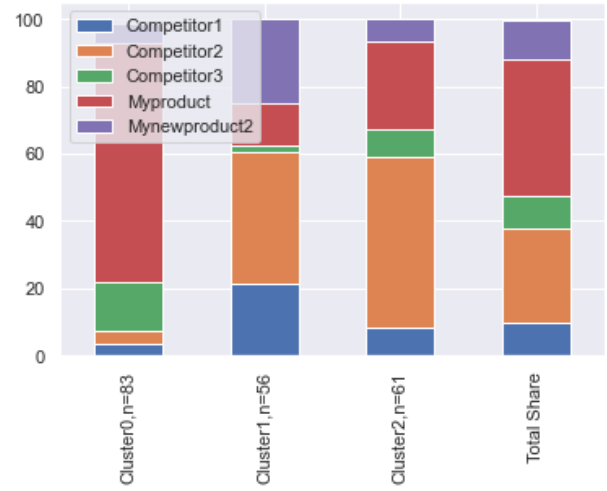
Figure 24: Analysis of my new product 1

- My new product 2:

	Cluster0,n=83	Cluster1,n=56	Cluster2,n=61	Total Share
Competitor1	3.614458	21.428571	8.196721	10.0
Competitor2	3.614458	39.285714	50.819672	28.0
Competitor3	14.457831	1.785714	8.196721	9.5
Myproduct	71.084337	12.500000	26.229508	40.5
Mynewproduct2	6.024096	25.000000	6.557377	11.5



(a) Elasticity curve

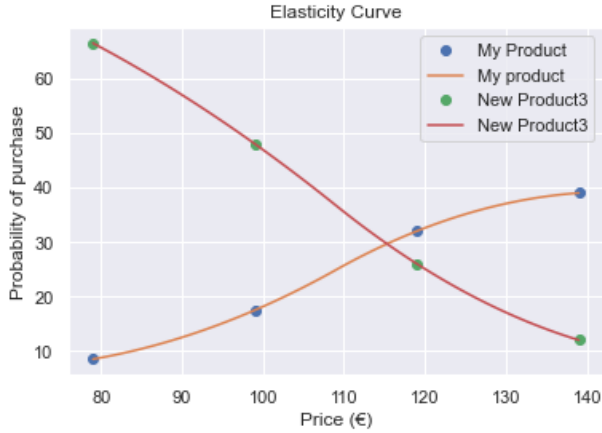


(b) Market Share Preference per cluster

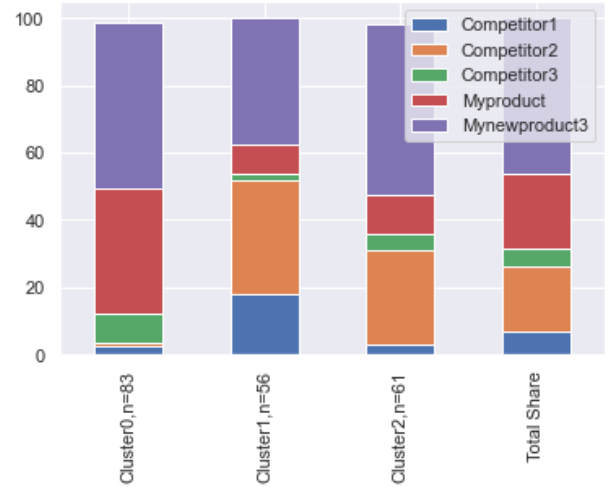
Figure 25: Analysis of my new product 2

- My new product 3:

	Cluster0,n=83	Cluster1,n=56	Cluster2,n=61	Total Share
Competitor1	2.409639	17.857143	3.278689	7.00
Competitor2	1.204819	33.928571	27.868852	19.00
Competitor3	8.433735	1.785714	4.918033	5.75
Myproduct	37.349398	8.928571	11.475410	22.00
Mynewproduct3	49.397590	37.500000	50.819672	46.25



(a) Elasticity Curve



(b) Market Share Preference per cluster

Figure 26: Analysis of my new product 3

The prices to maximize revenue, the total revenue and the total market share are giving by:

Optimal Price Product 1: 85.97€, Total Revenue: 4917.25 €, Total Market Share: 55%

Optimal Price Product 2: 91.42€, Total Revenue: 5671.25 €, Total Market Share: 52%

Optimal Price Product 3: 97.79€, Total Revenue: 6254.8 €, Total Market Share: 64.25%

The conclusion is therefore that we should choose to launch product 3, since it has much more impact and relevance than the other 2.

