Use R!

Antonio Páez
Geneviève Boisjoly

Discrete Choice Analysis with R

Springer

Use R!

This series of inexpensive and focused books on R is aimed at practitioners. Books can discuss the use of R in a particular subject area (e.g., epidemiology, econometrics, psychometrics) or as it relates to statistical topics (e.g., missing data, longitudinal data). In most cases, books combine LaTeX and R so that the code for figures and tables can be put on a website. Authors should assume a background as supplied by Dalgaard's Introductory Statistics with R or other introductory books so that each book does not repeat basic material.

How to Submit Your Proposal

Book proposals and manuscripts should be submitted to one of the publishing editors in your region per email – for the list of statistics editors by their location please see https://www.springer.com/gp/statistics/contact-us. All submissions should include a completed Book Proposal Form.

For general and technical questions regarding the series and the submission process please contact Faith Su (faith.su@springer.com) or Veronika Rosteck (veronika.rosteck@springer.com).

Antonio Páez · Geneviève Boisjoly

Discrete Choice Analysis with R

Springer

Antonio Páez
McMaster University
Hamilton, ON, Canada

Geneviève Boisjoly
Polytechnique Montréal
Montréal, QC, Canada

ISSN 2197-5736 ISSN 2197-5744 (electronic)
Use R!
ISBN 978-3-031-20718-1 ISBN 978-3-031-20719-8 (eBook)
https://doi.org/10.1007/978-3-031-20719-8

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

"Would you tell me, please, which way I ought to go from here?"

"That depends a good deal on where you want to get to," said the Cat.

"I don't much care where—" said Alice.

"Then it doesn't matter which way you go," said the Cat.

"—so long as I get SOMEWHERE," Alice added as an explanation.

"Oh, you're sure to do that," said the Cat, *"if you only walk long enough."*

—Lewis Carroll, Alice in Wonderland

"We are our choices."

—Jean-Paul Sartre

Choices, Choices, Choices

According to Sartre, we are our choices, which implies that we have at least a certain ability to choose what we are. The choices we make have important implications for how we interact with each other and the world. It is a truism to say that we live in a time when resources grow increasingly scarce—when, as a species, we are testing the limits to growth. In this context, the way we make choices is informative about our individual preferences and where we, collectively, are going. For this reason it is important to understand choice-making and individual preferences. This includes behavior around routine activities, such as deciding how to travel for everyday purposes (for instance walking or cycling, driving, or using transit). As well, longer term decisions are of interest, such as the frequency with which we travel by plane; the choice to live in a distant suburb where rent is low but transportation is expensive, or in a city's center where accessibility is high but space is at a premium; whether to contract more expensive but environmentally less damaging low impact energy sources; whether to buy hybrid, electric, or gasoline vehicles. These are just some examples of the myriads of choices that we all make in the course of a lifetime, and that collectively impact the economy, our natural environment, and social relationships.

Understanding decision-making is also important to ensure that the world, or more accurately the social institutions that collectively represent us in our interactions with the world, can better accommodate and possibly even nudge our preferences towards socially desirable outcomes. What do our choices tell us about our long term sustainability as an advanced technological species? What trade-offs are members of the public willing to contemplate when choosing alternative mobility? Is it the range of a new electric vehicle? Or its price? The time it takes to charge its battery? The satisfaction of being green? Do consumers prefer more range in electric vehicles, and how much are they willing to pay for it? Does the trade-off justify the increase in production costs? Should governments subsidize purchases of efficient heating systems? If so, what is the effect of a certain amount in subsidy?

These questions are important for governments and businesses as they try to understand public responses to fiscal, engineering, or production decisions.

In very simple terms, discrete choice analysis is a family of modeling techniques that are useful to understand behavior when decisions concern alternatives that are indivisible and mutually exclusive. These techniques are particularly relevant when the alternatives are bundles of attributes that collectively have an explicit *price* but whose individual features may only have *implicit* prices. In this way, discrete choice analysis represents a form of *hedonic price analysis*, a tool to understand the preferences of individuals in the presence of implicit markets (Wong 2018).

But what are prices? Broadly speaking, a price is a quantity offered in payment for one unit of a good or service. Economists speculate that early economies were based on bartering. This assumes relatively simple economic systems, and yet ones where people already have stopped being generalists to become specialists of some sort, and therefore sometimes desire to exchange goods and/or services with others. Someone who specializes in making bread probably has only limited time to grow wheat, or to mine salt, or to make shoes. The same thing goes for everyone who has specialized at something, and cannot meet all their needs using only their own labor and resources.

Imagine for a moment that you have specialized at baking. Some of your neighbors need bread and may be willing to offer a quantity of something in exchange for some of it. The person who breeds chicken may be willing to offer one chicken for three loafs of freshly baked bread. Is that a fair exchange? How can anyone determine whether that exchange is sensible? Well, if both parties agree that one chicken is a satisfactory exchange for three loafs of bread, we can maybe say that the price of one loaf of bread is one third of a chicken.

Imagine now that you, the baker, need someone to mend your roof, and therefore require the services of yet another member of your community who has specialized in going up ladders and thatching roofs. How many loafs of bread should you offer this person for making sure that you do not have leaks when it rains? And what if the thatcher has allergies and cannot eat bread? But maybe this person would agree to fix your roof if he received chickens in exchange.

As you can imagine, this simple system can rapidly become unwieldy [and may actually never have existed in reality in this form; see Graeber (2011)]. Besides the need to coordinate multiple actors in what potentially is multiple two-way transactions (baker-to-thatcher becomes baker-to-farmer-to-thatcher), the situation is further complicated by the reliance of bartering on some kind of trust system: as a person offering a good or a service in a bartering system, there are no simple ways of ensuring the quality of the exchange! For example, what if the thatcher is a crook, or the farmer gives you diseased chickens in exchange for your top-notch, high-quality, mouth-watering loafs of bread? In small systems, where agents can recognize each other, trust is enforced by reputation—if the thatcher is crooked, or the farmer is known to feed lead to the poultry, other actors can avoid transactions with them. If leaks begin as soon as it starts to rain and this becomes known, people will soon avoid doing business with the thatcher.

As these simple examples illustrate, even simple bartering systems are complicated ways of setting prices, something that becomes increasingly difficult (except in very exceptional situations) when an economy produces hundreds, thousands, or even more different products and services.

(An interesting exception is the time when Pepsico bartered with the Soviet Union; see this news item from 1990.[1] In this case, soft drinks were bartered for ships and vodka [Why did barter make sense in this situation? Hint: the Soviet ruble's official exchange rate compared to other currencies was more or less meaningless; Wyczalkowski (1950)]).

Something else needs to emerge to facilitate complicated exchanges in a complex economy, that something being money.

Price Mechanisms, or, Is Money the Root of All Evil

A sentiment commonly expressed in numerous cultures throughout history is a disapproval of greed. A well-known example is 1 Timothy 6:10, which warns that the love of money is the root of all of evil. And while love of money for its own sake may not strike us as virtuous, it is almost certain that no complex economy can exist without the invention of money. The complexities of bartering make clear the need for a common standard for exchange. Instead of needing to figure out how many shoes a chicken is worth and how many chickens are equivalent to a new roof, everything is

[1] https://www.nytimes.com/1990/04/09/business/international-report-pepsi-will-be-bartered-for-ships-vodka-deal-with-soviets.html.

measured using the same metric: shells, deer leather,[2] rupees, pesos, bilimbiques,[3] or dollars.

The limitations of bartering help to explain (if in a somewhat simplistic fashion) the necessity of monetary systems. A common currency frees the maker of shoes from the need to calculate the cost of his new roof in chickens. But it does not explain how prices are set in the common currency.

Price mechanisms depend to a large extent on the institutional framework. Several such frameworks exist.

Think for example of a centrally planned economy, where prices for goods or services are set by a designated agent. This could be the Elder of the Village. The Elder of the village decides how many pesos people should pay for shoes (alternatively, how much the shoe maker can charge for a pair of shoes), and how much you should pay a thatcher for each hour of work (or for a completed roof). Everyone in this kind of setup prays that the Elder knows what they are doing, and the potential for mistakes clearly is far from negligible. In the Soviet Union prices were set using so-called *material balances*, which aimed at balancing the inputs to the planned outputs. As history shows, this approach was not successful. There are several reasons for this, including ideological limitations in the mathematical tools used to calculate balances, as well as the limitations of such planning, which is inherently rigid and does not allow for deviations from the plan (what if you end up needing *two* pairs of shoes instead of only one or none?).

In a free market economy, on the other hand, prices are left to float with no intervention from central planning agencies. There is a voluminous literature explaining how this system can allocate resources efficiently. Although much of this literature is riddled with fantasies that wish away externalities and other market failures, it does contain numerous valuable insights, including the notion that prices are signals of how desirable a good or service is, and the incentives to produce more or less of it.

Economists explain this fundamental relationship using the concepts of demand and supply.

The basic assumption in this relationship (which happens to hold in many cases) is that the level of demand for a good or service (the quantity that consumers are willing to purchase) declines as the price increases. On the side of producers and providers of goods and services, the level of supply (the quantity that they are willing to produce) increases with price. Demand and supply, then, are influenced by price, but they do not happen in isolation—rather, they interact to set prices. A consumer cannot single-handedly demand that a good/service be sold at a certain (i.e., low) price when many consumers are willing to pay a somewhat higher price for the same (otherwise one could buy a brand new GameStation 2000 for 25 dollars or whatever it costs to produce it). Likewise, a producer of a good or provider of a service cannot expect to set a high price for a good/service when other producers are willing to sell at a lower price. There are, of course, deviations to this kind of rule, which can be fairly common or aberrant depending on the institutional framework. On the one

[2] https://commons.wikimedia.org/wiki/File:Tang_Dynasty_30_Kuan_banknote.jpg.

[3] https://es.wikipedia.org/wiki/Moneda_revolucionaria.

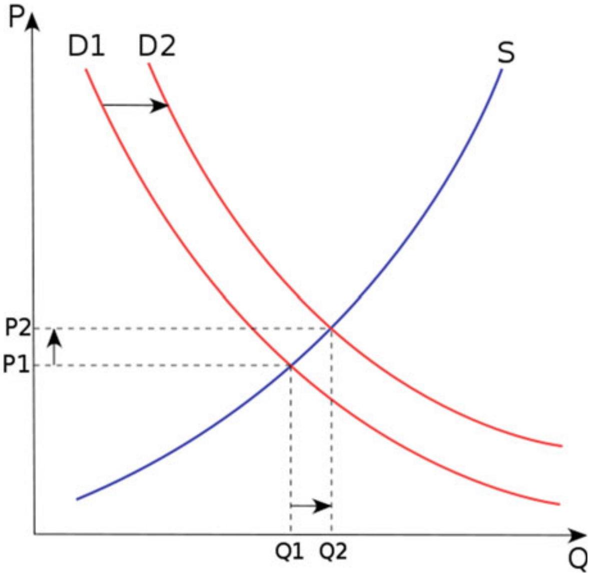


Fig. 1 Supply and demand relationships

hand, monopolies and cartels can manipulate prices on behalf of producers, whereas single-payer health care manipulates prices in favor of patients.

The intersection of the supply and demand curves determines simultaneously prices *and* the quantity of a good/service produced/consumed. Since prices “float”, they can adjust to changes in supply and/or demand. In Fig. 1, for example, when demand for a good or service (say roofs) increases, there is an incentive for thatchers to work more hours. Since there is a limited number of hours that thatchers can work (there is scarcity), this is reflected in a higher price, as long as those who can afford it are willing to pay more for a scarce but desirable service.

Between centrally planned economies and pure free markets exists an endless range of mixed economies, where prices are often left to float but within limits or with other corrective mechanisms, such as subsidies or taxes.

For most of us, the most familiar situation of price mechanisms would be in the form of free or mixed economies, after most centrally planned economies at the end of the 20th century collapsed or changed towards more market-based economies (or in some cases towards systems of crony capitalism). An underlying assumption in these systems is that a market exists for the good or service in question, that is, a medium where goods or services can be exchanged. Markets exist for many things: for milk, for bread, for insurance, and for complex financial instruments that no one really understands, such as derivatives. On the other hand, markets do not explicitly exist for particular features of composite products or services, and therefore the willingness to pay of consumers for *elements* of a specific good cannot be discerned directly. (In reality, even something as seemingly simple as milk can be seen as a bundle: the brand, the fat content, addition of vitamins, etc.)

Imagine, for instance, a durable good such as an automobile. Automobiles are composite goods in the sense that, even if their purpose is to enhance the natural capabilities of humans for movement, they can do this in many different ways to satisfy a diversity of needs or tastes: with variations in leg room, differences in acceleration, and various levels of fuel consumption, to name a few. Pricing mechanisms for the whole good (the vehicle) leave the question of willingness to pay for specific components in the dark. How much are consumers willing to pay for those extra centimeters of leg room? Or for more spacious seats? Or greater horsepower? The markets for each of these items are implicit in the market price of automobiles.

In fact, hedonic price analysis was invented to address precisely such questions about implicit prices by Andrew Court, an economist for the Automobile Manufacturers' Association in Detroit from 1930 to 1940 (see Goodman 1998). As part of his work for the Association, Court realized that price indexing procedures used at the time were not satisfactory for describing the relative importance of various components of automobiles in determining their price. This in turn was an important step towards a more refined understanding of consumer preferences, which was seen as a way to help producers in their efforts to differentiate products.

Court used the term "hedonic" (related to pleasure) to express the usefulness and desirability that consumers attach to different aspects of a composite product. Although Court invented this method in the late 1930s, it lay dormant and was not widely used for approximately twenty years until it was popularized by Zvi Griliches in the 1960s, with work on fertilizers and automobiles (Griliches 1991). Later on, Sherwin Rosen explained implicit markets within an economic foundation of supply and demand in equilibrium—that is, not just as willingness to pay on the side of consumers, but also as the result of decisions by producers (Rosen 1974). In brief, Rosen explained how the differentiated hedonic price function represents the envelope of a family of "value" functions (willingness to pay) and a family of offer functions (willingness to sell).

Since then, hedonic prices analysis has been found to be useful in numerous applications, ranging from the pricing of computers (Berndt and Rappaport 2001), to personal digital assistants (Chwelos, Berndt, and Cockburn 2008), wine (Unwin 1999), online purchases (Clemons, Hann, and Hitt 2002), and farmland (Isgin and Forster 2006).

The field of discrete choice analysis, as a form of hedonic price analysis, is concerned with the study of implicit markets when the outcome of the choice process is discrete. This requires a few things:

1. A decision-maker who can choose between at least two, and possibly more alternatives. If only one alternative is available, the decision-maker does not face a choice situation.
2. The decision-maker is assumed to act in a consistent way that economists call *rational*. Basically, the decision-maker is assumed to always try to maximize the *utility* (the hedonic value) that they get from making a choice. The framework allows for seemingly inconsistent behavior by decomposing the utility into two parts, one of which is random, hence the commonly used term *random utility*

modeling. (Forms of behavior other than utility maximization have been proposed as well, but they all have in common an assumption that the decision-maker acts in a consistent, i.e., “rational” fashion).

3. Importantly, the available alternatives must be mutually exclusive and individually indivisible. A decision-maker cannot choose only *part* of an alternative: it is an all-or-nothing proposition, hence the choice is *discrete.*

Contemporary discrete choice analysis is a sub-discipline of econometrics, statistics, and data analysis. It has evolved from its origins in psychology and economics into a set of highly refined and sophisticated tools to infer the preferences (i.e., willingness to pay) for the implicit attributes of discrete goods or services that are themselves bundles of attributes.

Beyond Prices

The roots of discrete choice analysis can be found in a micro-economic theory of behavior. Applications of discrete choice analysis have proliferated with time, in fields where the emphasis on implicit prices gives way to a more general interest about *preferences.* For example, consider active travel (Eldeeb, Mohamed, and Páez 2021): active travel does not typically involve a monetary component, since walking and cycling are basically free activities (as long as one owns shoes or a bicycle). But we know that travellers respond to changes in their built environment. What is the trade-off between minutes of travel time and sidewalk density? How do changes in facilities for active travel influence the willingness to use active modes of transportation? As another example, blood donation is a voluntary endeavor in many countries (including Canada, see Cimaroli et al. 2012). But does ease of reaching a blood donation clinic influence the frequency of donation by donors? These questions do not involve literal monetary prices, and yet are important to understand certain healthy and/or pro-social behaviors. Discrete choice analysis is sufficiently general to accommodate situations like those described, since it deals with trade-offs and preferences in a fairly general fashion. So, while implicit prices are important, the trade-offs between attributes are a more fundamental way to think about discrete choices.

Plan

This book is meant to serve as a gentle introduction to the fascinating field of discrete choice analysis, and its practical implementation using the `R` computer language for statistics and data analysis, according to the following plan.

The plan with this book is to introduce discrete choice analysis in what we hope is an intuitive way. To achieve this, we use examples and coding, lots and lots of coding. There are numerous books that can be and are used to teach and learn discrete choice analysis. There are some classical references, for instance Ben-Akiva and Lerman (1985) and Train (2009), and then more specialized books such as Louviere et al. (2000). Other books cover discrete choice analysis as one component of modeling systems [such as transportation; see Ortúzar and Willumsen (2011)], or cover related topics but from a statistical perspective (Maddala 1983). The present book should be appealing to students or others who are approaching this topic perhaps for the first time, and we strongly encourage readers to become acquainted with the classical references mentioned above in due course.

Each author organizes topics in a way that is logical to them. Some texts begin with a coverage of fundamental mathematics, probability, and statistics. Others with an introduction to a substantive topic (e.g., the context of travel demand analysis). In the book *Applied Choice Analysis: A Primer* by Hensher et al. (2005), the title of Chap. 10 is "Getting Started Modeling". Train (2009), in contrast, begins by discussing the properties of discrete choice models and discussing the logit model almost right away.

In our teaching practice, we have in the past heavily relied for presentation on Train's book to organize graduate seminars. We find this style of presentation sufficiently intuitive, when combined with some relevant topics introduced at key points. For example, we find that it makes sense to have a discussion of specification and estimation of models after introducing the logit model. In this way, the details of specifying utility functions can be presented in the context of an operational model. Readers will notice that the organization of the book tends to follow Train closely, using a thematic approach, starting from the fundamentals (both technological and technical), before introducing the logit model, and then by families of models, i.e., GEV, probit, and so on. Any of these models can be, and has been extended in the literature into more specialized modelling frameworks. We begin with relatively simple versions of the models, and add complexity in later chapters. In this way, the reader will find that we do not discuss ways to parameterize the scale of distributions until Chap. 11—despite the fact that this modification is also available for the logit model introduced in Chap. 4. This presentation strategy is deliberate, since we believe it helps to avoid overwhelming readers early on as their comprehension of the methods discussed begins to develop.

We also consider that the methods are easier to learn when the reader can develop a good intuitive feeling about them, something better achieved with concrete examples using realistic data. Beginning early on in the text, readers are asked to get their hands dirty with code and data. Again, this is very much a deliberate decision. Most books on discrete choice analysis are software-agnostic, meaning that they cover the topics without making reference to a particular statistical package for analysis. Others rely for presentation on a specific software. For instance, Hensher et al. (2005) refer extensively to the software `NLOGIT`, a software package sold by Econometric Software, Inc.[4] Yet other packages were originally developed independently of a statistical computing project. One example is Michel Bierlaire's BIOGEME.[5] Not

[4] http://www.limdep.com/.

[5] http://biogeme.epfl.ch/.

being associated with a statistical computing project means that synergies with other packages cannot be realized. Newer versions of BIOGEME now exist written almost exclusively in Python which allow the package to benefit from the Python Data Analysis Library Pandas,[6] and more broadly the Python computation ecosystem.

For this text, we have chosen the R statistical computing project. R is a generalist statistical language with a broad user base. We personally find R more accessible than Python, for example, as an introduction to statistical and data analysis computing, particularly with the support of a good Interactive Development Environment such as RStudio.[7]

Packages (the fundamental units of shareable code in R) benefit from the synergies of many developers and users sharing their code in a transparent and open way. Ten years ago it would have been very difficult to write a book on discrete choice modeling based on R: the earliest version of Croissant's {mlogit} package (Croissant 2018) dates from 2009; the earliest version of Sarrias and Daziano's {gmnl} package (Sarrias and Daziano 2017) dates from 2015. Package {ordinal} was first released in 2010. Now these, and other relevant packages, are mature, well-tested units of code to support discrete choice analysis in R.

One important aspect as well is that R and all related packages are free. It is our conviction that research can be accelerated by the generous contributions of developers who graciously share their code with the world. By doing this, they help to maintain the cost of research low, and thus enable more people around the world to engage in it. That said, there is a potential disadvantage: unlike more established (especially commercial) packages that have been kicking around for years if not decades, newer R packages may still have some limitations. To mention one, earlier versions of the packages {mlogit} and {gmnl} were implemented for universal choice sets, that is, under the assumption that all alternatives are available to all decision-makers. There are situations, where this is not realistic; for example, suppose a decision-maker who does not have a bus stop within walking distance of the origin of their trip. It is not reasonable to include the mode "bus" as part of their choice set. That said, the advantages in our opinion far outweigh the disadvantages, especially for an introductory course that can serve as a launchpad before approaching more sophisticated and powerful implementation of discrete choice analysis in R, such as Apollo.[8]

Our plan for this text is to cover a topic in each section that builds on previous material. We have used the materials presented in this book (in various previous incarnations) for teaching discrete choice analysis in different settings. The original notes were developed for use in the course **GEOG 738** *Discrete Choice and Policy Analysis* that Antonio Páez teaches at McMaster University. This course is a full (Canadian) graduate term, which typically means 11 or 12 weeks of classes. The course is organized as a 2-h seminar that is offered once per week. Accordingly, each section is designed to cover very approximately the material required for a 2 h

[6] https://pandas.pydata.org/.

[7] https://www.rstudio.com/.

[8] http://www.apollochoicemodelling.com/manual.html.

seminar. As our collaboration evolved, Geneviève Boisjoly adopted, improved, and contributed to the original notes for use in the course **CIV 6719** *Transport demande modelling*, which she teaches at Polytechnique Montreal. This is a graduate course offered within the transport specialization of the civil engineering program. The course comprises 13 sessions of 3 h, one per week, which leaves plenty of room for in-class exercises, discussions and classwork.

Audience

The original notes were designed to support teaching a graduate course, but are not necessarily limited to graduate students, and could indeed be a valuable resource to senior undergraduate students, instructors teaching discrete choice analysis, practitioners, experienced discrete choice modelers who wish to learn more about the `R` ecosystem, or applied researchers. Discrete choice analysis has applications in economics, geography, travel behavior, residential choices, urban planning, transportation, and public health, to name just a few relevant fields, and this book should be of interest to people conducting empirical research and policy analysis in these fields. The prerequisites for using this book are an introductory college/university level course on multivariate statistics, ideally covering the fundamentals of probability theory and hypothesis testing. Working knowledge of multivariate linear regression analysis is a bonus but not strictly required.

We do not assume previous knowledge of `R`, and instead take what we hope is a gentle approach to introducing it in an intuitive way. The philosophy of the book is to start doing data analysis early and use many practical examples to explain the key concepts of discrete choice analysis. The availability of free and open software, in our case `R`, means that the book can take a more practical approach to teach and learn discrete choice analysis than some of the older classics that predate many software packages (e.g., Ben Akiva and Lerman's classic Discrete Choice Analysis: Theory and Application to Travel Demand). By embedding the family of discrete choice analysis techniques into the larger `R` ecosystem, the skills developed by readers can be easily transferred and augmented with advanced data management, processing, and visualization techniques, in addition to having access to a multitude of data sets for practice purposes. The book begins with basic data analysis skills (with an emphasis on how to conduct meaningful and easy-to-communicate descriptive analysis that includes both continuous and discrete variables) and continues to discrete choice statistical modeling (from simple binomial logit model to more complex mixture models). It also includes a wealth of material on the interpretation and presentation of results, including through predicted probabilities and scenario analysis.

Requisites

This book is not a course to learn R. The language is introduced progressively and assumes that readers are computer-literate and have possibly done some basic coding in the past. For readers who wish to learn basic and advanced R, there are other valuable resources such as Wickham and Grolemund[9] (2016) or Albert and Rizzo (2012).

To fully benefit from this text, up-to-date copies of R[10] and RStudio[11] are highly recommended. There are different packages that implement discrete choice methods in R. We will particularly rely on the packages {mlogit},[12] {gmnl},[13] and {ordinal}.[14] Skills developed by working with this book serve as a launching pad to later jump to more sophisticated, if more demanding, packages such as Hess and Palma's Apollo.[15]

Getting Started

This book has a companion package called {discrtr}.[16] The package includes data sets and other supporting materials, some of which are required in the examples. The package can be installed as follows:

```
if (!require("remotes"))
  install.packages("remotes",
                   repos = "https://cran.rstudio.org")
if (!require("discrtr"))
  remotes::install_github("paezha/discrtr")
```

[9] https://r4ds.had.co.nz/.

[10] https://cran.r-project.org/.

[11] https://www.rstudio.com/.

[12] https://CRAN.R-project.org/package=mlogit.

[13] https://CRAN.R-project.org/package=gmnl.

[14] https://CRAN.R-project.org/package=ordinal.

[15] http://www.apollochoicemodelling.com/manual.html.

[16] https://paezha.github.io/discrtr/.

Session Information

This book was written using R Version 4.2.1. This is the session information:

```r
sessionInfo()
```

```
R version 4.2.1 (2022-06-23 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19043)

Matrix products: default

locale:

[1] LC_COLLATE=English_United States.utf8
[2] LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

attached base packages:

 [1]  stats  graphics  grDevices  utils  methods  base
                       datasets


other attached packages:
```

```
[1]   webshot2_0.1.0       treemapify_2.5.5   tidyr_1.2.0         tibble_3.1.8
[5]   stargazer_5.2.3      sf_1.0-8           plyr_1.8.7          plotly_4.10.0
[9]   ordinal_2019.12-10   readr_2.1.2        mvord_1.1.1         dfoptim_2020.10-1
[13]  ucminf_1.1-4         BB_2019.10-1       minqa_1.2.4         mlogit_1.1-1
[17]  dfidx_0.0-5          kableExtra_1.3.4   htmlwidgets_1.5.4   gridExtra_2.3
[21]  gplots_3.1.3         gmnl_1.1-3.3       Formula_1.2-4       maxLik_1.5-2
[25]  miscTools_0.6-26     ggspatial_1.1.6    ggridges_0.5.3      ggmosaic_0.3.3
[29]  ggalluvial_0.12.3    ggplot2_3.3.6      evd_2.3-6.1         dplyr_1.0.9
[33]  discrtr_0.0.0.9700   cowplot_1.1.1      biscale_1.0.0       AER_1.2-10
[37]  survival_3.3-1       sandwich_3.0-2     lmtest_0.9-40       zoo_1.8-10
[41]  car_3.1-0            carData_3.0-5
```

```
loaded via a namespace (and not attached):
 [1] websocket_1.4.1     colorspace_2.0-3    ellipsis_0.3.2
 [4] class_7.3-20        rstudioapi_0.13     proxy_0.4-27
 [7] ggfittext_0.9.1     ggrepel_0.9.1       chromote_0.1.0
[10] fansi_1.0.3         mvtnorm_1.1-3       xml2_1.3.3
[13] codetools_0.2-18    splines_4.2.1       mnormt_2.1.0
[16] knitr_1.39          jsonlite_1.8.0      compiler_4.2.1
[19] httr_1.4.3          assertthat_0.2.1    Matrix_1.4-1
[22] fastmap_1.1.0       lazyeval_0.2.2      cli_3.3.0
[25] later_1.3.0         htmltools_0.5.2     tools_4.2.1
[28] gtable_0.3.0        glue_1.6.2          Rcpp_1.0.9
[31] msm_1.6.9           vctrs_0.4.1         svglite_2.1.0
[34] xfun_0.31           stringr_1.4.0       ps_1.7.1
[37] rbibutils_2.2.8     rvest_1.0.2         lifecycle_1.0.1
[40] renv_0.15.5         gtools_3.9.3        statmod_1.4.36
[43] MASS_7.3-57         scales_1.2.0        promises_1.2.0.1
[46] hms_1.1.1           expm_0.999-6        stringi_1.7.8
[49] plotrix_3.8-2       e1071_1.7-11        caTools_1.18.2
[52] optimx_2022-4.30    truncnorm_1.0-8     Rdpack_2.4
[55] rlang_1.0.4         pkgconfig_2.0.3     systemfonts_1.0.4
[58] bitops_1.0-7        evaluate_0.15       lattice_0.20-45
[61] purrr_0.3.4         processx_3.7.0      tidyselect_1.1.2
[64] magrittr_2.0.3      R6_2.5.1            generics_0.1.3
[67] DBI_1.1.3           pillar_1.8.0        withr_2.5.0
[70] units_0.8-0         abind_1.4-5         KernSmooth_2.23-20
[73] utf8_1.2.2          tzdb_0.3.0          rmarkdown_2.14
[76] grid_4.2.1          data.table_1.14.2   pbivnorm_0.6.0
[79] digest_0.6.29       classInt_0.4-7      webshot_0.5.3
[82] numDeriv_2016.8-1.1 munsell_0.5.0       viridisLite_0.4.0
[85] quadprog_1.5-8
```

Check the following repository for instructions about using package {renv}[17] to create a replicable environment to work with the code in this book:

https://github.com/paezha/Discrete-Choice-Analysis-with-R.

Hamilton, Canada — Antonio Páez
Montréal, Canada — Geneviève Boisjoly

[17] https://rstudio.github.io/renv/index.html.

Contents

Chapter 1
Data, Models, and Software

"You can have data without information, but you cannot have information without data."

—Daniel Keys Moran

"Essentially, all models are wrong, but some are useful."

—George E.P. Box

"All models are wrong but some I am emotionally attached to."

— Isabella Bicalho-Frazeto

1.1 What Are Models?

Models propose a simplified representation of the reality, which is useful to develop a common ground for describing, analyzing, and understanding complex phenomena.

Model building requires three things:

1. Raw materials.
2. Tools.
3. Technical expertise (hopefully!).

This is true whether the model is physical (for instance a sculpture), conceptual (a mental map), or statistical/mathematical (the gravity model or a regression model).

In the case of a sculpture, the raw materials can be marble, wood, or clay; the tools chisels, mallet, and spatula; and the technique the mastery of the sculptor when working with the tools and the materials. Anyone can try sculpture, and most people can create sculptures. These kind of models are evaluated by their aesthetic value, not necessarily their usefulness. But if the sculpture is poorly balanced and falls and

A. Páez and G. Boisjoly, *Discrete Choice Analysis with R*, Use R!,
https://doi.org/10.1007/978-3-031-20719-8_1

breaks, then its value is limited by its structural integrity—the skill of the sculptor matters even if only in this sense.

In the case of a mental map, the raw materials are ideas, the tools are a drawing surface and tools for writing, or maybe an app, and the technical expertise is the ability of the modeler to organize ideas in a useful way. To be useful, a conceptual model must be as complete, clear, and logical as possible. Figure 1.1 shows an example of a conceptual model of "change agents impinging on planet construction" (Fig. 15.1 in Greed 2000). A second conceptual model is shown in Fig. 1.2 where a vicious circle of car and public transportation is presented (Fig. 1.2 in Ortúzar and Willumsen 2011). Are the parts of the models clear? Can you identify things that were simplified in each of these two conceptual models? Do either of these models include superfluous or unnecessary elements? Are proposed relationships sensible? Are they complete?

Conceptual models are often a pre-requisite for the development of mathematical/statistical models. In fact, mathematical/statistical models are simply more formal versions of a conceptual (or theoretical) model. In the case of statistical models, the raw materials are data; the tools are descriptive statistics and statistical plots, and various forms of regression analysis; and the technical expertise is the ability of the modeler to select tools that are appropriate to the data, and to bring the data to "speak": in other words, to extract information from the data. As Moran said in the aphorism quoted at the beginning of this chapter, you can have data without informa-

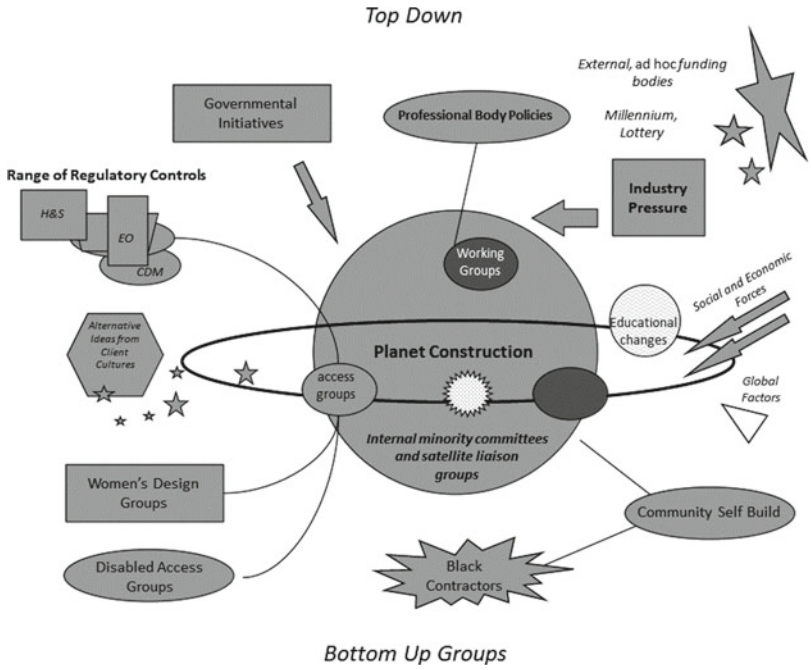


Fig. 1.1 Example of a conceptual model (adapted from Fig. 15.1 in Greed 2000)

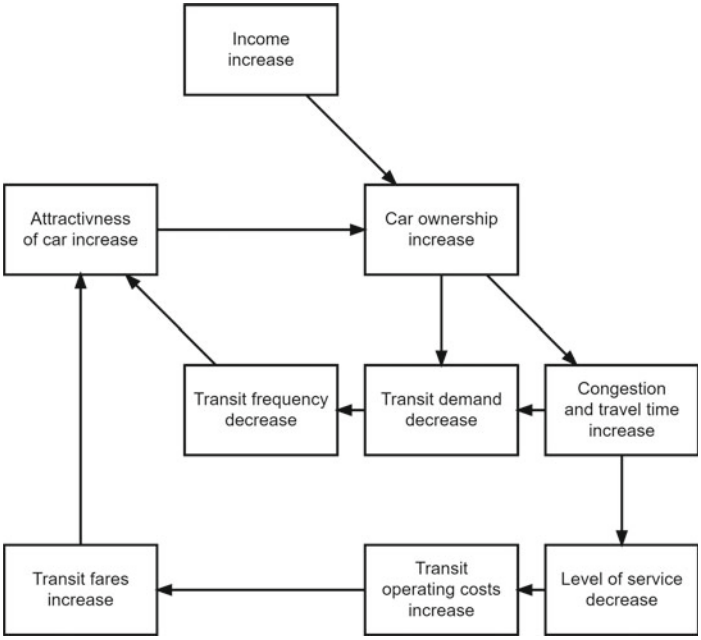

Income increase
Attractivness of car increase
Car ownership increase
Transit frequency decrease
Transit demand decrease
Congestion and travel time increase
Transit fares increase
Transit operating costs increase
Level of service decrease

Fig. 1.2 Example of a conceptual model (adapted from Fig. 1.2 in Ortúzar and Willumsen 2011)

tion, but not information without data. Technical mastery is the ability of a modeler to obtain information from data that is useful, accurate, and precise, to the extent that the raw materials permit.

Before moving on to the technical skills required for modeling, it is important to understand the raw materials and the tools. The objective of this chapter is to introduce some important concepts concerning data and data manipulation, and some useful tools, with an emphasis on the unique features of discrete data.

1.2 How to Use This Chapter

The source for the document you are reading is available as *code chunks*. Throughout the book you will find examples of code such as this one:

```
# Function `print()` displays the argument on the screen.
# `print()` is a generic method in `R` which means that many types
# of objects can be printed, including a string as in the example
# below:
print("Hello, Discrete Choice Analysis!")
```

```
[1] "Hello, Discrete Choice Analysis!"
```

If you are working with R or RStudio you can type the chunks of code to experiment with them. As an alternative, you may copy and paste the code into scripts or notebooks,[1] to save it and any experiments you may conduct. Whichever way you are working, you might want to give it a try now! You will see that the chunk of code above instructed R (and through R the computer) to `print` (or display on the screen) some text.

1.3 Learning Objectives

In this chapter, you will learn about:

1. Different ways to measure stuff.
2. Importing data and packages in R.
3. Data classes, data types, and data transformations.
4. Indexing and data manipulation.
5. Basic visualization.

1.4 Suggested Readings

- Grolemund, G., & Wickham, H. (2016). *R for data science*. **Chapters** 3–5, O'Reilly Media.
- Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*, **Chapters** 2–3. New York: Springer.

1.5 Ways of Measuring Stuff

Previously we said that data are the raw material for statistical models, but we did not say precisely what we meant by 'data'. You probably already have a working understanding of what 'data' means, but nonetheless we will begin with a definition. According to Merriam-Webster, data[2] are:

> factual information (such as measurements or statistics) used as a basis for reasoning, discussion, or calculation

[1] Notebooks are a form literate programming (see https://en.wikipedia.org/wiki/Literate_programming) a style of document that uses code to illustrate a discussion, as opposed to the traditional programming style that uses natural language to discuss/document the code. It flips around the usual technical writing approach to make it more intuitive and accessible.

[2] https://www.merriam-webster.com/dictionary/data.

As an aside, it is interesting to note that Tukey's classic book Exploratory Data Analysis (Tukey 1977), which established data analysis as a respectable field of mathematics, does not define 'data' in the glossary. Apparently, the meaning of data was self-evident!

Data are obtained from observation. Measurement theory is a branch of mathematics concerned with observing and recording the *facts* about something. It is important to note that measurements are not meant to be the same as the thing being measured; partly, this is because a measurement typically refers to a single, well-defined aspect, feature, or dimension of the thing being measured (e.g., its weight, or its density, or its height, or its price, but not everything at once). That said, we would like the measurements to be a reasonably close approximation of the feature of the thing that is being measured—with "reasonable" in this case meaning that reasonable people can agree that the measurements are a sufficiently close approximation of the thing being measured (measurement theory breaks down in the face of unreasonable people). If this is not the case, the measurements obtained might be pretty useless, and an inadequate way to learn something valuable from the thing we are measuring (try measuring speed in units of sandwiches or churros).

One fundamental contribution of the scientific method has been to produce standardized ways of measuring things. How would you measure the following things?

- The temperature at which water freezes.
- The temperature at which nitrogen freezes.
- The length of a trip.
- Blood donations.
- Different brands of peanut butter.
- The value of a two bedroom apartment.
- Someone's opinion regarding taxes.

Generally, there are multiple ways of measuring something, but not all of them are necessarily appropriate, partly because the scales of measurement may result in some loss of information. The interpretation of a measurement, as well, depends on what the scale is.

We can distinguish between two broad scales of measurement.

1.5.1 Categorical Measurements

Categorical measurements assign a label or category to the thing being measured. For example, a way to measure different brands of peanut butter could be to measure their sugar content, their fat content, their consistency, and so on and so forth, and in this way describe what makes each brand unique. We could call this measurement a snoopy, for instance. This approach is not very practical, and not only because a snoopy is actually a combination of various more fundamental measures (we do this all the time: speed, as an example, is a combination of two more fundamental measures, distance and time). But measuring all these things may be impractical for the objective at hand. A different way to measure peanut butter would be to label one brand "Spooky" and another "Peter's". This has the effect of reducing a large amount of information into much simpler categories. Is this loss of information inappropriate? Well, it really depends on what is the intended use of data! Categorical measurements are interesting because they may tell us something about the power of brands. They are also relevant when specific characteristics cannot be directly measured or observed.

Within the class of categorical variables there are two distinct scales of measurement:

- Nominal scale. When the categories do not follow a natural order. For example, there is no reason to say that the "Spooky" brand precedes "Peter's" or vice versa. Similarly, when measuring modes of travel "walking" does not come before, or is intrinsically higher or lower or better or worse than "cycling" or "riding bus".
- Ordinal scale. When the categories follow a natural or logical sequence. A common way of measuring opinions is by means of the Likert scale,[3] which classifies responses into levels, for instance as "strongly disagree", "disagree", "neutral", "agree", and "strongly agree". In this case, it is sensible to order the responses, since "strongly agree" is probably closer to "agree" than to "strongly disagree". Responses of this type are often represented by numbers, say, from 1 to 5, however, it is a mistake to treat the measurements as numbers instead of labels. When treated as numbers there is a temptation to think of the difference between 4 and 5 and the difference between 3 and 4 as being equivalent. In this case, it is possible that the strength of disagreement could be stronger than the strength of agreement (or viceversa). In other words, the interval between "strongly disagree" and "disagree" may not be the same as "agree" and "strongly agree". With ordinal scales we do not know that, all that we know is that they measure a different opinion that has a natural ordering.

It is interesting to note, as well, that sometimes different measurement scales might represent different behavioral mechanisms, as Bhat and Pulugurta discuss in their comparison of categorical and ordinal measurements for vehicle ownership, which respectively correspond to unordered and ordered response mechanisms (Bhat and Pulugurta 1998).

1.5.2 Quantitative Measurements

Quantitative measurements assign a number to an attribute, and the number quantifies the presence, strength, or intensity of the attribute being measured. Within this class of variables, there are also two ways of measuring things.

[3] https://en.wikipedia.org/wiki/Likert_scale.

- Interval scale. A quantity can be assigned to an attribute, the values follow an order, and their differences can be computed and remain constant. Temperature is typically measured in interval scale. The difference between 10 and 11 °C is the same as the difference between 25 and 26 °C. The intervals are meaningful. However, 0 °C does not imply the absence of temperature! Which is why measurements in Celsius and Fahrenheit do not coincide at zero. The lack of a natural zero for these scales means that the *ratios* between two values are not meaningful: 4 °C is not twice as hot as 2 °C, and −12 °C is not four times as cold as −3 °C.
- Ratio scale. When there is an absolute value of zero to the thing being measured (to indicate absence), attributes can be measured in a ratio scale. This combines the features of the previous scales of measurement: a number is essentially a label that follows a logical order and with differences that are meaningful. In addition to that, the ratios of variables are meaningful. For example, twenty dollars are twice as valuable as ten, and zero is the absence of value. Weight is a way of measuring mass, and zero is the absence of mass. Two hundred kilograms is twice as much mass as one hundred kilograms.

It is important to understand the different scales of measurement to be able to choose the appropriate tools for each. More on this later. But first, we can bring some actual data to play with.

1.6 Importing Data

There are several different ways to import data in R. For this example, we will use a data set that was analyzed by Whalen et al. (2013).

At the very beginning, it is good practice to clear the workspace, to ensure that there are no extraneous items there. The workspace is where objects reside in memory during a session with R. The function for removing variables from the workspace is rm(). Another useful function is ls, which retrieves a list of things in the workspace. So essentially we are asking R to remove all things in the workspace:

```
# Function `rm()` removes objects from the _environment_
# that is objects currently in memory. The argument list = ls()
# removes all objects currently in memory
rm(list = ls())
```

Once that the workspace is empty, we can proceed to load a few packages that are useful. Packages are the basic units of reproducible code in the R multiverse. Packages allow a developer to create a self-contained unit of code that often is meant to achieve some task. For instance, there are packages in R that specialize in statistical techniques, such as cluster analysis,[4] visualization,[5] or data manipulation.[6] Some

[4] https://cran.r-project.org/web/packages/cluster/index.html.

[5] https://cran.r-project.org/web/packages/ggplot2/index.html.

[6] https://cran.r-project.org/web/packages/dplyr/.

packages can be miscellaneous tools, or contain mostly data. Packages are a very convenient way of maintaining code, data, and documentation, and also of sharing all these resources.

Packages can be obtained from different sources (including making them!). One of the reasons why `R` has become so successful is the relative ease with which packages can be distributed. A package frequently used is called {tidyverse}. The {tidyverse} is a collection of functions for data manipulation, analysis, and visualization. This package can be downloaded and installed in your personal library of `R` packages by using the function `install.packages`, as follows:

```
install.packages("tidyverse")
```

The function `install.packages` retrieves packages from the Comprehensive `R` Archive Network,[7] or CRAN for short. CRAN is a collection of sites (accessible via the internet) that carry identical materials for distribution for `R`.

Installing a package is similar to acquiring a book for your library. The book is there, but if you want to use it, you need to bring it to your workspace, so to speak. The function for retrieving a package from the library is natural enough `library()`. For the moment, we need the following packages. If you have not done so, take a moment to install them, as illustrated in the previous chunk.

```
library(discrtr) # A companion package for the book Introduction to Discrete Choice Analysis with `R`
library(dplyr) # A Grammar of Data Manipulation
library(ggplot2) # Create Elegant Data Visualisations Using the Grammar of Graphics
library(mlogit) # Multinomial Logit Models
library(readr) # Read Rectangular Text Data
library(stargazer) # Well-Formatted Regression and Summary Statistics Tables
```

Occasionally there are messages displayed when loading a package. These messages are informative (they ask you to cite them in a certain style) or may give you warnings, for instance that identically named functions exist in several packages.

The function that we need to read the sample data set is `read_csv()`, which is part of the package {readr}. Note that you can name the value (or output) of a function by using `<-`. In this case, we wish to read an external file, and assign the results to an object called `mc_mode_choice`. The external file is part of the package {discrtr} and can be read as follows:

```
# Read a csv file data and name the object
mc_mode_choice <- read_csv(system.file("extdata",
                                       "mc_commute.csv",
                                       package = "discrtr"),
                           show_col_types = FALSE)
```

It is possible to quickly examine the contents of the object by means of the function `head()`, which prints the top few rows of the object, if appropriate. For example (showing only the first four columns):

[7] https://cran.r-project.org/.

```
# `head()` displays the first few rows of a data object
# Indexing of the object in this example is used to display
# only columns 1 through 4
head(mc_mode_choice[,1:4])
```

```
# A tibble: 6 x 4
  RespondentID choice avcycle avwalk
         <dbl>  <dbl>   <dbl>  <dbl>
1    566872636      3       0      1
2    566873140      3       0      1
3    566874266      3       0      0
4    566874842      2       1      1
5    566881170      2       1      1
6    566907438      2       0      1
```

Here we can see that what we just read is a table with several variables: a unique identifier for each respondent (`RespondentID`), a variable called `choice`, some variables for time, etc. Hopefully, when reading data there is also a metadata file, a data dictionary or something that defines what the data are.

For example, what does it mean for choice to be “3” or “1”? If we converted the variable `RespondentID` to unique names (e.g., Johnny McSupreme, Pepita González, etc.) would the variable still be meaningful? If so, are the numbers currently stored there really a quantity? Was time measured in hours, seconds, minutes, or something else?

These variables will be described below.

Before that, however, we can use a different function to get further insights into the contents of the table by means of the `summary()` function (output not shown):

Other packages in `R` can be used to produce elegantly formatted tables, including {stargazer}. Below, this package is used to tabulate the first five variables of table `mc_mode_choice`:

```
# `stargazer()` takes as an input a data frame
stargazer(as.data.frame(mc_mode_choice[,1:5]),
          # change the type to text, html, or latex depending on the desired output
          type = "latex",
          header = FALSE, # do not print package version info in the output
          title = "Example of a table with summary statistics", # Title of table
          omit.summary.stat = c("N",
                                "median"), # summary statistics to omit from output
          font.size = "small") # font size can be changed
```

This function will print a set of summary statistics for the variables in the table. The statistics presented by `R` are generally determined based on the scale of measurement of the variable—that is, the way the variable is coded. Presently, all summary statistics are calculated for quantitative variables (since all variables are coded that way as we will see below). We don’t know if this makes sense, until we know what the variables are supposed to measure (Table 1.1).

Table 1.1 Example of a table with summary statistics

Statistic	Mean	St. Dev.	Min	Max
RespondentID	570,566,454.000	3,786,118.000	566,872,636	587,675,235
Choice	2.618	0.845	1	4
Avcycle	0.275	0.447	0	1
Avwalk	0.661	0.473	0	1
Avhsr	0.961	0.194	0	1

For example, the variable `choice` measures the use of one mode of transportation. There are four values in this scale: 1 through 4, with each indicating one of "Cycle", "Walk", "Car", or "HSR" (the local transit agency in Hamilton, ON, Canada, where these data were collected). Check the results of the summary. What does it mean to say that the mean of choice is 2.618? Does this number make sense? What about the mean of `RespondentID`?

1.7 Data Classes in R

To understand why 2.618 of mode of transportation is not an appropriate summary measure for the variable `mode`, we need to know that `R` can work with different data classes, which include the following:

- Numerical
- Character
- Logical
- Factor

The ability to store information in different forms is important, because it allows `R` to distinguish what kind of operations are appropriate for a certain variable. Consider the following example (using indexing):

```
# Indexing allows us to choose parts of a data object
# In this example, we are extracting the first row of
# column `choice` in table `mc_mode_choice` and then
# The fourth row of the same column
mc_mode_choice$choice[1] - mc_mode_choice$choice[4]
```

```
[1] 1
```

Let us unpack what this chunk of code does. First, we call our table `mc_mode_choice`. The string sign `$` is used to reference columns in the table. Therefore, we asked `R` to go and look up the column `choice` in the table `mc_mode_choice`. Finally, the number between square brackets `[]` asks `R` to retrieve a specific element out of the column, in this example the first element in that column and then the fourth element. This system of referring to elements in tables is called *indexing*. Most computer languages use it, but the syntax is different. Again: `$` refers to a column, and `[]` is used to call elements in that column.