### 6.3 Remarks

Under very controlled conditions (such as markets with equal information and distribution), market simulators often report results that closely match long-range equilibrium market shares. But conjoint utilities cannot account for many realworld factors that shape market shares, such as length of time on the market, distribution, out-of-stock conditions, advertising, effectiveness of sales force, and awareness [4]. Conjoint analysis predictions also assume that all relevant attributes that influence share have been measured. Therefore, the share of preference predictions usually should not be interpreted as market shares, but as relative indications of preference. Divorcing oneself from the idea that conjoint simulations predict market shares is one of the most important steps to getting value from a conjoint analysis study and the resulting simulator. While external-effect factors can be built into the simulation model to tune conjoint shares of preference to match market shares, we suggest avoiding this temptation if at all possible. No matter how carefully conjoint predictions are calibrated to the market, the researcher may one day be embarrassed by differences that remain.

## 7 Future Research

To conclude this project we show here the future challenges that we have found in the last weeks, but that we have not had time to carry out. This section is thought in case someone in the future would like to implement this Conjoint Analysis, in order to have new ways of improvement.

- Cleaning bad respondents: Among the respondents, it is very likely that there were people who answered randomly, without giving much thought to the answer or giving excluding answers to different questions of the questionnaire. Eliminating these respondents is of vital importance to ensure that the data we have for future analysis is of high quality. If we do not eliminate these respondents, the results obtained in the Conjoint may not be entirely accurate. We referentiate to [15] to show a procedure for data cleaning.
- Improving the Questionnaire Building: In the last week we have come across R `idefix` library in R [16]. We have not tested this library in depth, but at first glance it looks like a powerful library that could help us to improve the way we build the questionnaire. This library enables users to generate optimal designs for discrete choice experiments.
- The Sampling of Scenarios: This a useful generalized approach to compute WTP when there isn't certainty as to the base case product specifications or when there is uncertainty about competitor composition and reactions. It is based on repeatedly sampling among randomly selected competitive positioning as well as random variations in the firm's product for which we are estimating WTP. For each sampled scenario, we estimate WTP in the same way we have described in Subsubsection 6.2.2: through finding the equalization price for the enhanced product that sets its share back to the original share prior to enhancement.
- Product Search Optimization: For the implementation that we built in Subsection 6.2 we had to introduce the products that we were interested in. However, it would seem more efficient to let an automated search algorithm find an optimal product or set of products rather than to proceed manually. The genetic algorithms are quite powerful in that respect. We referentiate to [17] to show how genetic algorithm works and how can we apply them to our case. Moreover, the code for carrying out the previous computations and some guidelines to customise it to our data problem are available in `ProductPortfolioOptimisation.R`.
- Rpy2 package: Much of the implementation has been carried out using R software. The main reason has been the convenience of this software, which has been in some parts of the work much more powerful than Python. The problem arises when Symson uses mainly Python for implementations. The only way I have come up with to solve this is to use the Rpy2 Python library. This library is used to connect Python and R.
- Finally, we recommend reading more papers about the Sawtooth Software 2021 conference<sup>11</sup>, In order to go deeper into the topic of Conjoint Analysis and see what are the current research directions of this branch.