As we can see, the difference between the two values (elements) retrieved is 1. But what is the meaning of "cycle" minus "walk", for instance?

In reality, the variable `choice` was measured as a nominal variable: it just corresponds to a label indicating what mode was chosen by a respondent. But `R` does not know this. Before `R` can treat it as a nominal variable, the numbers need to be converted to a *factor*. Factors are the way `R` stores categorical variables (both nominal and ordinal). To convert the variable `choice` to a factor, we use the `factor()` function:

```r
# Function `factor()` is used to convert a variable (which could be character or numeric)
# into a factor, that is, a label or category; when we want a factor to be ordered (treated
# as an ordinal variable) we specify argument ordered = TRUE. Non-ordinal variables by default
# are displayed alphabetically, but changing their order when  specifying the labels changes
# the order they are displayed _without necessarily making them ordinal_
mc_mode_choice$choice <- factor(mc_mode_choice$choice,
                                labels = c("Cycle",
                                           "Walk",
                                           "HSR",
                                           "Car"))
```

In the chunk above, we ask `R` to replace the contents of `mc_mode_choice$choice` with the value (output) of the function `factor`. Factor takes the contents of `mc_mode_choice$choice` and converts to a factor with labels as indicated by the argument `labels =` (the function `c()` is used to concatenate several values).

We can summarize the result, by using the `summary()` function but only for this variable:

```r
summary(mc_mode_choice$choice)
```

```
Cycle  Walk   HSR   Car 
   48   711   336   281 
```

Now the summary is appropriate for a categorical variable, and is a table of frequencies: as seen there, there were 48 respondents who chose "Cycle", 711 who chose "Walk", and so on. What if we tried to calculate the difference?

```r
mc_mode_choice$choice[1] - mc_mode_choice$choice[4]
```

```
Warning in Ops.factor(mc_mode_choice$choice[1], mc_mode_choice$choice[4]): '-'
not meaningful for factors
```

```
[1] NA
```

The message indicates that the operation we tried to perform is not meaningful for factors. As long as R knows the appropriate measurement scale for your variables, it will try to steer you away from doing silly things with them.

Other variables included in this table relate to time. These variables measure the duration in minutes (actual or imputed) for trips by different modes. For example, timecycle is the duration of a trip by bicycle for the journey reported by the respondent.

Let us summarize this variable again:

```
summary(mc_mode_choice$timecycle)
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
     0.29      3.79      5.83  34014.86 100000.00 100000.00
```

Notice that the shortest trip by bicycle would be less than a minute long, whereas the maximum is $100,000$ min long. Wait, what? That is over $1,600$ h long. Is that even possible? In fact, no, it is not. The reason for these values is that when the original data were coded, whenever a respondent said that cycling was not a mode that was available to them, the time was coded as a very large and distinctive value. There were no trips taking $100,000$ min, this is just a code for "information not available". One problem with this manner of coding is that R does not know that the information is actually missing, but rather thinks it is a legitimate quantity. As a consequence, the mean is tens of thousands of minutes, despite the fact that half of all trips by bicycle were measured at less than 6 min long (see the median).

Next we will see a way to address this. One last thing before doing so: you can check the class an object with the function class

```
# Find the class of an object
class(mc_mode_choice$choice)
```

```
## [1] "factor"
```

```
class(mc_mode_choice$timecycle)
```

```
## [1] "numeric"
```

The RStudio application also allows you to quickly see the variables and their format in the Environment panel.

1.8 More on Indexing and Data Manipulation

Indexing is a way of making reference to elements in a data object. There are numerous indexing methods in R that are appropriate for specific objects. Tables such as mc_mode_choice (called *data frames* in R) can be indexed in a few different ways. For example the next three chunks are equivalent in that they call the second column (choice) and in that column the second element:

```
mc_mode_choice[2, 2]
```

```
# A tibble: 1 x 1
  choice
  <fct>
1 HSR
```

```
mc_mode_choice$choice[2]
```

```
[1] HSR
Levels: Cycle Walk HSR Car
```

```
mc_mode_choice[["choice"]][2]
```

```
[1] HSR
Levels: Cycle Walk HSR Car
```

It is also possible to index by ranges of values. For example, the next chunk retrieves rows 2–5 from columns 7 and 8:

```
mc_mode_choice[2:5, 7:8]
```

```
# A tibble: 4 x 2
  timecycle timewalk
      <dbl>    <dbl>
1      3.73     12.8
2 100000    100000
3      5.83     20
4      5.83     20
```

Indexing is useful to subset data selectively. For example, we know that travel times coded as 100, 000 are actually cases where the corresponding mode was not available. Suppose that we wanted to summarize travel time by bicycle but without those cases. We can use logical statements when indexing. We could tell `R` to retrieve only those values that meet a certain condition. In the next chunk, we save the results of this to a new variable:

```
time.Cycle.clean <- mc_mode_choice$timecycle[mc_mode_choice$timecycle != 100000]
```

where `!=` is `R` for *not equal to*. In other words, "find all values *not equal to* 100000, and retrieve them".

The result of this is a numeric object:

```
class(time.Cycle.clean)
```

```
## [1] "numeric"
```

If we summarize this object now:

```
summary(time.Cycle.clean)
```

```
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.2914  2.9141  4.3711  4.8957  5.8282 45.0000 
```

The summary statistics are much more sensible: the longest trip by bicycle was measured at 45 min, and the mean trip at less than 5 min.

Indexing is a powerful technique, but can be cumbersome (`mc_mode_choice$timecycle[mc_mode_choice$timecycle != 100000]`!). The package {dplyer} (part of the {tidyverse}) provides a grammar for data manipulation that is more intuitive. We will explore three of its elements here, namely the pipe operator (`%>%`), `select`, and `filter`.

Suppose that we wanted to select two of the time variables, for cycling and walking, and wanted to retrieve only values other than the offending 100, 000, and save these values in a new object called `time.Active.clean`. In the grammar of {dplyr}, this is done as follows:

```
time.Active.clean <- mc_mode_choice %>% # Pipe data frame `mc_mode_choice`
  select(c("timecycle", # Select columns from the data frame that was piped
           "timewalk")) %>%
  filter(timecycle != 100000 & timewalk != 100000) # Filter observations that are _not_ 100000
```

In natural language this would be something like "take mc_mode_choice and select columns timecycle and timewalk; pass the result to filter and retrieve all rows that meet the conditions `timecycle != 100000` AND `timewalk != 100000`. The verb `select` is used to select columns from a data frame, and the verb `filter` to filter rows.

The alternative, using indexing would look something like this:

```
time.Active.clean.the.hard.way <- mc_mode_choice[mc_mode_choice$timecycle != 100000 &
                                                  mc_mode_choice$timewalk != 100000, 7:8]
```

The expression becomes more convoluted and not as easy to read. It is also easier to make mistakes when writing it.

Compare the summaries of the two data frames, to make sure that they are identical:

```
summary(time.Active.clean)
```

```
   timecycle           timewalk
 Min.    : 0.2914   Min.    : 1.00
 1st Qu.: 2.9141   1st Qu.:10.00
 Median : 4.3711   Median :15.00
 Mean    : 4.5852   Mean    :16.10
 3rd Qu.: 5.8282   3rd Qu.:20.00
 Max.    :17.4845   Max.    :62.11
```

```
summary(time.Active.clean.the.hard.way)
```

```
   timecycle           timewalk
 Min.    : 0.2914   Min.    : 1.00
 1st Qu.: 2.9141   1st Qu.:10.00
 Median : 4.3711   Median :15.00
 Mean    : 4.5852   Mean    :16.10
 3rd Qu.: 5.8282   3rd Qu.:20.00
 Max.    :17.4845   Max.    :62.11
```

The grammar of data manipulation in {dplyr} is a powerful way of working with data in an intuitive way. We will explore several other aspects of this grammar, but for the time being you are welcome to consult more about {dplyr} here.[8]

1.9 Visualization

The last item in this section is related to data visualization (much more on this in Chap. 2). Humans are very much visual creatures, and much can be learned from *seeing* the data. For example, the data frame, in essence a table, is informative in many ways, but does not make it particularly easy to observe trends or regularities in

[8] https://dplyr.tidyverse.org/.

the data. The summary statistics are also informative, but partial, and do not convey information to the same effect as a statistical plot. For visualization, this book relies mainly on the package {ggplot2}.[9] Like {dplyr} {ggplot2} implements a grammar, but in its case the grammar is for graphics (the name stands for "grammar of graphics for 2-D plots").

Take the following list of summary statistics:

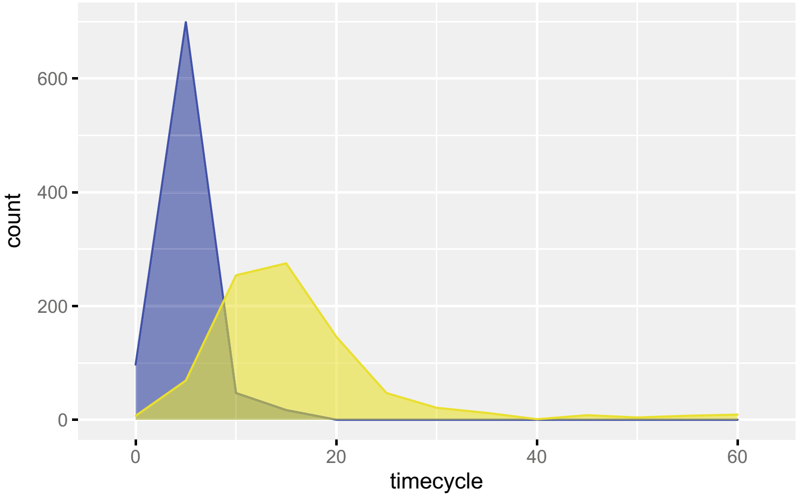
```
summary(time.Active.clean)
```

```
   timecycle          timewalk
 Min.    : 0.2914   Min.    : 1.00
 1st Qu.: 2.9141   1st Qu.:10.00
 Median : 4.3711   Median :15.00
 Mean    : 4.5852   Mean    :16.10
 3rd Qu.: 5.8282   3rd Qu.:20.00
 Max.    :17.4845   Max.    :62.11
```

Now, compare these summary statistics to the following plot (do not worry too much about the code just now):

```
ggplot(data = time.Active.clean) +
  geom_area(aes(x = timecycle),
            stat = "bin",
            binwidth = 5,
            fill = "blue",
            color = "blue",
            alpha = 0.6) +
  geom_area(aes(x = timewalk),
            stat = "bin",
            binwidth = 5,
            fill = "yellow",
            color = "yellow",
            alpha = 0.6)
```

[9] https://ggplot2-book.org/.



Package {ggplot2} is also part of the {tidyverse} and works by layering a series of objects, beginning with a blank plot, to which we can add things. The command to create a plot is `ggplot()`. This command accepts different arguments. For instance, we can pass data to it in the form of a data frame. We can also indicate different *aesthetic* values, that is, the things that we wish to plot and what aspect of the plot they map to (i.e., does a variable map to the x-axis? Or to the y-axis? To the color of lines, or to the color of the areas of polygons?).

None of this is plotted, though, until we indicate which kind of *geom* or geometric object we wish to plot. For instance, you can see above that to create the figure we used `geom_area`. The corresponding geometric object is essentially a smoothed histogram.

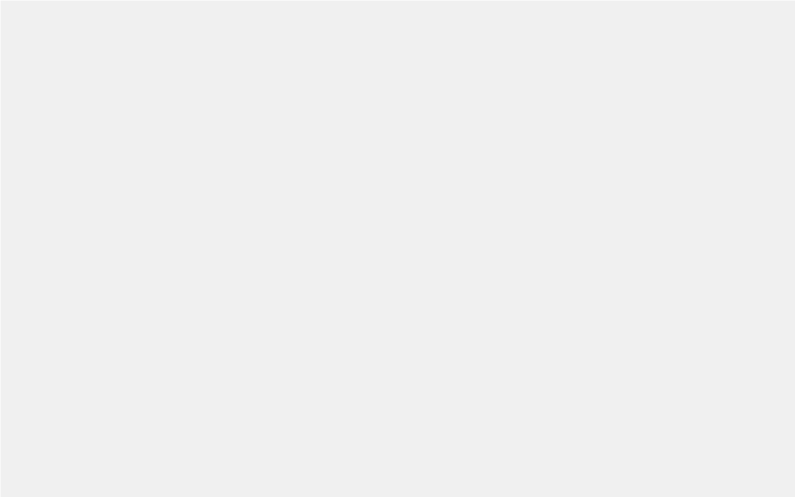
Let us break down these instructions.

First we ask {ggplot2} to create a plot that will use the data frame `time.Active.clean`. The plot is a data object, so we can give it a name to place it in memory. In this example, we will name this object `p`:

```r
# Initialize a `ggplot` object that will use table `time.Active.clean`
# as an input, and name it `p`
p <- ggplot(data = time.Active.clean)
```

Notice how `ggplot2` creates a blank plot, but it has yet to actually render any of the population information in there:

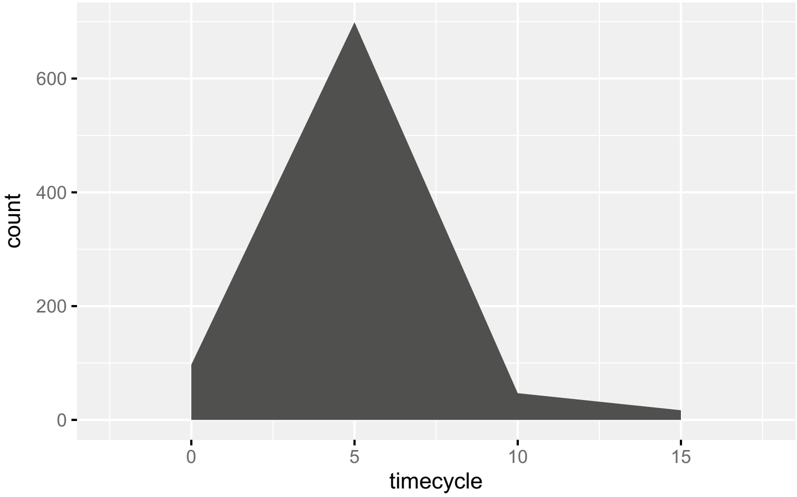
```r
# By typing the name of a ggplot object, the default
# behavior is to render it
p
```



We have yet to tell {ggplot2} what is the x-axis, what is the y-axis, what should be plotted on the x-axis, and so on.

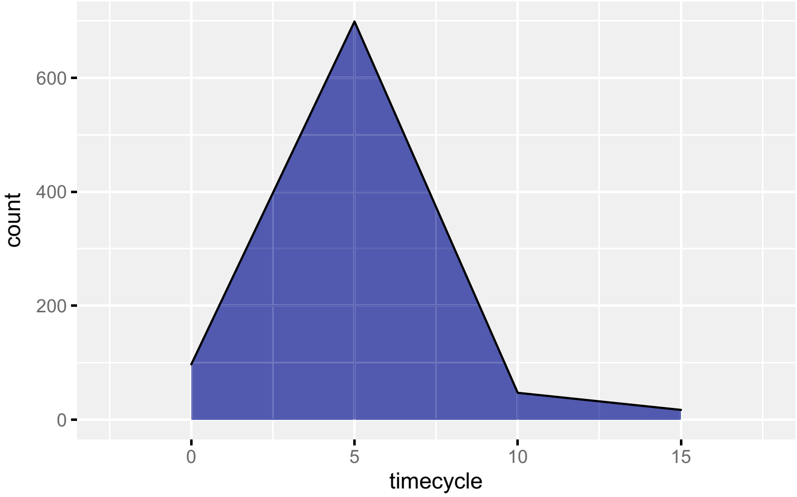
We layer elements on a plot by using the + sign. It is only when we tell the package to add one or more geometric elements that it renders something on the plot. In the current example case, we told {ggplot2} to use `geom_area()` to create a smoothed histogram. Next, we need to indicate which `aes` (short for *aesthetics*) we wish to plot. The aesthetics map aspect of the data set to the plot. For instance, by saying that the aesthetics include something for x, we tell {ggplot2} that something should be mapped to the x axis (in this case one of the time variables in the data frame). The second argument is `stat = 'bin'`, which indicates that there is a statistical operation that happens, namely the data are binned for smoothing (small bins lead to less smoothing, large bins lead to more smoothing; try it!). After playing around with a few bin values, a width of 5 was selected:

```
p +
  # Add a geometric object of type area to the plot
  # Map the variable `timecycle` to the x-axis. Notice
  # that the y-axis is a calculated statistic, the count
  # of cases (returned by stat =bin), so we do not need
  # to specify it
  geom_area(aes(x = timecycle),
            stat = "bin",
            # The bindwidth controls the size of the bins
            # needed to count the number of cases at levels
            # of the variable mapped to the x-axis
            binwidth = 5)
```



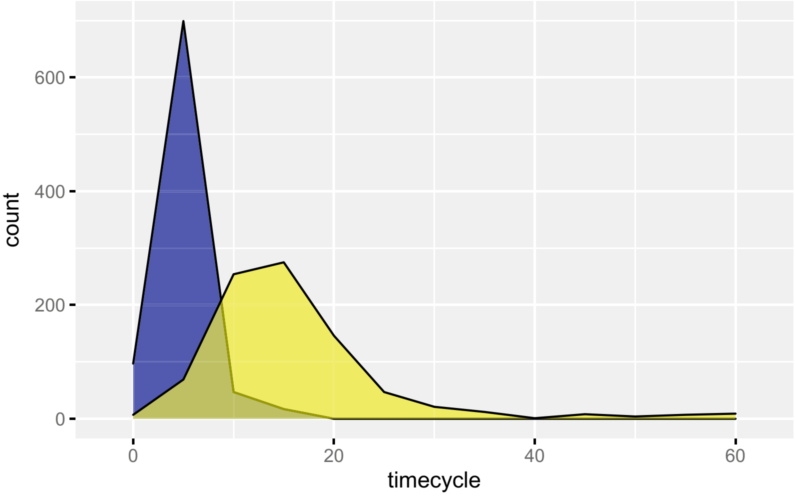
To improve the appearance of the plot, we also asked that the geom be rendered using a named color (blue for the color of the line, and also blue for the fill), and that it be transparent (the argument `alpha` controls opacity; a value of zero is transparent, a value of 1 is solid):

```
p +
  geom_area(aes(x = timecycle),
            stat = "bin",
            binwidth = 5,
            # fill controls the color of the polygon
            fill = "blue",
            # color controls the color of the perimeter
            # of the polygon or of lines more generally
            color = "black",
            alpha = 0.6)
```



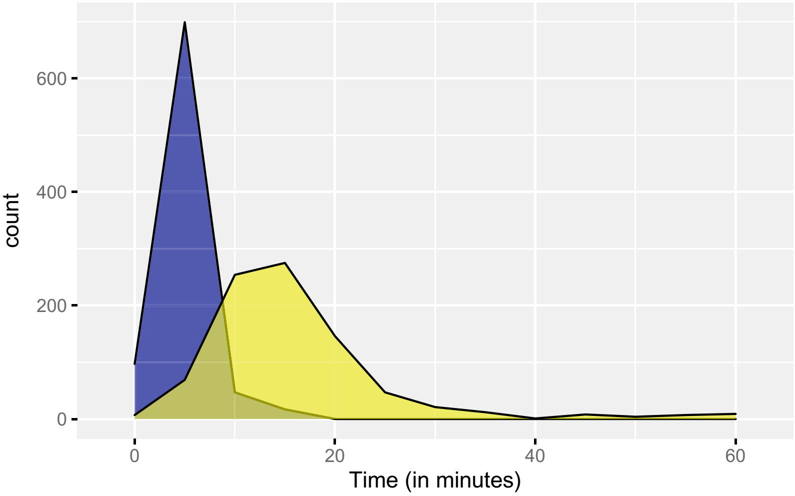
The final plot was obtained by layering a second geom (in yellow) for a different variable (time by walking), so that we could compare them:

```
ggplot(data = time.Active.clean) +
  geom_area(aes(x = timecycle),
            stat = "bin",
            binwidth = 5,
            fill = "blue",
            color = "black",
            alpha = 0.6) +
  # We can plot a second geometric element to the x-axis
  # using a different variable from the same table
  geom_area(aes(x = timewalk),
            stat = "bin",
            binwidth = 5,
            fill = "yellow",
            color = "black",
            alpha = 0.6)
```



Notice that the x-axis is labeled as “timecycle” despite the fact that the plot also includes time by walking. This can be fixed by changing the label as follows:

```
ggplot(data = time.Active.clean) +
  geom_area(aes(x = timecycle),
            stat = "bin",
            binwidth = 5,
            fill = "blue",
            color = "black",
            alpha = 0.6) +
  geom_area(aes(x = timewalk),
            stat = "bin",
            binwidth = 5,
            fill = "yellow",
            color = "black",
            alpha = 0.6) +
  xlab("Time (in minutes)")
```



What do we learn from this plot? Would it have been possible to learn the same from the summary statistics? Which was more effective, the plot or the summary statistics?

The plot above is an example of a *univariate* plot, since it is created to display the distribution of a single variable, not the way two or more variables relate. Imagine now that you would like to see how mode choice and sidewalk density at the place of residence relate. An appropriate statistical plot for two variables, one of which is nominal (`choice`) and another that is continuous (`side_den`), is the boxplot.

Before creating the plot, let us summarize these two variables (notice the use of the pipe operator):

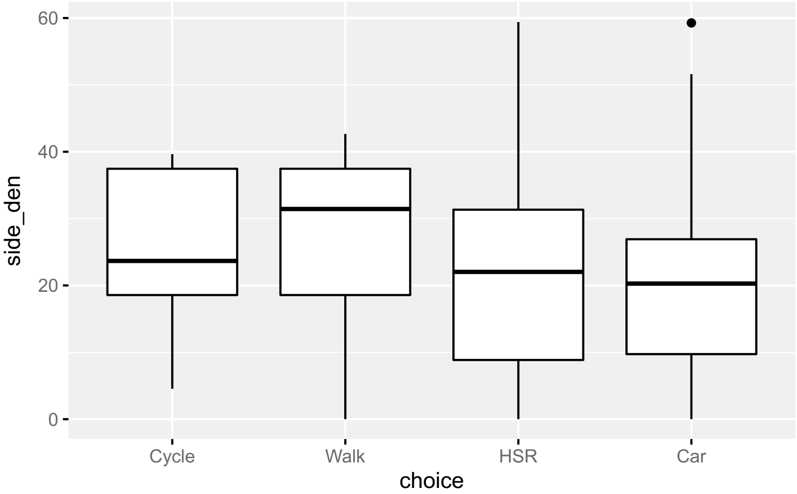
```
# The pipe operator `%>%` takes an object and passes it on
# to the next function where it is used as the first argument
mc_mode_choice %>%
  # `select()` retrieves columns from a data frame
  select(c("choice", "side_den")) %>%
  summary()
```

```
   choice        side_den
 Cycle: 48   Min.   : 0.00
 Walk :711   1st Qu.:18.19
 HSR  :336   Median :22.63
 Car  :281   Mean   :24.18
             3rd Qu.:35.70
             Max.   :59.41
```

Sidewalk density is measured in km/km^2.

We will create the boxplot next. We begin by defining a {ggplot2} object with the data frame and aesthetics that we wish to use. In this case, we want to plot the categorical variable on the x-axis and the quantitative variable on the y-axis:

```
# Pipe the table to `ggplot()` where it is assumed to be the
# first argument of the function, i.e., data
mc_mode_choice %>%
  # Map `choice` to the x-axis and `side_den` to the y-axis
ggplot(aes(x = choice,
           y = side_den)) +
  # Add a geometric object of type boxplot
  geom_boxplot()
```



What do we learn from this plot? Could we have derived a similar insight from the summary statistics?

There are many different geometric objects, or `geoms_*` that can be used in {ggplot2}. You can always consult the help/tutorial files by typing `?ggplot2` in the console. See:

```
?ggplot2
```

You can also check the {ggplot2} Cheat Sheet[10] for more information on how to use this package.

A last note. There are many other software packages that are useful for creating visualizations (for instance, Excel). Many of these packages provide point-and-click functions for creating plots. In contrast, creating plots programmatically, as we illustrated in this chapter with {ggplot2} in `R`, requires that the plot be created meticulously, by instructing the package what to do step by fastidious step. While this is more laborious, it has at least two important advantages: (1) you have complete

[10] https://raw.githubusercontent.com/rstudio/cheatsheets/main/data-visualization.pdf.

control over the creation of plots, which in turn allows you to create more flexible and creative visuals; and (2) you can also easily recreate the figures if your data changes, for instance if you are analyzing the same data set after cleaning an error in your initial database, or a different data set with similar variables, without having to point-and-click all your way to a plot again.

1.10 Exercises

1. Define “model”.
2. Why are models on a 1-to-1 scale undesirable?
3. Invoke data set `Mode` from package `mlogit`. To do this you need to first load the package. This is a data set with choices about mode of transportation. This is done as follows:

```
library(mlogit)
data("Mode")
```

Once you have loaded the data set, answer the following questions:

4. Describe this data set. How many variables are there and of which type (i.e., categorical/quantitative)?
5. How many different modes of transportation are in this data set? What is the most popular mode? What is the least popular mode?
6. In general, what is the most expensive mode? The least expensive?
7. Create a plot showing the univariate distributions of time by car and time by bus. Discuss.
8. How do choices relate to cost by the different modes?

References

Bhat, C. R., & Pulugurta, V. (1998). A comparison of two alternative behavioral choice mechanisms for household auto ownership decisions. *Transportation Research Part B, 32*(1), 61–75.

Greed, C. (2000). *Introducing planning* (3rd ed.). NY: Bloomsbury Academic.

Ortúzar, J. D., & Willumsen, L. G. (2011). *Modelling transport* (4th ed.). New York: Wiley.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Reading, Massachussets: Addison-Wesley Publishing Company.

Whalen, K. E., Páez, A., & Carrasco, J. A. (2013). Mode choice of university students commuting to school and the role of active travel. *Journal of Transport Geography, 31*, 132–42. https://doi.org/10.1016/j.jtrangeo.2013.06.008.

Chapter 2
Exploratory Data Analysis

The greatest value of a picture is when it forces us to notice what we never expected to see.

— John Tukey (1977, vi)

2.1 Why Exploratory Data Analysis?

Discrete choice modeling covers a family of techniques useful to infer decision-making processes in many disciplines, including economics, geography, transportation engineering and planning. These techniques are well represented in a variety of journals, including specialized outlets such as the Journal of Choice Modeling, and is a preferred tool in many applications due to the rich behavioral interpretation of the models.

Thousands of applications of discrete choice analysis are found in the literature, which have greatly contributed to our understanding of individual behavior and implicit preferences. This includes studies on transportation mode choices, altruistic behavior, residential choices, and so on. Experienced modelers know well that estimating and interpreting a discrete choice model are only two aspects of a more extended data analysis process, one that ranges from data collection to the presentation of results to inform policy and decision-making. As hinted at in Chap. 1, modeling is partly science and partly art. The raw materials in the art of modeling are the data, and the tools in discrete choice analysis are the different statistical techniques used to reveal the underlying patterns. From a data analysis perspective, it is important for a modeler to understand the characteristics of the raw materials that they will work with. To accomplish this task, a good understanding of the tools in the toolbox of Exploratory Data Analysis (EDA) is appropriate.

A. Páez and G. Boisjoly, *Discrete Choice Analysis with R*, Use R!,
https://doi.org/10.1007/978-3-031-20719-8_2

Exploratory Data Analysis, while essential, has not always enjoyed the cache of modeling. Tukey, in his classical text, likens EDA to detective work (Tukey 1977, 1). With a heavy emphasis in statistics and econometrics on *confirmatory analysis*, the importance of basic detective work, the equivalent of checking for fingerprints well before a trial, is often overlooked. EDA, in fact, is an aspect of modeling that often seems to be taken for granted (or worse: as Tukey notes, it might be "decried as 'mere descriptive statistics' "). Particularly in the short format of journal articles, where space to present background, data, analysis, and results is limited, any exploratory data analysis conducted often goes unreported, and virtually none of the available texts that cover discrete choice analysis discuss it in any level of detail, despite the important role that it plays in the development and use of discrete choice models.

The lack of discussion of EDA in the literature on discrete choice analysis may be due to the somewhat informal, creative, and often playful nature of data exploration—somewhat of a contrast to the more formal, almost liturgical tone used in the confirmatory analysis. As the name suggests, the focus with EDA is to approach the data at hand with as few assumptions as possible, and to let the data speak for themselves, as it were. The objective is to make a (possibly large) collection of observations easier for a brain to manage and understand. Accordingly (see Tukey 1977, v):

1. EDA aims to *simplify* descriptions to make them easier to handle with available cognitive power; and
2. Tries to look *below* previously described surfaces to make the description more effective.

The main tools of EDA are descriptive statistics and visualization techniques. Visualization, in particular, can be enlightening. As Franconeri et al. (2021) note in a recent review, sight engages several parallel processing channels that allow the human brain to detect patterns in a highly efficient fashion. These channels include positions, shapes, lengths, areas, angles, colors, and intensities, to name a few. Humans excel at recognition of spatial relationships, and two-dimensional representation of numerical data encoded in these channels can engage some of the most powerful parts of the brain.

As an example, consider Table 2.1.

This table presents the first few rows of selected variables in data frame `mc_commute_wide` (available from package {discrtr}, see "Getting started" in the Preface). As briefly discussed in Chap. 1, this data set includes information about modal choices (mode of travel) for respondents to a travel survey as reported by Whalen et al. (2013). The columns include the density of the sidewalk and street networks at the place of residence of respondents, the geocoded location (latitude–longitude), as well as the mode used to travel to school, in addition to the gender and living arrangement of the respondent.

Table 2.1 First eight rows of selected variables in data frame

Street_density	Sidewalk_density	LAT	LONG	Choice	Gender	Shared	Family
14.376	22.633	43.263	−79.901	HSR	Man	Living in shared accommodations	No
19.498	39.640	43.259	−79.905	HSR	Man	No	No
13.557	8.228	43.252	−79.940	HSR	Man	No	Living with family
14.308	37.458	43.258	−79.919	Walk	Man	Living in shared accommodations	No
14.308	37.458	43.256	−79.920	Walk	Woman	Living in shared accommodations	No
11.884	18.567	43.258	−79.922	Walk	Man	Living in shared accommodations	No
15.394	26.068	43.256	−79.913	Walk	Man	No	Living with family
10.108	20.282	43.218	−79.918	HSR	Man	No	Living with family

Scanning the table visually (go ahead, try it!) will probably convince you that visual inspection is not the most effective way of searching for patterns among all the alphanumeric data. The reason is that most humans can process only a handful of numeric symbols per second. A search for patterns, moreover, increases in difficulty as the size of the number of symbols grows because new numbers or patterns tend to displace others previously perceived and stored in short-term memory. This is just an innate limitation of human cognition (Franconeri et al. 2021, 113). In contrast, the largest single processing system in the brain deals with two-dimensional visual recognition (Essen et al. 1992).

In contrast to the table, Fig. 2.1 displays four of the same dimensions (street and network density, latitude and longitude) plus two more (the urban/suburban classification). That is six dimensions represented in a single figure, almost fourteen hundred data points, and yet the brain more readily detects patterns when the inputs engage visual channels associated with spatial recognition. Compared to inspecting the table, we can more easily perceive the *locations* where the street and sidewalk density are both high or low, as well as the spatial distribution of the observations, the geographical coverage of the survey, and the types of development in the region. This all happens practically faster than the blink of an eye (Franconeri et al. 2021, 123).

Descriptive statistics are another way of “flattening” a data set, to facilitate seeing it from a simplified perspective. The data frame `mc_comute_wide` (of which we

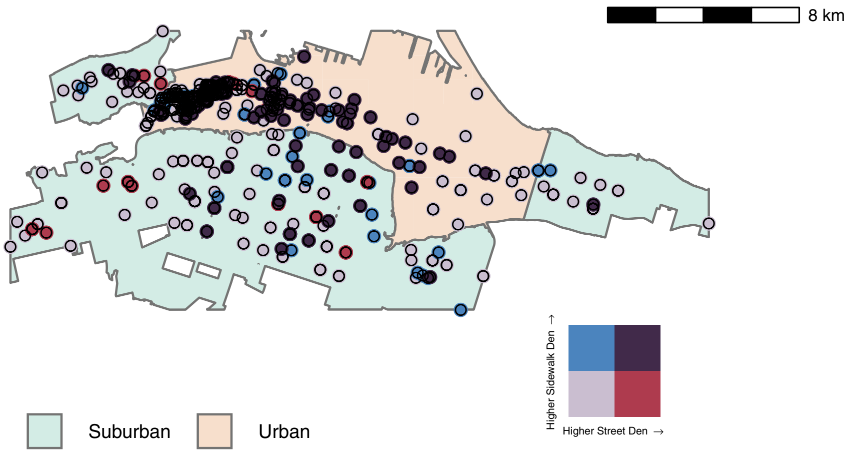


Fig. 2.1 Multidimensional data visualization: map

saw only the first few rows of selected variables in Table 2.1) can be summarized as a table of frequencies (see Table 2.2). The table is essentially a three-way cross-tabulation, showing the number of responses by three categorical variables (gender, living arrangements, and mode choice). This way of summarizing the information compresses more than thirteen hundred rows of data to only 24 rows, while excluding other information (EDA *always* excludes some information, since the point is to simplify). Even so, a visual representation of the data can provide deeper insights by allowing us to look below the surface of the table. Figure 2.2 shows an alluvial plot of the exact same information shown in Table 2.2.

The figure makes it clear that exploiting the multi-channel processing capabilities of sight is superior to trying to see patterns from the alphanumerical table. For example, we can immediately see that the sample includes more women than men; we can also see that shared accommodations (living in a rental property with other students) is the most common form of residential arrangement; and that walking is the most popular mode of transportation in the sample. Of course, we could have retrieved all these quantities from the table of frequencies, but it would have been more time consuming and tedious, and hence less efficient. Instead, when coded in the form of shapes, areas, and colors, the brain can make virtually immediate sense of the underlying values. Furthermore, there are additional aspects of the table that would have been harder to grasp, since they lay somewhat under the surface. In this way, the alluvial plot also makes it easier to understand relationships between the distributions of the three categorical variables, and so we notice that those who stayed in the family home are more likely to be women, that the vast majority of students who walk to school are those who live in shared accommodations, and that women tend to walk more than men (the women-to-men ratio in the sample is NaN, but NaN among those who walk).

Table 2.2 Table of frequencies by three categorical variables

Gender	Living_arrangements	Choice	Frequency
Woman	Family	Cycle	1
Woman	Family	Walk	12
Woman	Family	HSR	71
Woman	Family	Car	126
Woman	Other	Cycle	4
Woman	Other	Walk	24
Woman	Other	HSR	43
Woman	Other	Car	26
Woman	Shared	Cycle	15
Woman	Shared	Walk	405
Woman	Shared	HSR	79
Woman	Shared	Car	18
Man	Family	Cycle	1
Man	Family	Walk	6
Man	Family	HSR	50
Man	Family	Car	74
Man	Other	Cycle	5
Man	Other	Walk	19
Man	Other	HSR	30
Man	Other	Car	24
Man	Shared	Cycle	21
Man	Shared	Walk	245
Man	Shared	HSR	63
Man	Shared	Car	13

These are but two examples of EDA techniques applied to a data set that includes categorical variables. The aim of this chapter is to introduce some valuable tools useful to uncover relevant patterns in the data, with an emphasis on understanding the most appropriate tools to visualize and communicate these patterns. While a majority of readers are probably more familiar with the analysis of quantitative variables (given that we are usually exposed to them from earlier in our education), the discrete nature of at least one variable in DCA calls for specialized approaches.

In terms of the timing of using EDA, the task of exploration is seldom linear, and a modeler will often find that they need to revisit the data to verify, contrast, or refute their assumptions. That said, in fairly general terms, we can think of EDA as being useful before a model is estimated (pre-estimation) and after a model has been estimated (post-estimation). The focus at each stage is different: pre-estimation, the modeler's intent is often to learn as much as possible from potential patterns in the data; post-estimation, the intent shifts to effectively communicating the results

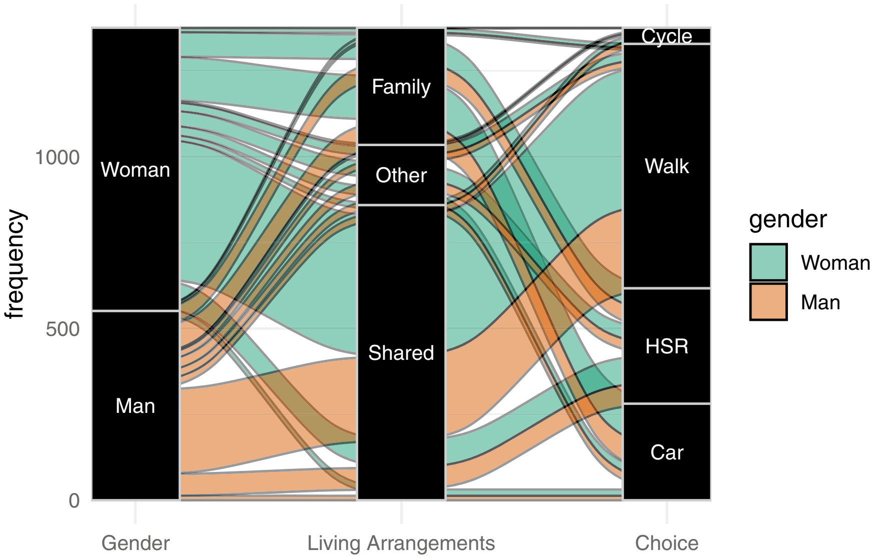


Fig. 2.2 Multidimensional data visualization: alluvial plot

of models. In this chapter, we will focus on pre-estimation EDA. It is mainly a knowledge discovery technique that allows:

- Exploring the various variables and detecting outliers.
- Exploring the relationships between the variables and formulating hypotheses.
- Detecting, revealing, describing and representing patterns of association and underlying structures of dependencies.

Pre-estimation EDA can be divided in two parts:

1. Exploring the variables independently from each other (univariate analysis). Often we are interested in the distributions of variables.
2. Exploring the relationships between the variables (bivariate and multivariate analyses).

Furthermore, the choice of tools depends on whether the variables under examination are all categorical, or a combination of variables in categorical and quantitative scales. The techniques discussed in this chapter are also the foundation for post-estimation data analysis, and we will return to them as appropriate in subsequent chapters.

2.2 How to Use This Chapter

The source for the document you are reading is available as *code chunks*. Throughout the book, you will find examples of code such as this one:

```
# Function `print()`displays the argument on the screen.
print("Is that a bar chart inyour pocket, or are you just happy to see me?")
```

```
[1] "Is that a bar chart in your pocket, or are you just happy to see me?"
```

If you are working with RStudio, you can type the chunks of code to experiment with them. As an alternative, you may copy and paste the source code into your `R` or RStudio console, or create a script/notebook to save the code and any experiments you may conduct.

2.3 Learning Objectives

In this chapter, you will learn about:

1. Different ways to explore data sets that contain categorical variables.
2. Bivariate and multivariate visualization techniques.

2.4 Suggested Readings

- Buliung, R. N., & Morency, C. (2010). “Seeing is believing”: Exploring opportunities for the visualization of activity–Travel and land use processes in space–Time. In A. Páez, J. Le Gallo, R. N., & Buliung, S., Dall’erba (Eds.), *Progress in spatial analysis, progress in spatial analysis*, pp. 119–47. Springer.
- Knaflic, C. N. (2015). *Storytelling with data: A data visualization guide for business professionals*. Wiley
- Tukey, J. W. (1977). *Exploratory data analysis*. Reading, Massachussets: Addison-Wesley Publishing Company.

2.5 Preliminaries

As a first step, it is good practice to clean the work space

```
# Function `rm()` removes objects from the environment
# i.e., it clears the working memory
rm(list = ls())
```

We will now load the packages that will be used in this chapter. If you have not previously installed these packages on your work station, make sure to do so now. Note that R will give you a message of error if you try to load a package that is not installed on your work station. This can be used as a reminder to know which packages you have already installed.

```
library(discrtr) # A companion package for the book Introduction to Discrete Choice Analysis with `R`
library(dplyr) # A Grammar of Data Manipulation
library(ggplot2) # Create Elegant Data Visualisations Using the Grammar of Graphics
library(gplots) # Various R Programming Tools for Plotting Data
library(ggmosaic) # Mosaic Plots in the 'ggplot2' Framework
```

We will now reload the data set that will be explored further in this chapter. The data set can be found in the package {discrtr}. Whereas in Chap. 1, we read a text file (in csv format), the package also includes the same data as a data frame that can be loaded with the function `data()`.

```
# Load the data set
data("mc_commute_wide",
     package = "discrtr")
```

The `mc_commute_wide` data set contains information about commute trips to McMaster University, together with individual characteristics and attitudinal questions. This is a clean version of the data set used in the previous chapter.

You can access the documentation of the data set using the following code. It will appear in the Help tab.

```
?mc_commute_wide
```

2.6 Univariate Analysis of Categorical Variables

Before doing any meaningful analysis, it is essential to understand the variables in the data set. Univariate analysis consists of exploring each variable separately. This step is first helpful in detecting outliers, missing values or errors, and/or to exclude non-relevant variables. It also allows the analyst to begin developing an understanding of the distribution of the variables. When conducted in an appropriate manner, univariate analysis ensures the quality of the data that enters the model. The information is also informative for specifying the model.

The univariate analysis techniques consist mainly of descriptive statistics, and data can be visualized using a variety of plots, including barplots and treemaps (pie charts too, although they are controversial because interpretation of the size of pizza slices is more difficult than interpretation of the height of rectangles, like in bar plots).

Let us start by summarizing the whole data set. For each variable, we have some information about the range and distribution of the values of each variable. The `summary()` function provides descriptive statistics appropriate for each variable, depending on whether it is a categorical or a quantitative measurements. We will also change the labels of two categorical variables as follows, to facilitate presentation:

```
# Pipe table `mc_commute_wide`
mc_commute_wide <- mc_commute_wide %>%
  # Function `mutate()` creates new columns in the table; mutate
  # the table to convert variables `child` and `vehind` to factors
  # with more informative labels
  mutate(child = factor(child,
                        levels=c("Yes",
                                 "No"),
                        # Give the factor categories more descriptive labels
                        labels=c("Living with a child",
                                 "Not living with a child")),
         # Relabel `vehind` variable
         vehind = factor(vehind,
                         levels=c("No",
                                  "Yes"),
                         # Give the factor categories more descriptive labels
                         labels=c("No ind. vehicle access",
                                  "Ind. vehicle access")))
```

And now the summary of the data set

```
summary(mc_commute_wide)
```

```
      id               choice    available.Cycle available.Walk
Min.   :566872636   Cycle: 47   No :997         Length:1375
1st Qu.:567814164   Walk :711   Yes:378         Class :character
Median :568681701   HSR  :336                   Mode  :character
Mean   :570562590   Car  :281
3rd Qu.:574923360
Max.   :587675235

available.HSR available.Car   time.Cycle        time.Walk        time.HSR
No :  54      No :622       Min.   : 0.3106   Min.   : 1.00   Min.   :  1.00
Yes:1321      Yes:753       1st Qu.: 2.9141   1st Qu.:10.00   1st Qu.:  4.00
                            Median : 4.3711   Median :15.00   Median :  8.00
                            Mean   : 5.4278   Mean   :16.09   Mean   : 16.82
                            3rd Qu.: 6.2112   3rd Qu.:20.00   3rd Qu.: 20.00
                            Max.   :45.0000   Max.   :62.11   Max.   :120.00
                            NA’s   :997       NA’s   :513     NA’s   :54
   time.Car      access.Cycle  access.Walk   access.HSR      access.Car
Min.   : 1.00   Min.   :0     Min.   :0    Min.   : 0.00   Min.   :0
1st Qu.: 5.00   1st Qu.:0     1st Qu.:0    1st Qu.: 2.48   1st Qu.:0
Median :10.00   Median :0     Median :0    Median : 6.21   Median :0
Mean   :12.86   Mean   :0     Mean   :0    Mean   :11.06   Mean   :0
3rd Qu.:20.00   3rd Qu.:0     3rd Qu.:0    3rd Qu.:12.42   3rd Qu.:0
Max.   :72.00   Max.   :0     Max.   :0    Max.   :62.11   Max.   :0
NA’s   :622
  wait.Cycle    wait.Walk     wait.HSR        wait.Car transfer.Cycle
Min.   :0    Min.   :0   Min.   : 0.00   Min.   :0   Min.   :0
1st Qu.:0    1st Qu.:0   1st Qu.:10.23   1st Qu.:0   1st Qu.:0
Median :0    Median :0   Median :10.23   Median :0   Median :0
Mean   :0    Mean   :0   Mean   :10.25   Mean   :0   Mean   :0
3rd Qu.:0    3rd Qu.:0   3rd Qu.:10.23   3rd Qu.:0   3rd Qu.:0
Max.   :0    Max.   :0   Max.   :50.00   Max.   :0   Max.   :0

transfer.Walk  transfer.HSR      transfer.Car parking
Min.   :0     Min.   :     0   Min.   :0     No :1260
1st Qu.:0     1st Qu.:     0   1st Qu.:0     Yes: 115
Median :0     Median :     0   Median :0
```

```
Mean   :0     Mean   :  3928   Mean   :0
3rd Qu.:0     3rd Qu.:     1   3rd Qu.:0
Max.   :0     Max.   :100000   Max.   :0

                   vehind       gender         age
No ind. vehicle access:1022   Woman:824   Min.   :17.00
Ind. vehicle access   : 353   Man  :551   1st Qu.:20.00
                                          Median :21.00
                                          Mean   :22.08
                                          3rd Qu.:23.00
                                          Max.   :60.00

                            shared                   family
No                             :516   No                :1034
Living in Shared Accommodations:859   Living with Family: 341




                    child      street_density  sidewalk_density
Living with a child    : 291   Min.   : 0.00   Min.   : 0.00
Not living with a child:1084   1st Qu.:10.36   1st Qu.:18.19
                               Median :14.29   Median :22.63
                               Mean   :13.27   Mean   :24.18
                               3rd Qu.:16.18   3rd Qu.:35.70
                               Max.   :25.22   Max.   :59.41

     LAT            LONG        PersonalVehComf_SD PersonalVehComf_D
Min.   :43.08   Min.   :-80.09   Min.   :0.000000   Min.   :0.00000
1st Qu.:43.25   1st Qu.:-79.92   1st Qu.:0.000000   1st Qu.:0.00000
Median :43.26   Median :-79.91   Median :0.000000   Median :0.00000
Mean   :43.25   Mean   :-79.90   Mean   :0.005091   Mean   :0.03055
3rd Qu.:43.26   3rd Qu.:-79.90   3rd Qu.:0.000000   3rd Qu.:0.00000
Max.   :43.28   Max.   :-79.64   Max.   :1.000000   Max.   :1.00000

PersonalVehComf_A PersonalVehComf_SA     Fun_SD           Fun_D
Min.   :0.0000    Min.   :0.0000     Min.   :0.00000   Min.   :0.0000
1st Qu.:0.0000    1st Qu.:0.0000     1st Qu.:0.00000   1st Qu.:0.0000
Median :1.0000    Median :0.0000     Median :0.00000   Median :0.0000
Mean   :0.6269    Mean   :0.1935     Mean   :0.06036   Mean   :0.2429
3rd Qu.:1.0000    3rd Qu.:0.0000     3rd Qu.:0.00000   3rd Qu.:0.0000
Max.   :1.0000    Max.   :1.0000     Max.   :1.00000   Max.   :1.0000

    Fun_A           Fun_SA        ActiveNeigh_SD    ActiveNeigh_D
Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
Median :0.0000   Median :0.00000   Median :0.00000   Median :0.0000
Mean   :0.2829   Mean   :0.04727   Mean   :0.02255   Mean   :0.1949
3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000
Max.   :1.0000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000

ActiveNeigh_A    ActiveNeigh_SA   UsefulTrans_SD    UsefulTrans_D
Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000
1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000
Median :0.0000   Median :0.0000   Median :0.00000   Median :0.0000
Mean   :0.3745   Mean   :0.1069   Mean   :0.04509   Mean   :0.2109
3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.0000
Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :1.0000

UsefulTrans_A    UsefulTrans_SA      BusComf_SD        BusComf_D
Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
```

```
Median :0.0000   Median :0.00000   Median :0.00000   Median :0.0000
Mean   :0.3047   Mean   :0.06327   Mean   :0.04509   Mean   :0.2022
3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000
Max.   :1.0000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000

  BusComf_A        BusComf_SA       TravelAlone_SD    TravelAlone_D
Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
Median :0.0000   Median :0.00000   Median :0.00000   Median :0.0000
Mean   :0.4051   Mean   :0.02909   Mean   :0.04364   Mean   :0.2516
3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.0000
Max.   :1.0000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000

TravelAlone_A    TravelAlone_SA     Shelters_SD        Shelters_D
Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
Median :0.0000   Median :0.00000   Median :0.00000   Median :0.0000
Mean   :0.2713   Mean   :0.04291   Mean   :0.04436   Mean   :0.2291
3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000
Max.   :1.0000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000

  Shelters_A      Shelters_SA       Community_SD      Community_D
Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
Median :0.0000   Median :0.00000   Median :0.00000   Median :0.0000
Mean   :0.3578   Mean   :0.02764   Mean   :0.06255   Mean   :0.2495
3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000
Max.   :1.0000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000

 Community_A      Community_SA     personal_veh_comfortable getting_there_fun
Min.   :0.0000   Min.   :0.00000   SD:  7                   SD: 83
1st Qu.:0.0000   1st Qu.:0.00000   D : 42                   D :334
Median :0.0000   Median :0.00000   N :198                   N :504
Mean   :0.3033   Mean   :0.04291   A :862                   A :389
3rd Qu.:1.0000   3rd Qu.:0.00000   SA:266                   SA: 65
Max.   :1.0000   Max.   :1.00000

like_active_neighborhood commute_useful_transition buses_comfortable
SD: 31                   SD: 62                    SD: 62
D :268                   D :290                    D :278
N :414                   N :517                    N :438
A :515                   A :419                    A :557
SA:147                   SA: 87                    SA: 40


prefer_travel_alone shelter_good_quality sense_community     numna
SD: 60              SD: 61               SD: 86          Min.   :2.00
D :346              D :315               D :343          1st Qu.:2.00
N :537              N :469               N :470          Median :2.00
A :373              A :492               A :417          Mean   :2.41
SA: 59              SA: 38               SA: 59          3rd Qu.:3.00
                                                         Max.   :4.00
```

The clean data set provided includes different classes of variables. By looking at the summary statistics, what can you say about the `choice` variable? The `age` variable? The `Shelters_SD` variable?