---

<sup>11</sup><https://sawtoothsoftware.com/resources/technical-papers/conferences/sawtooth-software-conference-2021>

## References

- [1] Felix Eggers et al. “Choice-Based Conjoint Analysis”. English. In: *Handbook of Market Research*. Ed. by Christian Homburg, Martin Klarmann, and Arnd Vomberg. Springer, Apr. 2018. ISBN: 978-3-319-57411-0. DOI: [10.1007/978-3-319-05542-8\\_23-1](https://doi.org/10.1007/978-3-319-05542-8_23-1).
- [2] SKIM Group. *Choice Modeling*. URL: <https://skimgroup.com/services/choice-modeling/>.
- [3] Conjoint.ly. *Conjoint Preference Share Simulator*. URL: <https://conjointly.com/guides/conjoint-preference-share-simulator/>.
- [4] Sawtooth Software. “Market Simulators for Conjoint Analysis”. English. In: (2019). URL: <https://sawtoothsoftware.com/resources/technical-papers/introduction-to-market-simulators-for-conjoint-analysis>.
- [5] Inc. Sawtooth Software. “The CBC System for Choice-Based Conjoint Analysis”. English. In: (2017). URL: <https://content.sawtoothsoftware.com/assets/0891a76f-93d3-4838-a38d-8ac0a2cda519>.
- [6] Inc. Sawtooth Software. “ACBC Technical Paper”. English. In: (2014). URL: <https://sawtoothsoftware.com/resources/technical-papers/acbc-technical-paper>.
- [7] Stefan Binner Peter Kurz. “Enhance Conjoint with a Behavioral Framework”. English. In: *Proceeding of the Sawtooth Conference*. 2021. URL: <https://sawtoothsoftware.com/resources/technical-papers/conferences/sawtooth-software-conference-2021>.
- [8] Sawtooth Software Bryan Orme. “Three Ways to Treat Overall Price in Conjoint Analysis”. English. In: (2007). URL: <https://sawtoothsoftware.com/resources/technical-papers/three-ways-to-treat-overall-price-in-conjoint-analysis>.
- [9] Yves Croissant. “Estimation of Random Utility Models in R: The mlogit Package”. In: *Journal of Statistical Software* 95.11 (2020), pp. 1–41. DOI: [10.18637/jss.v095.i11](https://doi.org/10.18637/jss.v095.i11). URL: <https://www.jstatsoft.org/index.php/jss/article/view/v095i11>.
- [10] Mauricio Sarrias and Ricardo Daziano. “Multinomial Logit Models with Continuous and Discrete Individual Heterogeneity in R: The gnm1 Package”. In: *Journal of Statistical Software* 79.2 (2017), pp. 1–46. DOI: [10.18637/jss.v079.i02](https://doi.org/10.18637/jss.v079.i02). URL: <https://www.jstatsoft.org/index.php/jss/article/view/v079i02>.
- [11] Inc. Sawtooth Software. “The ACA/Hierarchical Bayes v3.0 Technical Paper”. English. In: (2006).
- [12] Ph.D. Ryan Sermas assisted by John V. Colias. “Package ‘ChoiceModelR’”. English. In: (2012). URL: <https://cran.r-project.org/web/packages/ChoiceModelR/ChoiceModelR.pdf>.
- [13] Inc. Keith Chrsan Sawtooth Software. “How Many Holdout Tasks for Model Validation?” English. In: (2015). URL: <https://sawtoothsoftware.com/resources/technical-papers/how-many-holdout-tasks-for-model-validation>.
- [14] Bryan Orme. “Estimating Willingness to pay (WTP) given Competition in Conjoint Analysis”. English. In: *Proceeding of the Sawtooth Conference*. 2021. URL: <https://sawtoothsoftware.com/resources/technical-papers/conferences/sawtooth-software-conference-2021>.

- [15] Keith Chrzan and Inc. Cameron Halversen Sawtooth Software. “Diagnostics for Random Respondents in Choice Experiments”. English. In: (2020). URL: <https://sawtoothsoftware.com/resources/technical-papers/diagnostics-for-random-respondents-in-choice-experiments>.
- [16] Frits Traets, Daniel Gil Sanchez, and Martina Vandebroek. “Generating Optimal Designs for Discrete Choice Experiments in R: The idefix Package”. In: *Journal of Statistical Software* 96.3 (2020), pp. 1–41. DOI: [10.18637/jss.v096.i03](https://doi.org/10.18637/jss.v096.i03). URL: <https://www.jstatsoft.org/index.php/jss/article/view/v096i03>.
- [17] Christopher Chapman and James Alford. “Product portfolio evaluation using choice modeling and genetic algorithms”. In: Jan. 2010.