For continuous variables, the summary statistics provide meaningful information and can be combined with a graph of the distribution (smoothed histogram), as done in Chap. 1.

For discrete variables, some caution is required. Let us examine the following variables: `Shelters_SD`, `Shelters_D`, `Shelters_A`, and `Shelters_SA`. These variables are dummy variables indicating the response to the statement: *Shelters and other public transportation facilities that I commonly use are of good quality*. `Shelters_SD`, for example, is equal to 1 if the respondent answered *Strongly disagree* and equal to 0 otherwise. If we look at the summary statistics below, we can see that the`Shelters_SD` variable is coded as a continuous variable.

```
summary(mc_commute_wide$Shelters_SD)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
0.00000 0.00000 0.00000 0.04436 0.00000 1.00000 
```

The information provided here with the summary statistics is limited. The mean informs us about the proportion of responses that are set to 1. The min and max values allow us to see if there are any errors (values above 1 or below 0, or continuous values between 0 and 1).

To have a meaningful representation of the data, we are interested in the combined responses to all the agreement variables (from *Strongly disagree* to *Strongly agree*) associated with the statement. We will, therefore, create a new single variable that combines all responses to this question into a single variable. We first create a new variable `Shelters` with values that depend on the various columns that collectively contain the response to the one question about shelters

```
# Pipe data frame `mc_mode_wide` to next function
mc_commute_wide <- mc_commute_wide %>%
  # Use mutate to create a new variable
  # Function `case_when()` is a vectorized form of if-else statements
  mutate(Shelters = case_when(Shelters_SD == 1 ~ -2,
                              Shelters_D == 1 ~ -1,
                              Shelters_A == 1 ~ 1,
                              Shelters_SA == 1 ~ 2,
                              TRUE ~ 0))
```

You will notice that there is no variable corresponding to the “Neutral” state in the data set. Therefore, if a respondent has a value of 0 for all four agreement statements, then we know that they must have answered *Neutral*, which is set to 0 here. Now all the information is in a single variable, ranging from -2 to 2. This variable has a mean of 0.0953; what does this number mean?

We recoded all opinions about shelters in a single variable, but it is still numerical, when in fact we know that the variable is categorical. For this reason, we will convert it into a *factor*, which is how `R` deals with categorical variables:

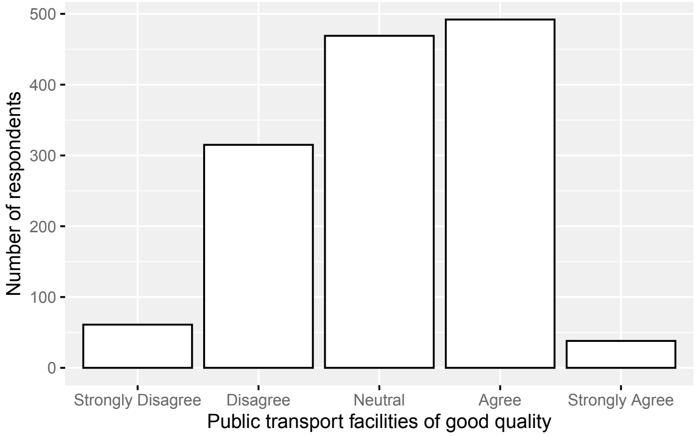
```
mc_commute_wide <- mc_commute_wide %>%
  # Use `mutate()` to modify the content of an existing variable
  mutate(Shelters = factor(Shelters,
                           levels = c(-2, -1, 0, 1, 2),
                           labels = c("Strongly Disagree",
                                      "Disagree",
                                      "Neutral",
                                      "Agree",
                                      "Strongly Agree"),
                           # The factor is an ordinal variable
                           ordered = TRUE))

summary(mc_commute_wide$Shelters)
```

```
  Strongly Disagree          Disagree           Neutral             Agree 
                 61               315               469               492 
     Strongly Agree 
                 38 
```

The summary is simply a table of frequencies. What can we now tell from the summary statistics? To avoid confusion between continuous and categorical variables, a best practice consists in coding categorical variables with labels rather than numbers. To do so, we assign character strings instead of numerical values. In addition, the variable for opinions about shelters is in Likert scale, which means it is not only categorical but ordinal, which we indicate by means of `ordered = TRUE` when we converted it into a factor.

Now that we have our factor variable coded with character strings, we can proceed to visualize the data. A simple way to see the distribution of a categorical variable is a *barplot*. Here, we do so using the {ggplot2} package introduced in Chap. 1.

```
# Pipe table `mc_Commute_wide to `ggplot()`
mc_commute_wide %>%
  # Create a ggplot object with the table that was piped
  # and map the variable `Shelters` to the x-axis
  ggplot(aes(x = Shelters)) +
  # Add a geometric object of type bar; we do not need
  # to specify the y-axis because the height of the bar
  # will be the statistic for the corresponding categorical
  # outcome
  geom_bar(color = "black",
           fill = "white") +
  # The function `labs()` adds labels to part of the plot, for instance the x and y axes
  labs(x = "Public transport facilities of good quality",
       y = "Number of respondents")
```

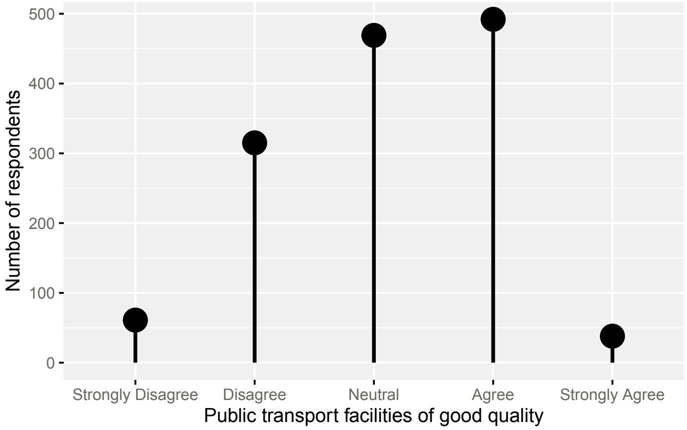


It is now much more intuitive to understand the distribution of the responses, most responses being in the center and fewer at the two extremes.

If you take a look at the `shelter_good_quality` variable that was already provided in the data set, you will see that it corresponds to the variable we just created! In fact, ordered factors have already been created in the data set for the agreement questions. Specifying ordered factors is of utmost importance when specifying and estimating a model with ordinal variables.

A variation of the bar chart is the *lollipop plot*. Lollipop plots use points for the ends of segments whose height depends on the frequency of a response. In other words, the position of the point on the y axis depends on the number of responses. Here, we group the table by the responses of `Shelters` and then summarize the number of responses by category:

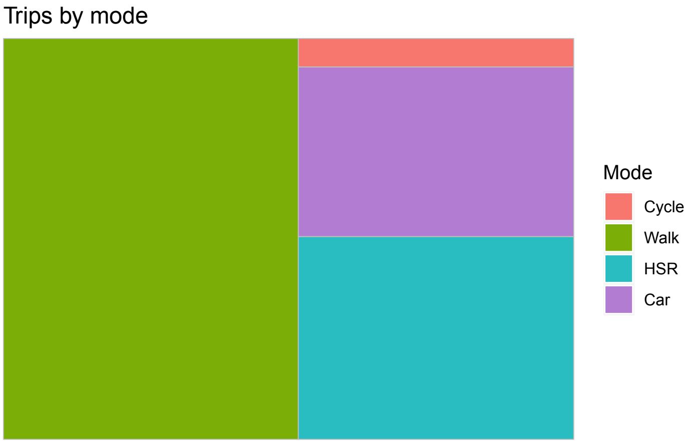
```
# Pipe table `mc_commute_wide`
mc_commute_wide %>%
  # Use `group_by()` to group the table by the values
  # of variable `Shelters`
  group_by(Shelters) %>%
  # Summarize: calculate the number n() of cases by
  # category of `Shelters`
  summarize(n = n()) %>%
  # Pipe the result to `ggplot()`; map `Shelters` to the x-axis
  # and map the number of cases to y; to create segments map
  # the end of the segment to y = 0 and keep it constant on x,
  # this will create vertical line
  ggplot(aes(x = Shelters,
             xend = Shelters,
             y = n,
             yend = 0)) +
  # Add geometric featues of type point
  geom_point(color = "black",
             fill = "white",
             size = 6) +
  # Add geometric features of type segment (line segments)
  geom_segment(size = 1) +
  # Label the axes
  labs(x = "Public transport facilities of good quality",
       y = "Number of respondents")
```



Lollipop plots use less “ink” to convey similar information as bar charts. This relates to what Tufte calls the https://infovis-wiki.net/wiki/Data-Ink_Ratio *data-ink ratio*: ink should be used to provide data-information (the default light gray background in {ggplot2} is a lot of ink to show little information, essentially the grid lines in negative space).

Another way to visualize univariate distributions of categorical variables which is particularly useful when there are many levels of a response is by means of *treemaps* implemented in `R` in the package {treemapify}. In the example below, we first summarize the number of trips by mode (grouping first by choice and then calculating the number of trips in each category with `n()`):

```
library(treemapify)

# Pipe table
mc_commute_wide %>%
  # Group table based on `choice`
  group_by(choice) %>%
  # Count the number of cases by `choice` and pipe to `ggplot()`
  summarize(n = n()) %>%
  # Map the color of the rectangles to the variable `choice` and
  # their area to the number of cases
  ggplot(aes(fill = choice,
             area = n)) +
  # Layer geometric object of type treemap
  geom_treemap() +
  # Add labels
  labs(title = "Trips by mode",
       fill="Mode")
```



Treemaps also show the frequency relative to the total, something that is more difficult to grasp from bar and lollipop plots.

Another set of variables that deserves our attention are the `shared` and `family` variables. These two variables correspond to the living arrangement: whether an individual was living in a shared accommodation or in a family home. If none of these two options are true, it means that the individual was living alone (which we will refer here to as *solo*). As done for the `Shelters` variables, we will create a new single variable combining these answers:

```
mc_commute_wide <- mc_commute_wide %>%
  # Use `mutate()` to convert variable `housing` to a factor
  mutate(housing = case_when(shared != "No" ~ "shared",
                             family != "No" ~ "family",
                             TRUE ~ "solo"),
         housing = factor(housing))

summary(mc_commute_wide$housing)
```

```
 family shared   solo
    341    859    175
```

We will use this variable later in this chapter.

2.7 Bivariate Analysis

After having examined each variable separately, we are now ready to explore the relationships between our variables. As such, before specifying and estimating a model, it is important to understand how the variables relate to each other and to formulate hypotheses.

Bivariate analysis consists in visualizing the relationship between two variables. The choice of a tool depends on whether the variables are both quantitative, both categorical, or one of each. The boxplot presented in Chap. 1 is an example of a bivariate visualization for one categorical and one quantitative variable. Another technique that we will see in this chapter is the violin plot.

As there are often more than one discrete variable involved in DCA, being able to represent and visualize the relationship between two discrete variables is essential. There is an array of existing techniques that can be used to do so, and new ones emerge regularly. In this chapter, we only present a few, but we encourage the reader to not limit themselves to the techniques presented here.

2.7.1 *Exploring the Relationship Between a Categorical and a Quantitative Variables*

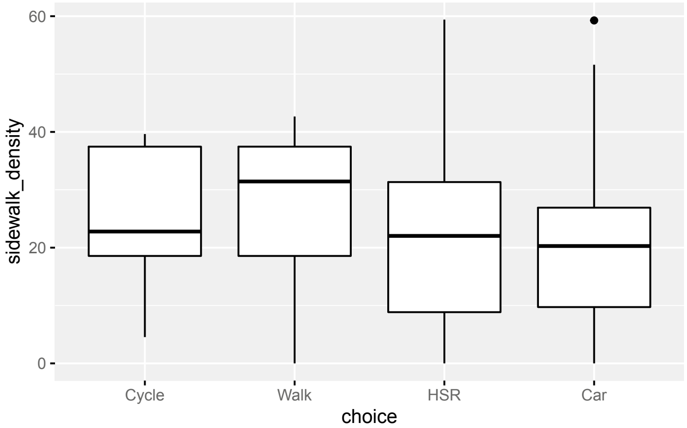
Before doing any bivariate analysis, it is important to inspect the summary statistics of the two variables of interest. Let us start by examining the relationship between mode choice and sidewalk density, as done in the previous chapter.

```
# Pipe table `mc_commute_wide` to `select()`
mc_commute_wide %>%
  # Select variables `choice` and `sidewalk_density`
  select(choice,
         sidewalk_density) %>%
  summary()
```

```
   choice     sidewalk_density
 Cycle: 47    Min.   : 0.00
 Walk :711    1st Qu.:18.19
 HSR  :336    Median :22.63
 Car  :281    Mean   :24.18
              3rd Qu.:35.70
              Max.   :59.41
```

The *boxplot* presented in the preceding chapter is a simple way of describing the relationship between one categorical and one continuous variable

```
mc_commute_wide %>%
  # Map `choice` to the x-axis and `sidewalk_density` to the y-axis
  ggplot(aes(x = choice,
             y = sidewalk_density)) +
  # Boxplots are useful for visualizing the relationship between
  # one categorical and one quantitative variable
  geom_boxplot()
```

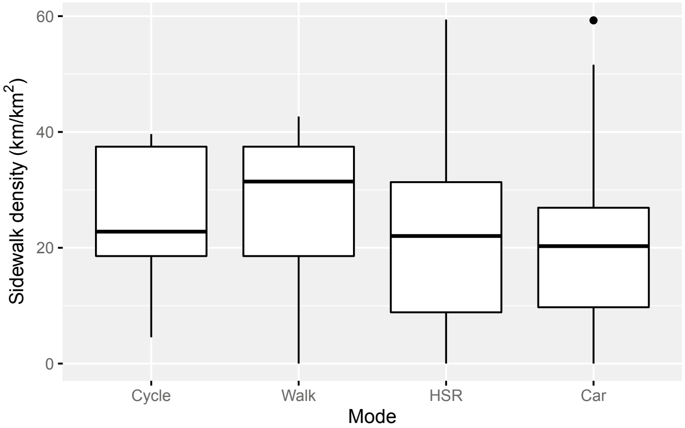


What do you learn from this figure? What are the variables? How is sidewalk density measured?

It is important to provide the reader with the basic information directly in the figure. This way, the reader can directly interpret the results, without having to go back to the definition of the variables for example.

We will, therefore, clarify the titles and units of the labels

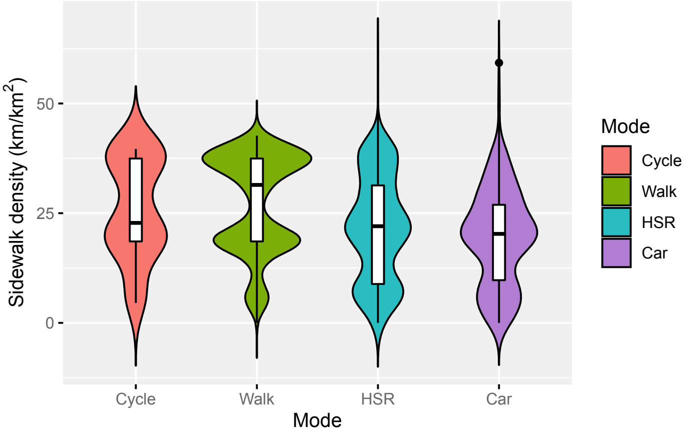
```
mc_commute_wide %>%
  ggplot(aes(x = choice,
             y = sidewalk_density)) +
  geom_boxplot() +
  # Label the axes
  labs(x="Mode",
       # The expression function allows us to include superscripts
       # and subscripts in labels and titles
       y = expression("Sidewalk density (km/km"^2*")"))
```



Another way to plot the bivariate relation between a quantitative and a categorical variable is by means of the *violin plot*. The boxplot seen above shows the quartile values: median is the thick line in the box; the box covers 50% of the observations (between the first and third quartile), and the stems extend to 1.5 of the interquartile range; any points beyond the ends of the stems are “outliers”. The boxplot obscure the distribution of observations beyond these statistics.

A violin plot is a way to reveal this information, as shown in the figure below (created with the `geom_violin()` function), which illustrates a more detailed distribution of the values for each category

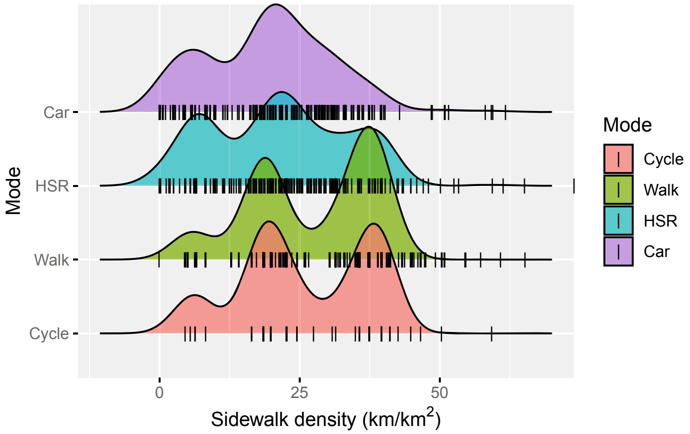
```
mc_commute_wide %>%
  ggplot(aes(x = choice,
             y = sidewalk_density,
             # Map the color of the polygons to `choice`
             fill = choice)) +
  # Add a geometric object of type violin
  geom_violin(trim = FALSE) +
  # Add geometric object of type boxplot
  geom_boxplot(width = 0.1,
               fill = "white") +
  labs(x="Mode",
       y = expression("Sidewalk density (km/km"^2*")"),
       # Add a label for the fill
       fill = "Mode")
```



Comparing the violin plot with the boxplot, we observe that some pieces of information were not visible in the boxplot. For example, if you look at the distribution of sidewalk density for respondents who walked, what can you observe? What could be the reason for this?

The shape of the violin results from duplicating the distribution along the symmetry axis, which means more ink for the same amount of information. An alternative are ridge plots, implemented in `R` in package {ggridges}. The following plot is of the same variables but each distribution is plotted only once. The specific data points from which the distributions are generated are indicated on the plot

```
library(ggridges)

mc_commute_wide %>%
  ggplot(aes(x = sidewalk_density,
             y = choice,
             # Map the color of the polygons to `choice`
             fill = choice)) +
  # Add geometric object of type ridges with jittered points
  geom_density_ridges(jittered_points = TRUE,
                      bandwidth = 3.5,
                      position = position_points_jitter(width = 0.05,
                                                        height = 0),
                      point_shape = '|',
                      point_size = 3,
                      point_alpha = 1,
                      alpha = 0.7) +
  labs(y="Mode",
       x = expression("Sidewalk density (km/km"^2*")"),
       # Add a label for the fill
       fill = "Mode")
```



Some techniques allow us to extract more information than others. Yet, this is not always useful, it always depend on the data at hand and on the relationships you want to examine for model specification, estimation, results, and interpretation.

2.7.2 Exploring the Relationship Between Two Categorical Variables

The previous examples explored the relationship between a categorical variable and a quantitative variable. Yet, discrete choice analysis often involves several categorical variables.

Perhaps the first step to explore two categorical variables is to create a contingency table. This table presents the joint frequency distribution or the cross-classification of frequencies for a set of two variables. The contingency table, or frequency distributions, can then be explored visually through a diversity of charts and plots.

As mentioned previously, there is a multitude of functions that exist in `R` to visualize data. Below are a few examples.

The *balloon plot* is a direct representation of the joint frequency, where the size of the circle is proportional to the joint frequency. Also, the marginal frequencies of each variables are illustrated via bars on the top and side of the table.

To illustrate the balloon plot, we will now examine the variable `child` in relation to the mode used to commute to McMaster. The `child` variable indicates the presence of dependent minors in household. First, let us examine the new variable we are interested in

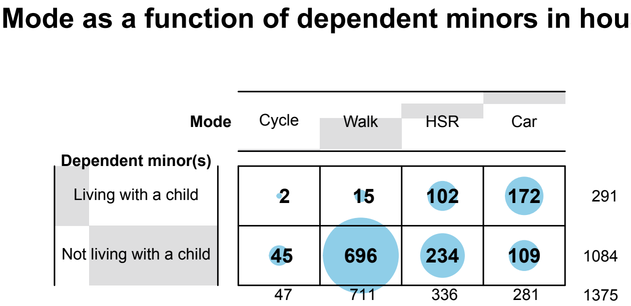
```r
summary(mc_commute_wide$child)
```

```
    Living with a child Not living with a child 
                    291                    1084 
```

We see that the majority of the respondents do not live with a child. Is this reasonable in light of the data set we have?

The relationship between living with a child and mode choice is illustrated below using a balloon plot. Here, we use a simple function included in the `gplots` package. A contingency table is first created with the function `table` and the `balloonplot` function is then used. You can examine the object `tableau` to see the contingency table.

```r
# Create a table with the two variables of interest
tableau <- table(mc_commute_wide$choice,
                 mc_commute_wide$child)

balloonplot(as.table(tableau),
            # The parameters below control the aspect of the table
            # Labels
            xlab = "Mode",
            ylab = "Dependent minor(s)",
            # Adjust maximum dot size
            dotsize = 3/max(strwidth(19),
                            strheight(19)),
            # Symbol used for the dots
            dotcolor = "skyblue",
            text.size = 0.65,
            # Title of plot
            main = "Mode as a function of dependent minors in household",
            # Display the values in the cells
            label = TRUE,
            label.size = 0.80,
            # Scale balloons by volume (or diameter)
            scale.method = c("volume"),
            # Scale balloons relative to zero
            scale.range = c("absolute"),
            # Space for column/row labels
            colmar = 1,
            rowmar = 2,
            # Display zeros if present
            show.zeros = TRUE,
            # Display row and column sums
            show.margins = TRUE,
            # Display cumulative margins as cascade plots
            cum.margins = TRUE)
```

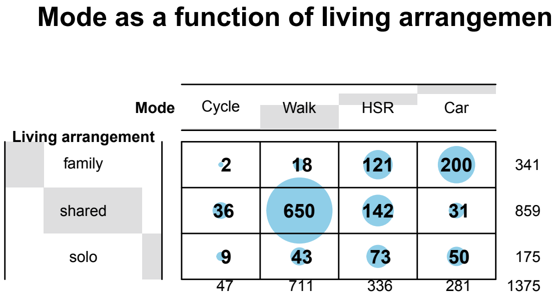


What can we interpret from this figure?

The size of the circles allows for a quick comparison between the frequency distributions. Looking at the figure, it is possible to conclude that almost all individuals who live with a child commute by public transport or by car, which is consistent with our expectations. We can also observe that almost all individuals who walk do not live with a child. In addition, looking at the gray bars, we can see that almost half the respondents walked and that a small share of the respondents live with a child.

Below is an another example of a balloon plot, this one looking at the relationship between living arrangement (`housing`, discussed in the section above) and mode choice.

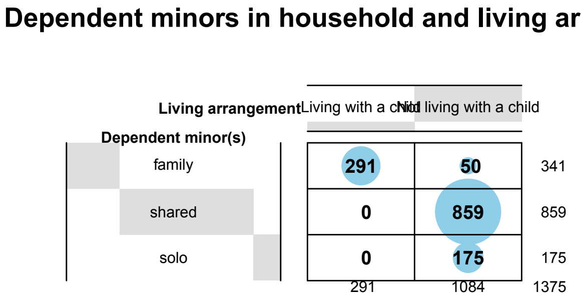
```
tableau <- table(mc_commute_wide$choice,
                 mc_commute_wide$housing)

balloonplot(as.table(tableau),
            # The parameters below control the aspect of the table
            # Labels
            xlab = "Mode",
            ylab = "Living arrangement",
            # Adjust maximum dot size
            dotsize = 3/max(strwidth(19),
                            strheight(19)),
            # Symbol used for the dots
            dotcolor = "skyblue",
            text.size = 0.65,
            # Title of plot
            main = "Mode as a function of living arrangement",
            # Display the values in the cells
            label = TRUE,
            label.size = 0.80,
            # Scale balloons by volume (or diameter)
            scale.method = c("volume"),
            # Scale balloons relative to zero
            scale.range = c("absolute"),
            # Space for column/row labels
            colmar = 1,
            rowmar = 2,
            # Display zeros if present
            show.zeros = TRUE,
            # Display row and column sums
            show.margins = TRUE,
            # Display cumulative margins as cascade plots
            cum.margins = TRUE)
```



How do these results relate to the previous balloon plot? Is there a relationship between the living arrangement and living with a child?

We notice that most people living in a family home commute by public transport and car. This could be related to the fact that individuals living in a family home are more likely to live with a child. We can directly examine the relationship between these two variables

```
tableau <- table(mc_commute_wide$child,
                 mc_commute_wide$housing)

balloonplot(as.table(tableau),
            # The parameters below control the aspect of the table
            # Labels
            xlab = "Living arrangement",
            ylab = "Dependent minor(s)",
            # Adjust maximum dot size
            dotsize = 3/max(strwidth(19),
                            strheight(19)),
            # Symbol used for the dots
            dotcolor = "skyblue",
            text.size = 0.65,
            # Title of plot
            main = "Dependent minors in household and living arrangement",
            # Display the values in the cells
            label = TRUE,
            label.size = 0.80,
            # Scale balloons by volume (or diameter)
            scale.method = c("volume"),
            # Scale balloons relative to zero
            scale.range = c("absolute"),
            # Space for column/row labels
            colmar = 1,
            rowmar = 2,
            # Display zeros if present
            show.zeros = TRUE,
            # Display row and column sums
            show.margins = TRUE,
            # Display cumulative margins as cascade plots
            cum.margins = TRUE)
```

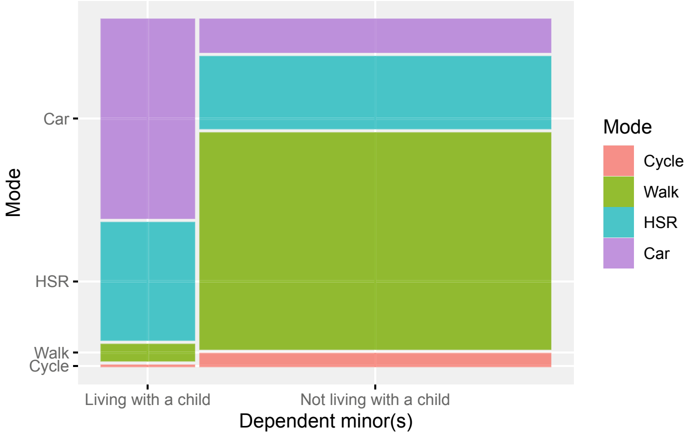

Dependent minors in household and living arrangement

The figure lends credence to our hypothesis. As expected, most individuals living in a family home live with a child. And unsurprinsgly, all individuals with a child live in a family home. This might explain the relationship between the living arrangement and mode choice for individuals residing in a family home.

Another way to visualize the contingency table is the *mosaic plot*. This plot is similar to a treemap, but it maps bivariate relations. Mosaic plots are implemented in {ggmosaic} package, and build on the two-way contingency table to display the conditional frequencies in the form of stacked bar charts, where the area is proportional to the observed joint frequency. The width of the bars reflects the conditional frequency of one variable and the height reflects the conditional frequency of the second variable.

Let us continue with the `child` variable (which represent the presence of a child in the household). We will create a mosaic plot to illustrate the relationship between mode choice and living with a child

```
mc_commute_wide %>%
  ggplot() +
  # Add geometric object of type mosaic
  # Map the interaction between `choice` and `child` to the x-axis
  geom_mosaic(aes(x = product(choice,
                              child),
                  fill = choice)) +
  # Add labels
  labs(x = "Dependent minor(s)",
       y = "Mode",
       fill = "Mode")
```

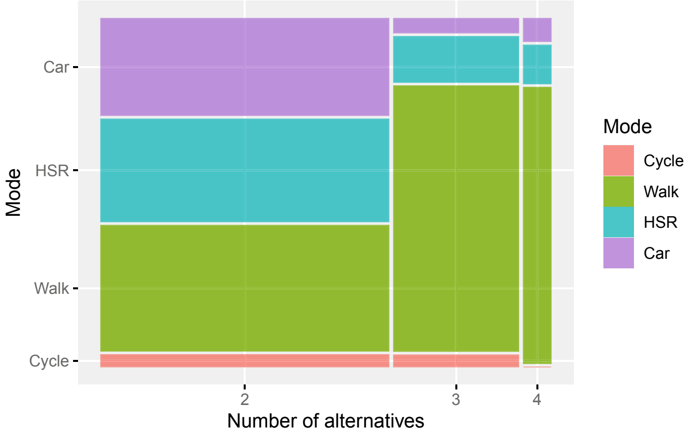


What does the figure tell us?

In comparison with the balloon plot, the mosaic plot provides a more intuitive representation of the conditional frequency distribution (width and height of bars), yet with no numbers. It is clear from the figure that for individuals living with a child, car represents the largest frequency, followed by public transport. As for individuals not living with a child, the majority walks, followed by public transport and car.

Below is another example of a mosaic plot comparing mode choice with the number of alternatives that are available to the respondents

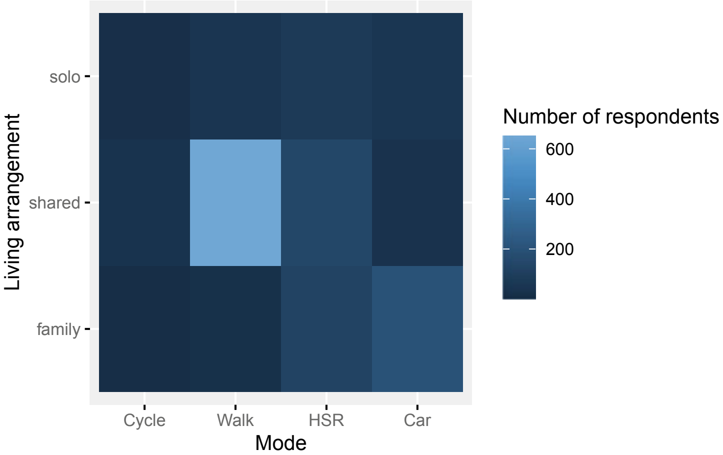
```
ggplot(data = mc_commute_wide) +
  # Add geometric object of type mosaic
  # Map the interaction between `choice` and `numna` to the x-axis
  geom_mosaic(aes(x = product(choice,
                              numna),
                  # Color rectangles based on `choice`
                  fill = choice)) +
  # Add labels
  labs(x = "Number of alternatives",
       y = "Mode",
       fill = "Mode")
```



Respondents who have only two alternatives have relatively similar distributions between walking, car and public transport, and they use car and public transport in a greater proportion compared to respondents with more alternatives. This could lead us to formulate the following hypothesis: the proportion of respondents for whom cycling and walking are not available forms of transportation is greater among respondents with only two alternatives. This could lead to a greater reliance on motorized modes for these respondents. We will get back to this later.

Another technique to visually present the contingency table is the *tile plot*. The tile plot assigns color intensities based on the frequency distribution. We will use the tile plot to illustrate the relationship between living arrangement and mode choice, as done previously with the mosaic plot

```
# Pipe table to next function
mc_commute_wide %>%
  # Group observations by `choice` and `housing`
  group_by(choice,
           housing) %>%
  # Calculate number of cases by combination of `choice` and `housing`
  summarize(n = n(),
            .groups = "drop") %>%
  ggplot(aes(x = choice,
             y = housing)) +
  # Add geometric object of type tile, map the color
  # of tiles to `n`, the number of cases
  geom_tile(aes(fill = n)) +
  # Add labels
  labs(x = "Mode",
       y = "Living arrangement",
       fill = "Number of respondents")
```



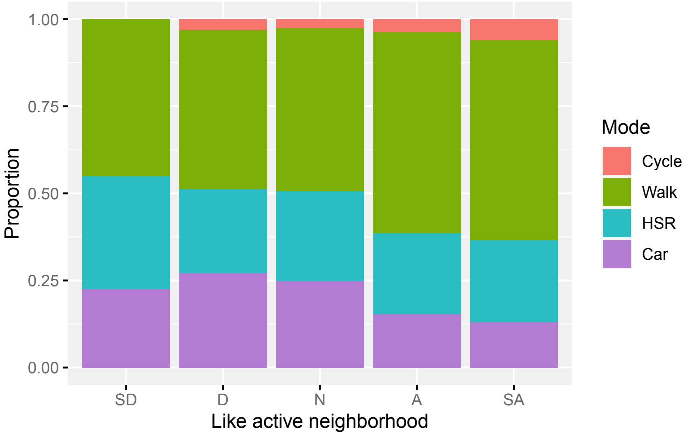
From this figure, the predominance of the individuals who walk in shared housing stands out even more clearly.

As mentioned in the introduction of this chapter, there is no strict rules for EDA. The choice of the visualization technique depends on the data at hand and on the patterns we are examining.

Another relevant technique to explore how the distribution of one variable varies across another variable is through the use of *stacked bar graphs*. While the techniques presented above display the joint frequency distributions, stacked bar graphs focus specifically on the distribution of one of the two variables. Bar graphs are especially relevant to uncover relationships associated with a categorical variable comprising multiple categories, including ordinal variables.

Let us take a look at one of the statement variables. We will examine here the agreement with the active neighbhorhood statement: *I like to live in a neighborhood where there's a lot going on* in relation to mode choice

```r
mc_commute_wide %>%
  ggplot(aes(x = like_active_neighborhood,
                      fill = choice)) +
  geom_bar(position = "fill") +
  labs(y = "Proportion",
       x = "Like active neighborhood",
       fill="Mode")
```



This figure displays a clear trend: the stronger the agreement with the statement, the greater the proportion of individuals who walk or cycle. While the figure does not provide information on the distribution of the responses to the agreement statement, it illustrates how the share of travel by mode varies with preferences for certain types of neighborhoods.

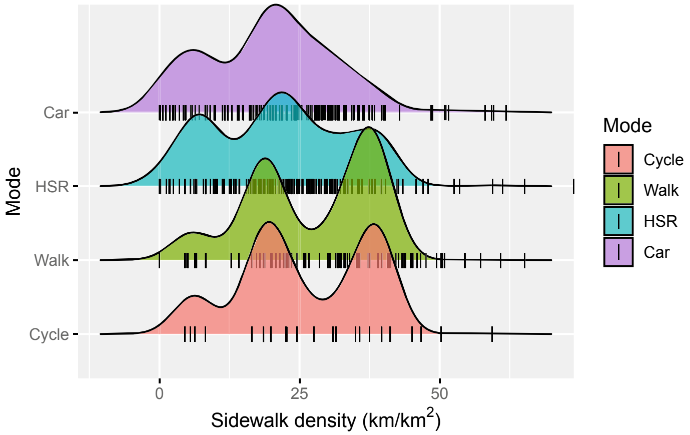
2.8 Multivariate Analysis

Up to this point, the techniques presented focused on two variables at a time. When conducting discrete choice analysis, modelers are typically interested in the joint distributions of multiple variables. For this reason, it is relevant to explore the relationship between more than two variables at the same time.

To this end, it helps to understand that every aspect of a plot, or *aesthetic* in the terminology of {ggplot2} drawing from the grammar of graphics (Wilkinson 2013), can be mapped to an underlying data dimension. Aesthetics include the x and y positions on a plot, the size of a geometric object (or `geom_` of which we have seen several already), the color of its boundaries or interior, its angle, transparency, and so on and so forth.

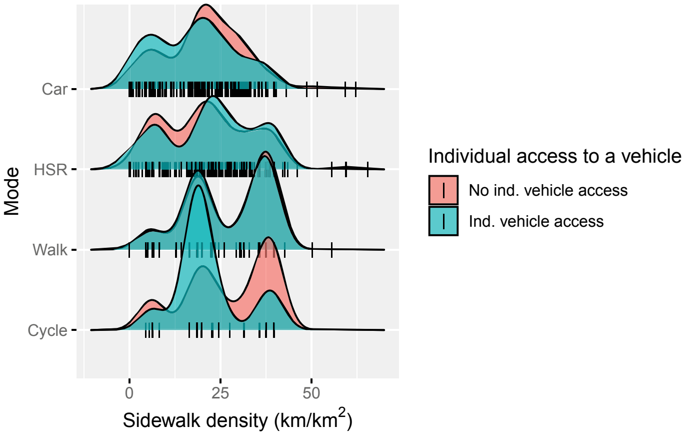
In some cases, mapping aesthetics to data values serves to reinforce an aspect of a visualization. If we revisit the ridge plot, we see that the colors of the ridges are repeats of the mode chosen, which was already mapped to the y axis

```
mc_commute_wide %>%
  ggplot(aes(x = sidewalk_density,
             # choice is mapped to the y axis
             y = choice,
             # choice is also mapped to the fill color!
             fill = choice)) +
  # Add geometric object of type density ridges
  geom_density_ridges(jittered_points = TRUE,
                      bandwidth = 3.5,
                      position = position_points_jitter(width = 0.05,
                                                        height = 0),
                      point_shape = '|',
                      point_size = 3,
                      point_alpha = 1,
                      alpha = 0.7)+
  # Add labels
  labs(y="Mode",
       x = expression("Sidewalk density (km/km"^2*")"),
       fill = "Mode")
```



Or we could have mapped the fill color to a *different* data dimension, like individual access to a car (to produce a multivariate visualization)

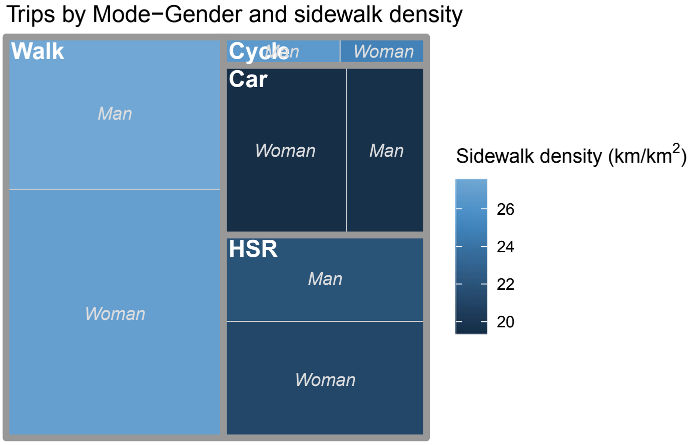
```
ggplot(data = mc_commute_wide,
       aes(x = sidewalk_density,
           y = choice,
           # By mapping the fill color to `vehind`
           # we introduce an additional data dimension to the plot
           fill = vehind)) +
  geom_density_ridges(jittered_points = TRUE,
                      bandwidth = 3.5,
                      position = position_points_jitter(width = 0.05,
                                                        height = 0),
                      point_shape = '|',
                      point_size = 3,
                      point_alpha = 1,
                      alpha = 0.7) +
  labs(y="Mode",
       x = expression("Sidewalk density (km/km"^2*")"),
       fill = "Individual access to a vehicle")
```



This plot allows us to examine three variables simultaneously. It shows that the distribution of mode by sidewalk density is pretty similar for people irrespective of whether they have individual access to a vehicle. There seems to be a difference for those who cycle, but we should not read too much into this, because as the distribution of points shows for this ridge, the number of people who cycle is relatively small so we can expect higher variance.

Before we used a treemap to explore the univariate distribution of travel by mode; in the following example, we map three data items (choice, gender, and the mean of sidewalk_density by choice-gender) to a single treemap

```
ggplot(data = mc_commute_wide %>%
         # Group by choice and gender
         group_by(choice,
                  gender) %>%
         # Summarize to obtain the number of responses by choice-gender combination
         # and the mean of sidewalk density for each group
         summarize(n = n(),
                   sidewalk_density = mean(sidewalk_density),
                   .groups = "drop"),
       # Map the area of the rectangles to the number of responses, the fill color
       # to mean sidewalk density, and group rectangles by choice
       aes(area = n,
           fill = sidewalk_density,
           label = gender,
           subgroup = choice)) +
  # Create main treemap
  geom_treemap()  +
  # Plot borders of subgroups
  geom_treemap_subgroup_border(size = 5)+
  # Add labels
  geom_treemap_subgroup_text(fontface = "bold",
                             colour = "white",
                             place = "topleft",
                             size = 13,
                             grow = FALSE) +
  geom_treemap_text(fontface = "italic",
                    colour = "lightgray",
                    place = "centre",
                    size = 10,
                    grow = FALSE) +
  labs(title = "Trips by Mode-Gender and sidewalk density", fill = expression("Sidewalk density (km/km"^2*")"))
```

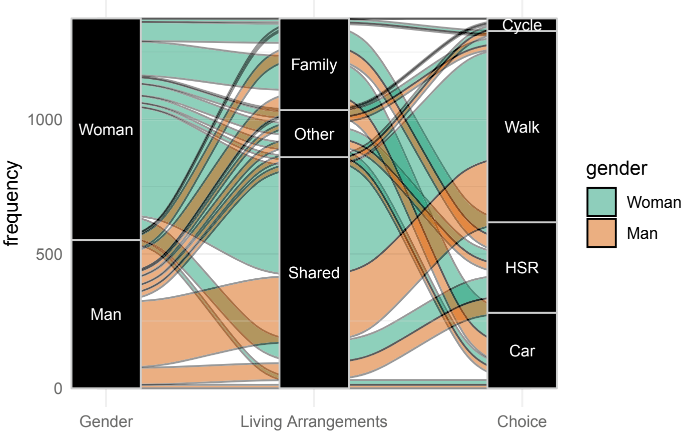


Lastly, here is the code for alluvial plot shown at the beginning of this chapter

```
# data preparation
mc_commute_alluvia <- mc_commute_wide %>%
  mutate(living_arrangments = case_when(shared == "Living in Shared Accommodations" ~ "Shared",
                                        family == "Living with Family" ~ "Family",
                                        TRUE ~ "Other")) %>%
  select(gender, living_arrangments, choice) %>%
  group_by(gender, living_arrangments, choice) %>%
  summarize(frequency = n(),
            .groups = "drop")
```

Render alluvial plot

```
# plot
mc_commute_alluvia %>%
  ggplot(aes(y = frequency,
             axis1 = gender,
             axis2 = living_arrangments,
             axis4 = choice)) +
  geom_alluvium(aes(fill = gender),
                width = 1/12,
                color = "black") +
  geom_stratum(width = 1/3,
               fill = "black",
               color = "grey") +
  geom_text(stat = "stratum",
            aes(label = after_stat(stratum)),
            color = "white",
            size = 3) +
  scale_x_discrete(limits = c("Gender",
                               "Living Arrangements",
                               "Choice"),
                   expand = c(.05, .05)) +
  scale_fill_brewer(type = "qual",
                    palette = "Dark2") +
  theme_minimal()
```



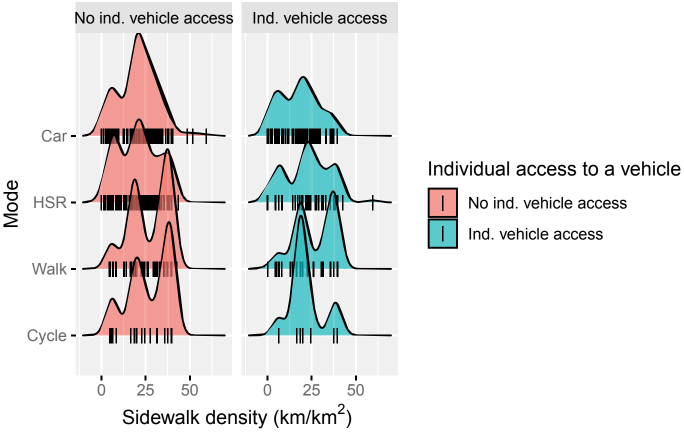
How many data dimensions are represented in this plot?

2.8.1 *Faceting*

Let us turn now to another technique useful to visualize multivariate distributions. Consider the ridge plot that introduced the variable for individual access to a vehicle. Some parts of the distributions are obscured due to overplotting. *Faceting* is a technique that divides a figure in multiple panels whereby each panel represents a segment of the data. In other words, faceting is a way to present the relationship between two variables, segmented across one or more additional variables. There is a simple function in {ggplot2} to do so, `facet_wrap`, which allows us to facet any type of plot available in the `ggplot` function.

Here we examine again the relationship between vehicle access, mode choice, and sidewalk density, using the faceting approach

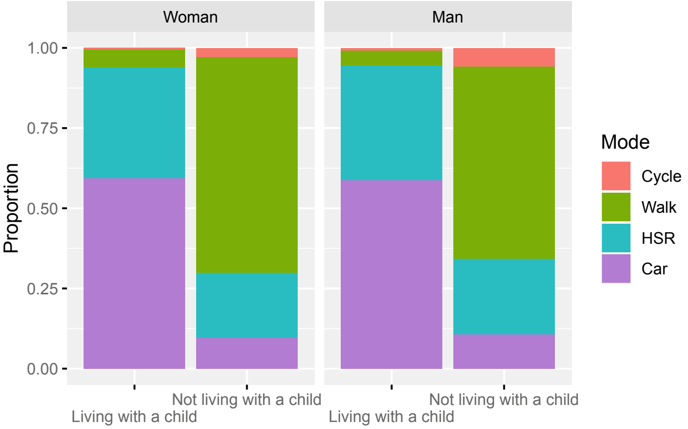
```
mc_commute_wide %>%
  ggplot(aes(x = sidewalk_density,
             y = choice,
             fill = vehind)) +
  geom_density_ridges(jittered_points = TRUE,
                      bandwidth = 3.5,
                      position = position_points_jitter(width = 0.05,
                                                         height = 0),
                      point_shape = '|',
                      point_size = 3,
                      point_alpha = 1,
                      alpha = 0.7) +
  labs(y="Mode",
       x = expression("Sidewalk density (km/km"^2*")"),
       fill = "Individual access to a vehicle")  +
  # `facet_wrap()` creates subplots after partitioning the data set
  # by the variable(s) specified, in this case `vehind`
  facet_wrap(~ vehind)
```



While in the previous figure, the ridge plots for individuals with and without access to a vehicle overlapped, and this figure shows them side by side. Overplotting is avoided, but the comparison of the two sets of distribution might not be as direct.

Here is another example of faceting with multiple variables. We will revisit the relationship between mode choice and living with a child, but now we create a bar plot and facet it based on the gender of the respondent

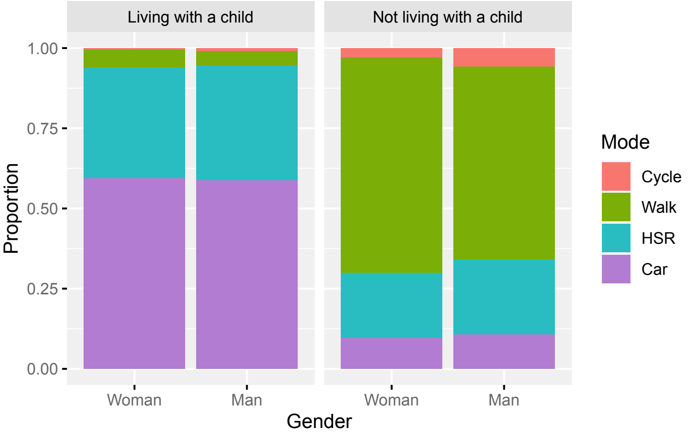
```
mc_commute_wide %>%
  ggplot(aes(x = child,
             fill = choice)) +
  geom_bar(position = "fill") +
  labs(y = "Proportion",
       x = "",
       fill="Mode") +
  # Facet the plots based on `gender`
  facet_wrap(~ gender)
```



At first sight, we do not see a large difference between the distributions for men and women.

Let us try visualizing the data differently. We will now plot the relationship between gender and mode choice, segmented based on whether the individual lives with a child. You will notice that the label names (that we created at the beginning of this chapter) appear at the top of each facet

```
mc_commute_wide %>%
  ggplot(aes(x = gender,
             fill = choice)) +
  geom_bar(position = "fill") +
  labs(y = "Proportion",
       x = "Gender",
       fill="Mode") +
  # Facet the plots based on `child`
  facet_wrap(~ child)
```

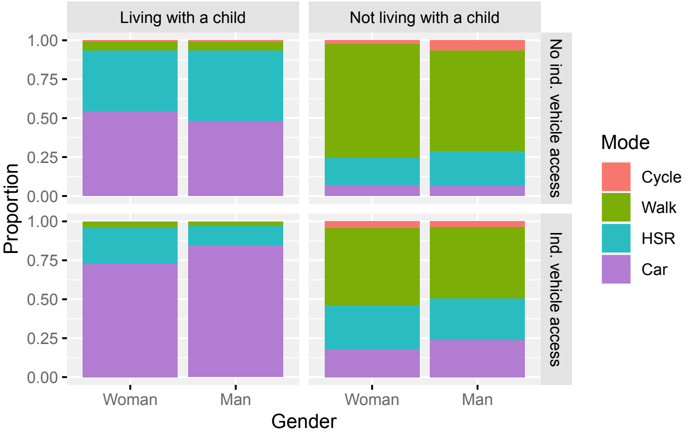


This visualization highlights a small difference in mode choice across gender for individuals not living with a child. We examine the same three variables as in the preceding figure, but presented differently, which makes some patterns easier to discern. Among individuals not living with a child, we observe that men cycle in greater proportion than women, while women walk more frequently. The differences between men and women are less pronounced for individuals living with at least one child.

This example illustrates the nature of exploratory data analysis. There is no right or wrong way to explore the data: just like modeling, some visualizations are more useful than others. We have to test different approaches to examine the patterns of association between variables, to explore which visualization technique reveals relevant patterns in a intuitive way.

Faceting can be applied to higher level segmentations, for instance, as a function of two discrete variables. To do so, we use the `facet_grid` function instead of the `facet_wrap` function as follows:

```
mc_commute_wide %>%
  ggplot(aes(x = gender,
             fill = choice)) +
  geom_bar(position = "fill") +
  labs(y = "Proportion",
       x = "Gender",
       fill="Mode") +
  # `facet_grid()` creates a "matrix" of subplots
  # with the variable on the left spread on the
  # x-axis and the one on the right on the y-axis
  facet_grid(vehind ~ child)
```



In this case, we faceted the data based on both living with a child and vehicle access (having individual access to a private car (i.e., car not shared with other household members)). Not surprisingly, the proportion of individuals commuting by car is greater among individuals with an individual access to a private car. For individuals living with a child, men drive in greater than women when they have individual access to a private car, whereas it is the opposite when the access to the vehicle is shared.

It is important to note, however, that segmenting the data reduces the number of observations in each panel. We examine below the number of observations in each panel

```
table(mc_commute_wide$vehind,
      mc_commute_wide$child)
```

```
                       Living with a child Not living with a child
No ind. vehicle access                 211                     811
Ind. vehicle access                     80                     273
```

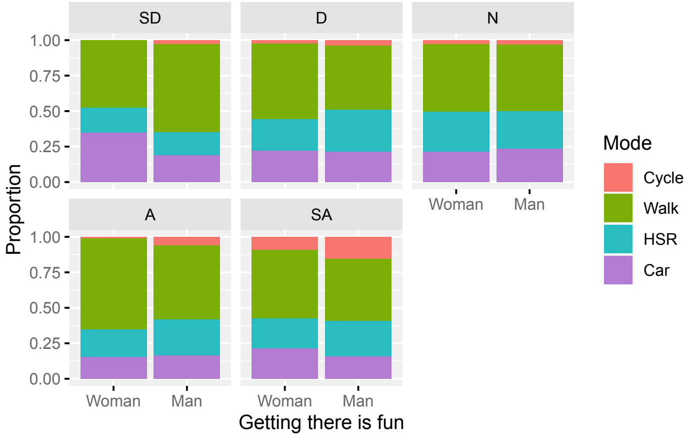
Knowing that there are only 80 individuals living with a child and with an individual access to a private car, we have to be careful when drawing conclusions, especially when we further segment by gender. Again, while there are several techniques that can be used to visualize the data, one has to be aware of the limitations and to select the most appropriate ones for the data.

2.8.2 More Examples of Faceting

Faceting is especially relevant when we have a discrete variable that contains several categories (levels). This could be income categories, geographic regions, Likert scale responses, etc.

We will illustrate this using the statement *Getting there is half the fun*. The figure below illustrates the relationship between gender and mode choice, segmented according to the response to the fun statement

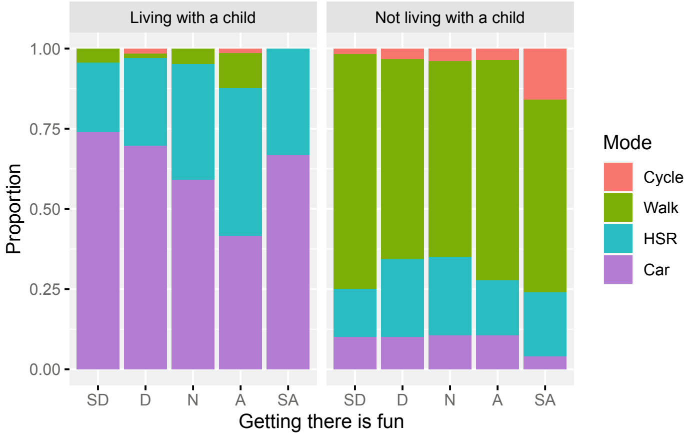
```
mc_commute_wide %>%
  ggplot() +
  (aes(x = gender,
       fill = choice)) +
  geom_bar(position = "fill") +
  labs(y = "Proportion",
       x = "Getting there is fun",
       fill = "Mode") +
  facet_wrap(~ getting_there_fun)
```



Some interesting patterns emerge. Among the respondents who are neutral, we see no differences between gender. However, among those that agree or strongly agree, we see that men cycle in a greater proportion than women. Such differences are also visible among respondents who disagree, but to a lower extent.

Let us now look again at the relationship between mode choice and the agreement with the fun statement, but as a function of the `child` variable. In this case, segmenting the data based on the child variable is more telling.

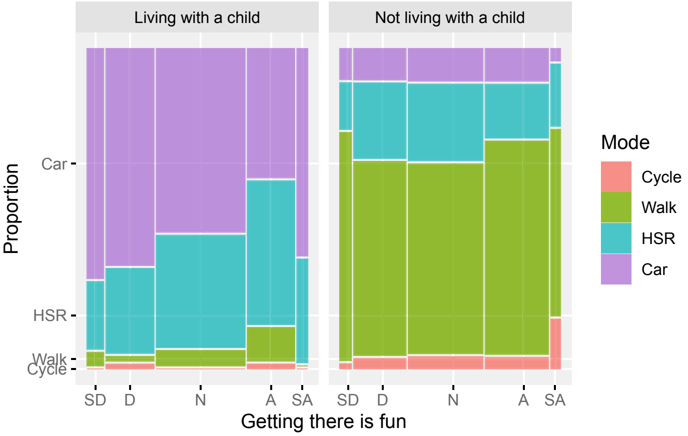
```
mc_commute_wide %>%
  ggplot(aes(x = getting_there_fun,
             fill = choice)) +
  geom_bar(position = "fill") +
  labs(y = "Proportion",
       x = "Getting there is fun",
       fill = "Mode") +
  facet_wrap(~ child)
```



A clear trend emerge for individuals not living with a child: cycling is associated with a greater agreement with the fun statement, especially for those who strongly agree. Among individuals living with a child, there is no clear trend for cycling, although this could be related to the small number of observations.

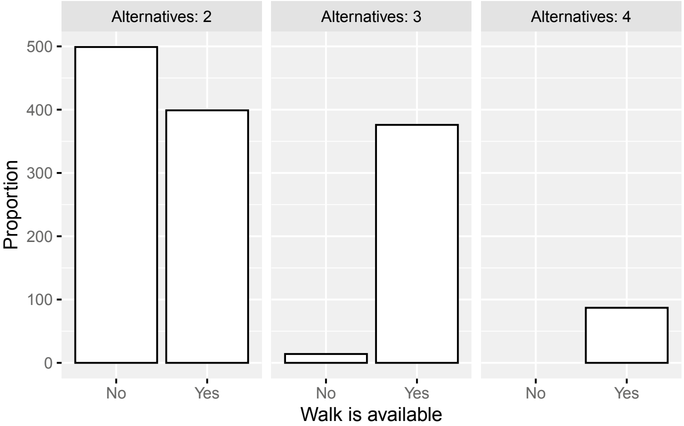
As mentioned above, faceting can be used with the various plots including in the {ggplot2} package. Here is an example using the same variables as above, but with a mosaic plot

```
mc_commute_wide %>%
  ggplot() +
  geom_mosaic(aes(x = product(choice,
                                getting_there_fun),
                  fill = choice)) +
  facet_wrap(~ child)+
  labs(y = "Proportion",
       x = "Getting there is fun",
       fill = "Mode")
```



Finally, let us come back to the number of alternatives that are available to each respondent. We hypothesized previously that a greater proportion of individuals with only two alternatives do not have walking as an alternative. Another approach to modifying the labels is to change the name of the variable and make it appear in the figure (`labeller=label_both`).

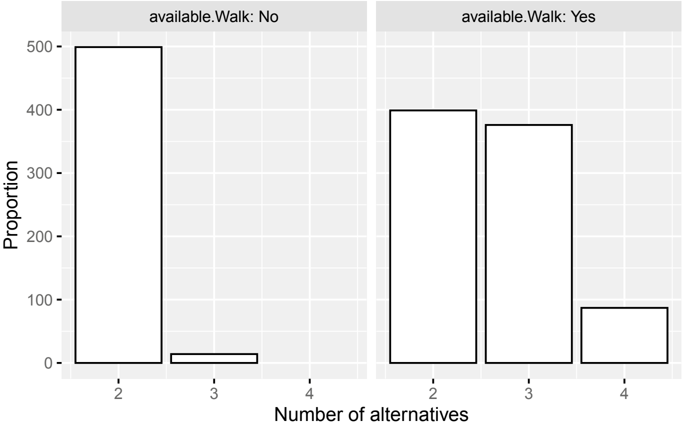
```
names(mc_commute_wide)[names(mc_commute_wide) == "numna"] <- "Alternatives" # Renaming variable

mc_commute_wide %>%
  ggplot(aes(x = available.Walk)) +
  labs(y = "Proportion",
       x = "Walk is available") +
  geom_bar(color="black",
           fill="white") +
  facet_wrap(~ Alternatives,
             labeller = label_both)
```



Very few individuals with 3 or 4 alternatives do not have walking as an alternative, as we initially thought. Conversely, the majority of respondents with 2 alternatives do not have walking as an alternative.

We can also segment the data based on the availability of the walking alternative.

```
mc_commute_wide %>%
  ggplot(aes(x = Alternatives)) +
  labs(y = "Proportion",
       x = "Number of alternatives")+
  geom_bar(color = "black",
           fill = "white") +
  facet_wrap(~ available.Walk,
             labeller = label_both)
```



It is clear from this figure that almost all individuals for whom walking is not an alternative only have two alternatives available to them.

2.9 Final Remarks

EDA is a creative process with no clear rules. In addition to exploring each variable individually, the aim of EDA is to uncover patterns of interest within the data. To do so, the analyst must test various relationships. Visualization techniques are especially helpful to provide an intuitive understanding of relationships. Several techniques can show the same patterns in a different way. It is up to the analyst to select the one that is the most appropriate to his/her work. It is important to remember: not just because we can do it we *should* do it. The objective is to illuminate, and too many dimensions in an exploratory exercise can obfuscate more than help.

2.10 Exercises

1. Why is EDA an important part of a modeling exercise?
2. How many aesthetic elements of visualization can you think of? List them and explain how they could be mapped to variables in different scales.
3. The data set explored in this chapter was in *wide* format: each row was a single respondent. However, discrete choice data is often found in *long* format: each row is a *choice situation*, and each individual now appears in the table as many times as they faced a choice situation. Would this change in any way how you approach data analysis?

Call data set `Car` from package `mlogit`. To do this, you need to first load the package. This is a data set with stated preferences about car types. This is done as follows:

```
library(mlogit)
```

```
Loading required package: dfidx
```

```
Attaching package: 'dfidx'
```

```
The following object is masked from 'package:stats':

    filter
```

```
data("Car")
```

Once you have loaded the data set, do the following:

4. How many variables are there in this data set and of which type (i.e., categorical/quantitative)?
5. Choose four relevant categorical variables from this data set and describe them using univariate, bivariate, and multivariate techniques. Discuss your results.
6. What ideas about individuals' choices regarding car do you develop from this EDA exercise?

References

Essen, Van, David, C., Anderson, Charles H., & Felleman, Daniel J. (1992). Information processing in the primate visual system: An integrated systems perspective. *Science, 255*(5043), 419–23.

Franconeri, S. L., Padilla, L. M., Shah, P., Zacks, J. M., & Hullman, J. (2021). The science of visual data communication: What works. *Psychological Science in the Public Interest, 22*(3), 61–110. https://doi.org/10.1177/15291006211051956.

Tukey, J. W. (1977). *Exploratory data analysis*. Reading, Massachussets: Addison-Wesley Publishing Company.

Whalen, K. E., Páez, A., & Carrasco, J. A. (2013). Mode choice of university students commuting to school and the role of active travel. *Journal of Transport Geography, 31*, 42–132. https://doi.org/10.1016/j.jtrangeo.2013.06.008.

Wilkinson, L. (2013). *The grammar of graphics*. Springer Science & Business Media.

Chapter 3
Fundamental Concepts

Ignorance gives one a large range of probabilities.

— George Eliot

3.1 Why Modeling Choices?

In Chap. 1, we discussed the use of models in a fairly general fashion. There, we argued that modeling is an activity useful to isolate in a systematic way certain aspects of a process or thing, by way of abstraction and generalization. There are many kinds of models: analog (like sculptures, maquettes, scale models), conceptual (like mental maps), and mathematical/statistical models.

The raw materials of mathematical/statistical models are observations about the process or thing of interest, usually measurements that provide *data*. The tools are the statistical and mathematical techniques used to convert data into *information*. And the technical expertise is the knowledge and ability of the modeler to use the appropriate tools, in order to extract as much information as possible, given the characteristics of the data and the process or thing.

Modeling choices is simply a specialized field in the much broader field of mathematical and statistical modeling. The task of modeling choices is in many ways similar to statistical models for qualitative and limited dependent variables [i.e., variables with limited ranges; see Maddala (1983)]. Discrete choice analysis is distinguished from models in statistics by its behavioral foundations. Indeed, where statistical models deal with probabilities of an item of interest being in a certain state, choice modeling deals with the probability of an agent *choosing* an alternative. This is a subtle but important difference that we will highlight in due course. For the time being, it is sufficient to say that to model choices we need a conceptual model first on which to build the rest of the apparatus required for applied choice modeling.

A. Páez and G. Boisjoly, *Discrete Choice Analysis with R*, Use R!,
https://doi.org/10.1007/978-3-031-20719-8_3

Before delving into the technical details, we can pause for a philosophical moment to think about human behavior and decision-making.

There are different perspectives on human behavior. Some schools of thought affirm that events are predetermined. A famous thought experiment, referred to as Laplace's Demon, is as follows:

> *We may regard the present state of the universe as the effect of its past and the cause of its future. An intellect [Laplace's Demon] which at a certain moment would know all forces that set nature in motion, and all positions of all items of which nature is composed, if this intellect were also vast enough to submit these data to analysis, it would embrace in a single formula the movements of the greatest bodies of the universe and those of the tiniest atom; for such an intellect nothing would be uncertain and the future just like the past would be present before its eyes.*
>
> — Pierre Simon Laplace, A Philosophical Essay on Probabilities

Some schools of sociological thought (see for example the discussion in Degenne and Fors 1999) see social interactions as a predominant, and even perhaps determinant factor that affects behavior. In its more extreme form, structuralism views social networks as structures that limit the ability of the individual to exercise independent agency, with the consequence that behavior is completely determined by position within the structure.

Laplace's Demon and other forms of causal determinism assume that all preceding events set the conditions for present and future events via immutable rules. Nowadays determinism cannot be seriously considered as a plausible explanation for physical or social phenomena for several reasons, of which it is useful to highlight two:

1. The practical impossibility of knowing at a certain moment all forces that set nature in motion, as well as the positions of all of nature's items.

With respect to physical processes, the uncertainty principle of quantum physics put an end to the idea that we can know all that there is to know about the fundamental items of nature. In terms of human behavior, this is complicated by the inability of an external observer to know the state of mind of a person who acts. On the other hand, it is possible that existing social structures influence behavior [and there is now a wealth of literature that makes this argument; see Páez and Scott (2007)]. However, social determinism seems as implausible as physical determinism, for similar reasons: the difficulties of knowing the state of a system with complete omniscience.

2. The assumption of immutable rules.

This assumption has been challenged by studies that suggest some important physical constants can change with the age of the universe (Webb et al. 2001). In terms of human behavior, the assumption is even more problematic, if for no other reason that humans can in general act, if they so wish, in a contrarian way simply to demonstrate that there are no immutable social rules. This is only one of many reasons why behavioral detection is problematic [for instance, in airports; Kirschenbaum (2013)]: if one knows the rules used for profiling, acting otherwise renders profiling ineffective.

Does this mean that the state of the universe is not determined by the past? Not at all. For all we know, from the perspective of a hypothetical all-knowing being, it is. However, in practical terms, and for the reasons described briefly here, we humble non-all-knowing beings, cannot rely on determinism for making sweeping statements about the state of the universe. In particular, we will make a distinction that is useful as part of developing a conceptual model of choice-making: (1) that there is an observer who typically lacks at least some relevant information about a choice process (let alone about the state of the universe), and (2) that the rules of decision-making are not completely known and/or humans can, for idiosyncratic reasons, alter them at whim.

3.2 How to Use This Chapter

Remember that the source code used in this chapter is available. Throughout the notes, you will find examples of code in segments of text called *chunks*. This is an example of a chunk:

```
print("¡Hola, Prof. Ortúzar!")
```

```
[1] "¡Hola, Prof. Ortúzar!"
```

If you are working with RStudio you can type the chunks of code to experiment with them. As an alternative, you may copy and paste the source code into your `R` or RStudio console, or create a script/notebook to save the code and any experiments you may conduct.

3.3 Learning Objectives

In this chapter, you will learn about:

1. Choice mechanisms: Utility maximization.
2. Probabilities and integration.
3. How to derive a simple choice model.
4. Other choice mechanisms.

3.4 Suggested Readings

- Ben-Akiva, M., & Lerman, S. R. (1985) *Discrete choice analysis: Theory and applications to travel demand*, **Chapter 3**. MIT Press.
- Hensher, D. A., Rose, J. M., & Greene, W. H. (2005) *Applied choice analysis: A primer*, **Chapter 3**. Cambridge University Press.
- Louviere, J. J., Hensher, D. A., & Swait, J. D. (2000) *Stated choice methods: Analysis and application*, **Chapter 1**. Cambridge University Press.

- Ortuzar, J. D., & Willumsen, L. G. (2011) *Modeling transport*, 4th ed., **Chapter 7**. Wiley.

3.5 Preliminaries

Load the packages used in this section:

```
library(dplyr) # A Grammar of Data Manipulation
library(evd) # Functions for Extreme Value Distributions
library(ggplot2) # Create Elegant Data Visualisations Using the Grammar of Graphics
```

3.6 Choice Mechanisms: Utility Maximization

A choice is a result of a decision-making process that can be described as follows: first, the decision-maker defines the problem for which a decision is required; second, the decision-maker identifies the existing alternatives and evaluates each alternative based on a set of attributes; finally, the decision-maker makes a choice based on a decision rule (also referred to as choice mechanism).

Let us take the example of a university student as the decision-maker. The choice problem could be to decide which mode to use to travel to university. The existing alternatives would be defined based on the modes available to the decision-maker, either as a result of transport network (e.g.: public transport supply) or personal characteristics (e.g.: car or bicycle ownership). Further, assume that the student has the possibility to either walk or use public transport to university. The student will then characterize each alternative based on a set of attributes that are important to them (e.g.:travel time and travel cost). Suppose that the choices have the following characteristics:

- walk: travel time = 25 min, cost = $0
- public transport: travel time = 7 min, cost =$3

Using this information, the student will finally make a choice based on a decision rule. The decision rule could be to take the cheapest mode, the fastest mode or make some trade-off between cost and travel time (e.g., the student is willing to pay $1 for each reduction of 5 min in travel time).

The decision rule refers to the internal mechanism or thought process made by the decision-maker to select a unique alternative, using the information that is available to them. This mechanism is central to the conceptualization of choice modeling. While different mechanisms are plausible, the literature has proposed a series of decision rules that can be used to model choices.

In this section, we will begin by defining a model of choice where the decision rule comes from neo-classical economics, fundamentally consumer choice. Conceptually, the choice framework in neo-classical economics is based on the concept of *utility*.

3.6.1 What Is *Utility?*

In simple terms, utility is a summary indicator of the pleasure, usefulness, enjoyment, or attractiveness associated with making a choice (for instance, buying a new phone).

We will begin by positing a very simple choice situation, in which a decision-maker can choose between one of two different *alternatives*. These alternatives constitute the *choice set*, and provide the context for the decision-making process. Imagine then that this simple choice example is as follows:

- Alternative 1: Do nothing (keep using current phone)
- Alternative 2: Buy new phone (in reality there are dozens of different phones, but for simplicity we will think at the moment of a generic model).

Further, assume that each alternative can be described by means of a vector of *attributes* X as follows:

$$X = [x_1, x_2, \ldots, x_k]$$

In other words, the attributes measure important aspects of the thing, as discussed in Chap. 1. In this case, the attributes describe each alternative in a way that the analyst believes is relevant to the decision-maker. In the present example, two relevant attributes are the cost of each alternative and the characteristics of the current and new phones, for instance, their download speeds. In this way, the two choices can be described by their attributes as follows:

$$\begin{aligned} \text{Do-Nothing:} \quad X_N &= [\text{cost}_\text{N},\ \text{speed}_\text{N}] \\ \text{Buy new phone:} \quad X_B &= [\text{cost}_\text{B},\ \text{speed}_\text{B}] \end{aligned}$$

If the decision-maker currently owns a phone that is fully paid, the out-of-pocket cost of doing nothing would presumably be zero. Buying a new phone, on the other hand, would have a positive (and possibly substantial) cost. The new phone, on the other hand is faster than the older, currently owned model.

A decision-maker can likewise be described by a vector of attributes, say Z:

$$Z = [z_1, z_2, \ldots, z_k]$$

Suppose, for example, that decision-maker i can be described in terms of their income, as follows:

$$Z_i = [\text{income}_i]$$

The attributes of the decision-maker help to capture heterogeneities in behavior: for instance, a decision-maker with a lower income may be more sensitive to cost, since buying a new phone is relatively more expensive.

A utility function is a way of summarizing the attributes of the alternatives and the attributes of the decision-makers in a single quantity, which is what the decision-maker is trying to maximize. We assume that each course of action gives this consumer a level of *utility*: in other words, they will be more or less happy with each alternative, taking into account the characteristics of the alternatives and the decision-maker's own condition or status:

$$\begin{aligned} V_{i,\mathrm{N}} &= V(\mathrm{cost}_{\mathrm{N}},\ \mathrm{speed}_{\mathrm{N}},\ \mathrm{income}_i) \\ V_{i,\mathrm{B}} &= V(\mathrm{cost}_{\mathrm{B}},\ \mathrm{speed}_{\mathrm{B}},\ \mathrm{income}_i) \end{aligned}$$

Notice that a utility function is specific to a decision-maker i and an alternative.

Here we define a decision-making rule. The decision-maker considers the utility of the alternatives, and chooses the one that gives the highest utility. In other words, decision-maker i will choose to keep the current phone if

$$V_{i,\mathrm{N}} > V_{i,\mathrm{B}}$$

If the reverse is true, then the decision-maker will choose to buy a new phone (in the case of a tie, the decision-maker is indifferent between the two alternatives).

We assume that decision-makers behave in a *rational* way, and that they do some kind of analysis of the costs and benefits associated with each alternative before making the choice. *Rational* here has the narrow meaning that the decision-maker *always* chooses the best alternative. The analyst, however, may fail to observe every single aspect of the decision-making process. For instance, a decision-maker may need faster speeds because of a lack of internet at home. Or younger people may be more willing to buy new phones than older people. Or a decision-maker may have just received a large gift from a relative. The analyst may observe a decision-maker with a relatively low income buying a new phone. While income alone would have suggested that the decision-maker would be better off keeping the old phone, the analyst is likely unaware of idiosyncratic conditions, such as a recent gift.

For this reason, it is convenient to define the utility in terms of (1) a systematic component, that is, the part that explains the decision-makers' response to the observed attributes of the alternative, and (2) a random component, which captures other aspects of the decision-making process that the analyst is ignorant about:

$$\begin{aligned} U_N &= V_N + \epsilon_N \\ U_B &= V_B + \epsilon_B \end{aligned}$$

The random part of the function is called the *random utility*. If there was no uncertainty at all, if we knew precisely all there was to know about the decision-making process, we would have no ϵ_{N} and ϵ_{B}. Accordingly, $U_{\mathrm{N}} = V_{\mathrm{N}}$ and $U_{\mathrm{B}} = V_{\mathrm{B}}$, and we could predict with complete certainty the choice. Conversely, the presence of the random components implies that we cannot be certain whether $U_A > U_B$.

While this is unfortunate, the presence of the random terms does allow us to make a probabilistic statement, such as

$$P_{\mathrm{N}} = P(U_{\mathrm{N}} > U_{\mathrm{B}}) = P(V_{\mathrm{N}} + \epsilon_{\mathrm{N}} > U_{\mathrm{B}} + \epsilon_{\mathrm{B}})$$

The above means that the probability of doing nothing equals the probability that the utility of doing nothing is greater than the utility of buying a new phone. After rearranging things, this is equivalent to

$$P_{\mathrm{N}} = P(\epsilon_{\mathrm{B}} - \epsilon_{\mathrm{N}} < V_{\mathrm{N}} - V_{\mathrm{B}})$$

The expression above is the cornerstone of random utility modeling in discrete choice analysis. Before we make more progress, however, we have to answer an important question.

3.7 What About Those Random Terms?

A probabilistic expression is clearly better than being unable to say anything interesting at all regarding choices. To make this expression of practical use, we must make some assumptions about the random terms, in particular regarding their distribution across a population or sample of interest. This means that we need to define some *probability distribution function*.

3.8 Probability Distribution Functions (PDFs) and Cumulative Distribution Functions (CDFs)

The random terms in discrete choice analysis are given by probability distribution functions (PDFs). PDFs are mathematical constructs that are useful to describe randomness. A candidate for a probability distribution function is any function that satisfies the following two conditions:

$$\begin{array}{l}\text{Condition 1: } f(x) \geq 0 \text{ for all } x\\ \text{Condition 2: } \int_{-\infty}^{\infty} f(x)dx = 1\end{array}$$

These two conditions say that the function must take values of at least zero for the interval of x of interest, and that the area under the curve (that is what the integral means) must equal 1.

Consider the following example to illustrate these properties. We will define the following function:

$$f(x) = \begin{cases} 0 & x \leq -L \\ \frac{1}{2L} & -L < x < L \\ 0 & x \geq L \end{cases}$$

This function is shown in Fig. 3.1, with $L = 2$:

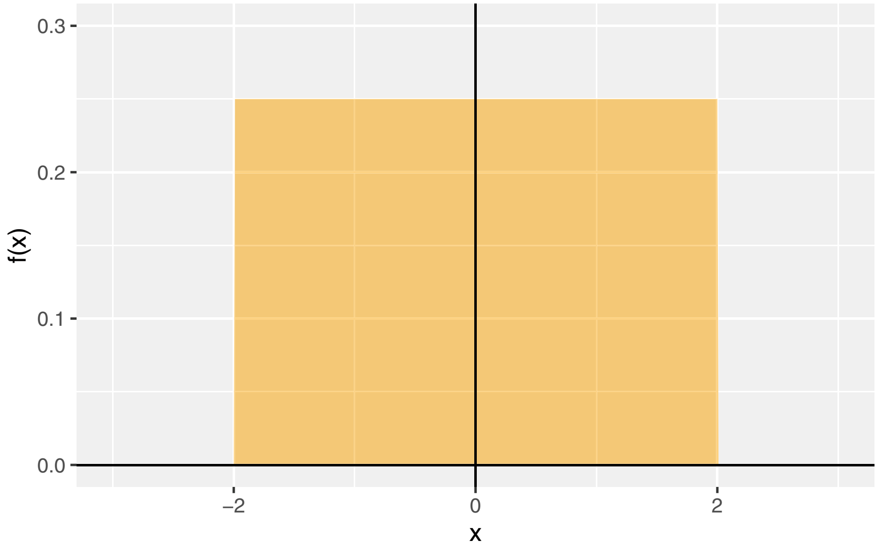


Fig. 3.1 Uniform distribution

```
# Define a function to return a value of one if
# -L > x <= L and zero otherwise
uniform <- function(x, L){
  # Logical condition: x greater than minus L and less than L
  ifelse(x > -L & x <= L,
         # Value if true
         1/(2 * L),
         # Value if false
         0)
}

# Define parameter L for the distribution
L <- 2

# Create a data frame for plotting
# The function `seq()` is used to create a sequence of
# values starting at `from` and ending at `to` with a
# step increase of `by`
df <- data.frame(x =seq(from = -(L+1),
                        to = L+1,
                        by = 0.01)) %>%
  # Mutate the data frame to add a new column with the
  # value of the function
  mutate(f = uniform(x, L))

# Plot
ggplot(data = df,
      # Map column x in the data frame to the x-axis
      # and column f to the y-axis
      aes(x = x,
          y = f)) +
  geom_area(fill = "orange",
            alpha = 0.5) +
  # Set the limits of the y axis
  ylim(c(0, 1/(2 * L) + 0.2 * 1/(2 * L))) +
  # Draw a horizontal line for the x axis
  geom_hline(yintercept = 0) +
  # Draw a vertical line for the y axis
  geom_vline(xintercept = 0) +
  ylab("f(x)")
```

It is easy to see that the value of the function is always equal to or greater than zero. You can also verify that the area under the curve in this case is simply the area of the rectangle $b \times h$, where the base of the rectangle is $b = L - (-L)$ and the height is $h = \frac{1}{2L}$:

```
(L - (-L)) / (2 * L)
```

```
  [1] 1
```

If you are working with the `R` code, try changing the value of parameter L to see what happens. What is the implication of larger values of L? And of smaller values of L?

Since the function above satisfies the two necessary conditions stipulated before, we conclude that it is a valid probability distribution function. In fact, it turns out to be a form of the uniform probability distribution function, which more generally is defined as

$$f(x) = \begin{cases} 0 & x \le b \\ \frac{1}{a-b} & b < x < a \\ 0 & x \ge a \end{cases}$$

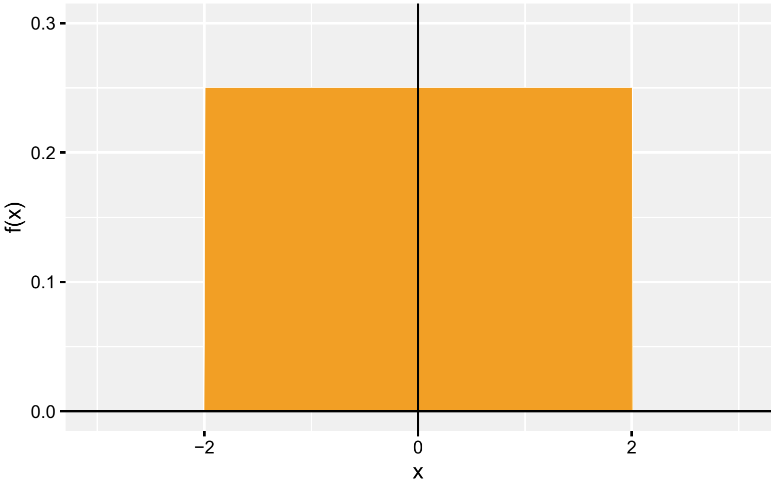
Given a probability distribution function, we can calculate the probability of a random variable x being contained in a defined interval. For instance, the probability of $x < -L$ is zero, since the area under the curve in that case is zero. The probability of $x \le X$ is

$$\int_{-\infty}^{X} f(x)dx$$

In the case of our uniform distribution function, this is simply the area of the rectangle defined by the limits of the integral:

```
# Define L
L <- 2
```

```
# Define an upper limit for calculating the probability
X <- 2

# Create a data frame for plotting the full distribution
df <- data.frame(x =seq(from = -(L + 1),
                        to = L + 1,
                        by = 0.01)) %>%
  # Mutate the data frame to add a new column with the
  # value of the function
  mutate(f = uniform(x, L))

# Create a data frame for plotting the portio of the
# distribution that is less than X
df_p <- data.frame(x =seq(from = -(L + 1),
                          to = X,
                          by = 0.01)) %>%
  mutate(f = uniform(x, L))

# Plot
ggplot() +
  # Use data frame `df` to plot the full distribution
  geom_area(data = df,
            # Map column x in the data frame to the x-axis
            # and column f to the y-axis
            aes(x = x,
                y = f),
            fill = "orange",
            alpha = 0.5) +
  # Use data frame `df_p` to plot area under the curve to X
  geom_area(data = df_p,
            # Map column x in the data frame to the x-axis
            # and column f to the y-axis
            aes(x = x,
                y = f),
            fill = "orange",
            alpha = 1) +
  # Set the limits of the y axis
  ylim(c(0,
         1/(2 * L) + 0.2 * 1/(2 * L))) +
  # Draw a horizontal line for the x axis
  geom_hline(yintercept = 0) +
  # Draw a horizontal line for the y axis
  geom_vline(xintercept = 0) +
  ylab("f(x)")
```



What is the probability that $x \leq 0$? Try changing the upper limit to see what happens. How does the value of the area under the curve change?

Given a probability distribution function we can define a cumulative distribution function (CDF) $F(x) = P(x \leq X)$, which maps how the probability changes as we change the interval. The cumulative distribution function of our uniform distribution is as follows:

$$F(x) = \begin{cases} 0 & x \leq -L \\ \frac{x+L}{2L} & -L < x < L \\ 1 & x \geq L \end{cases}$$

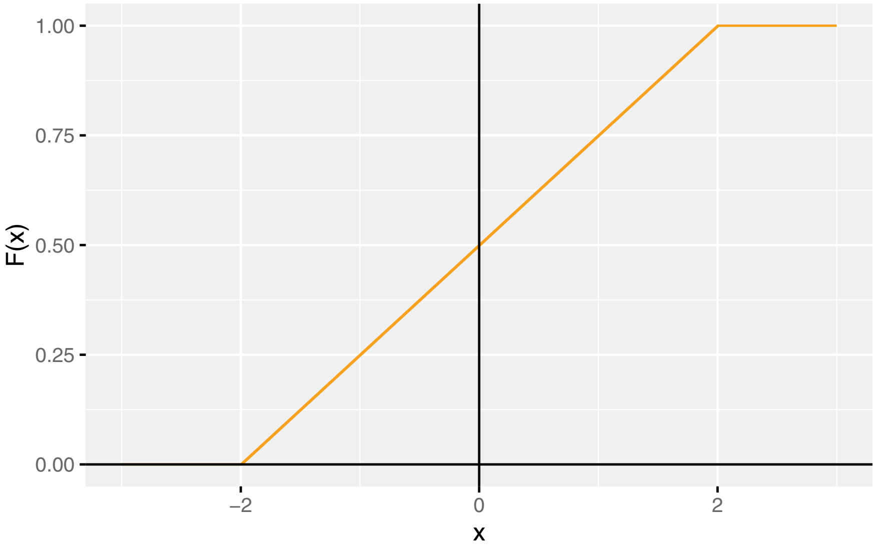


Fig. 3.2 Uniform cumulative distribution function

The cumulative distribution function for our uniform distribution appears in Fig. 3.2:

```
# Define the cumulative distribution function
punif <- function(x, L){
  # Logical statement: x less or equal to -L
  ifelse(x <= -L,
         # Value if true
         0,
         # If false, check whether x is between -L and L
         ifelse(x > -L & x <= L,
                # Value if true
                (x + L)/(2 * L),
                # Value if false
                1))
}


# Define L
L <- 2

# Create a data frame for plotting the cumulative distribution function
df <- data.frame(x =seq(from = -(L+1),
                        to = L+1,
                        by = 0.01)) %>%
  mutate(f = punif(x, L))

# Plot
ggplot(data = df,
       # Map column x in the data frame to the x-axis
       # and column f to the y-axis
       aes(x = x,
           y = f)) +
  # Add geometric object of type step to the plot to
  # plot cumulative distribution function
  geom_step(color = "orange") +
  ylim(c(0, 1)) + # Set the limits of the y axis
  geom_hline(yintercept = 0) + # Add y axis
  geom_vline(xintercept = 0) + # Add x axis
  ylab("F(x)") # Label the y axis
```

As you can see, the probability of $x \le -L$ (in this case -2) is zero, the probability of $x \le 0$ is 0.5, and the probability of $x \le L$ (in this case 2) is one.

We can consider a second example, with a function as follows:

$$f(x) = \begin{cases} 0 & x \le 0 \\ 2x & 0 < x < 1 \\ 0 & x \ge 1 \end{cases}$$

This function is shown in Fig. 3.3.

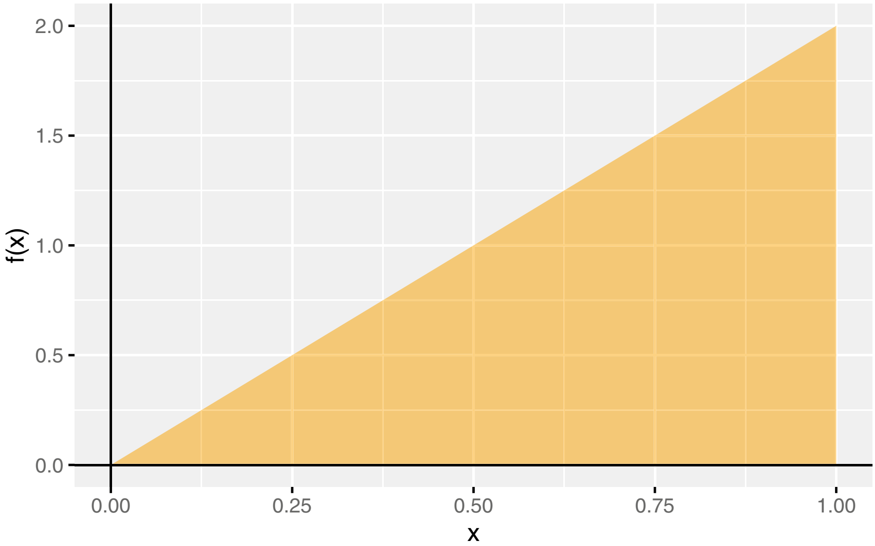


Fig. 3.3 Linear distribution

```r
# Define a function for the linear distribution function
linear <-
function(x){
  # Logical statement: x between 0 and 1
  ifelse( x > 0 & x <= 1,
          # Value if true
          2 * x,
          # Value if false
          0)
}

# Create a data frame for plotting
df <- data.frame(x =seq(from = 0,
                        to = 1,
                        by = 0.01)) %>%
  mutate(f = linear(x))

# Plot
ggplot(data = df,
       aes(x = x,
           y = f)) +
  geom_area(fill = "orange",
            alpha = 0.5) +
  ylim(c(0, 2)) +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0) +
  ylab("f(x)")
```

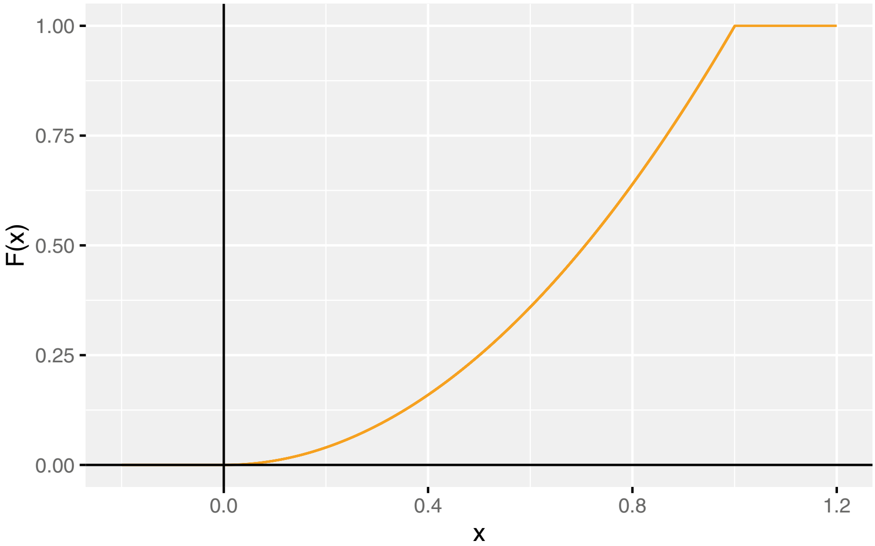


Fig. 3.4 Linear cumulative distribution function

Clearly, $f(x) \geq 0$ for all values of x in the interval $0 \leq x \leq 1$. We can verify that the area under the curve is 1. In this case the area is that of a triangle, i.e., $\frac{b \times h}{2}$. Since the base of the triangle is $b = 1$ and the height is $h = 2$, we see that the area under the curve is 1.

Since this is a valid probability distribution function, we can use it to calculate the probability of $x \leq X$ as shown previously. The area under the curve when $x \leq X$ is given by:

$$F(x) = \begin{cases} 0 & x \leq 0 \\ \frac{x \times 2x}{2} = x^2 & 0 < x < 1 \\ 1 & x \geq 1 \end{cases}$$

The plot of the cumulative distribution function in this case is shown in Fig. 3.4.

```
# Define a function for the cumulative distribution
plinear <- function(x){
  ifelse(x <= 0,
         0,
         ifelse( x > 0 & x <= 1,
                 x^2,
                 1))
}

# Create a data frame for plotting
df <- data.frame(x =seq(from = -0.2,
                        to = 1.2,
                        by = 0.001)) %>%
  mutate(f = plinear(x))

# Plot
ggplot(data = df,
       aes(x = x,
           y = f)) +
  # Add geometric object of type step to the plot to
  # to plot cumulative distribution function
  geom_step(color = "orange") +
  ylim(c(0, 1)) +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0) +
  ylab("F(x)")
```

It should be clear from the examples above that calculating probabilities from a probability distribution function is nothing other than finding the area under the curve of the function. When the function is relatively simple, as the uniform or the linear distributions that we used for the examples, calculating the areas is also straightforward, since the functions describe simple geometric shapes. This becomes more complicated when the function describes curves instead of straight lines [calculus was invented in part to work with curves; see the wonderful non-technical discussion in Strogatz (2019)]. We can illustrate this by means of the following function:

$$f(x) = \begin{cases} 0 & x \leq 0 \\ 4x^3 & 0 < x < 1 \\ 0 & x \geq 0 \end{cases}$$

This function is plotted in Fig. 3.5.

```
# Define a function
cubic <- function(x)
  ifelse(x <= 0,
         0,
         ifelse(x > 0 & x <= 1,
                4 * x^3,
                0 ))

# Create a data frame for plotting
df <- data.frame(x =seq(from = 0,
                        to = 1,
                        by = 0.01)) %>%
  mutate(f = cubic(x))

# Plot
ggplot(data = df,
       aes(x = x,
           y = f)) +
  geom_area(fill = "orange",
            alpha = 0.5) +
```

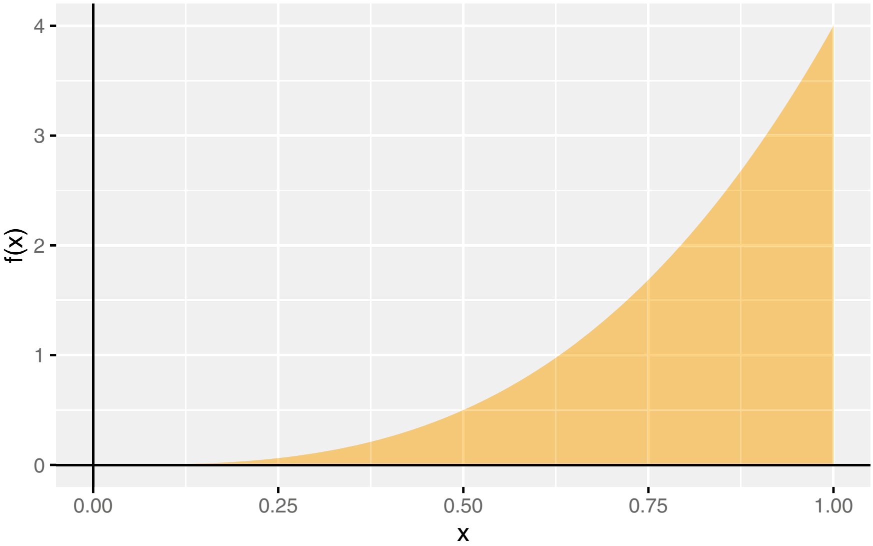


Fig. 3.5 Cubic distribution

```
  ylim(c(0, 4)) +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0) +
  ylab("f(x)")
```

Unlike the rectangle of the uniform distribution and the triangle of the linear distribution, this function involves curved lines, and the area under this distribution cannot be calculated using a formula such as $b \times h$ or $\frac{b \times h}{2}$. These formulas actually can be derived using integration, and they are what are called "closed form" solutions to the integral. In the case of the cubic function the area under the curve is calculated by integration as follows:

$$\int_0^1 4x^3dx = 4\left[\frac{x^4}{4}\right]_0^1 = \left[x^4\right]_0^1 = 1^4 - 0 = 1$$

Without worrying too much about the mathematics of working with integrals, this serves to show that the function is a valid probability distribution function. However, the need to integrate makes things more interesting, to say the least! Fortunately, for most applied discrete choice analysis we do not usually need to solve integrals manually (the monster minds have already done this for us!). The key here is to remember: given a valid probability distribution function **the probability that a random variable $x \leq X$ is the area under the curve in the interval $-\infty$ to X.**

3.9 A Simple Random Utility Discrete Choice Model

We are now ready to deploy a probability distribution function to our posited choice mechanism. Returning to our binary choice example, where the probability of choosing alternative N is equal to the probability that $V_{\mathrm{N}} - V_{\mathrm{B}} > \epsilon_{\mathrm{B}} - \epsilon_{\mathrm{N}}$. Let us assume that the difference in the random utility terms follows the uniform distribution with parameters $-L$ and L, that is

$$\epsilon_{\mathrm{B}} - \epsilon_{\mathrm{N}} \sim \mathcal{U}(-L, L)$$

Note that $\mathcal{U}$ here refers to the uniform distribution function and not the utility function.

The probability of choosing the "do-nothing alternative" is

$$P_{\mathrm{N}} = P(\epsilon_{\mathrm{B}} - \epsilon_{\mathrm{N}} < V_{\mathrm{N}} - V_{\mathrm{B}})$$

Since we know the probability that a random variable is less than a certain value in the uniform distribution, we have that

$$P_{\mathrm{N}} = \begin{cases} 0 & V_{\mathrm{N}} - V_{\mathrm{B}} \leq -L \\ \frac{V_{\mathrm{N}} - V_{\mathrm{B}} + L}{2L} & -L < V_{\mathrm{N}} - V_{\mathrm{B}} < L \\ 1 & V_{\mathrm{N}} - V_{\mathrm{B}} \geq L \end{cases}$$

This corresponds to the uniform cumulative distribution function $F_X(x) = P(x \leq X)$ seen above, which calculates the probability of $(V_{\mathrm{N}} - V_{\mathrm{B}})$ being less than a specific value.

We can unpack this expression as follows.

When the systematic utility of a new phone is greater than the systematic utility of doing nothing, the difference between these two terms is negative. The more negative this value is, the lower the probability of doing nothing. When the difference is more negative than $-L$, the probability of doing nothing becomes zero.

When the systematic utility of a new phone is identical to the systematic utility of doing nothing, the difference between these two terms is zero, in which case the probability of doing nothing is 0.5. In other words, there is a 50% chance that the decision maker will do nothing.

Finally, when the systematic utility of a new phone is less than the systematic utility of doing nothing, the difference between these two terms is positive. The more positive this value is, the higher the probability of doing nothing. When the difference is greater than L, the probability of doing nothing becomes one.

Now, since the choice set is an exhaustive collection of courses of action, it follows that the probability of the two courses of action must add up to one (the decision-maker does nothing OR buys a new phone):

$$P_{\mathrm{N}} + P_{\mathrm{B}} = 1$$

This means that once we know the probability of doing nothing, the probability of buying a new phone is simply the complement:

$$P_{\mathrm{B}} = 1 - P_{\mathrm{N}}$$

These probabilities are a discrete choice model. In fact, this is called the linear probability model (see Ben-Akiva and Lerman 1985, 66–68). Other models can be obtained by selecting different probability distribution functions, as we will see in later chapters. The procedure followed here will be the same, even if the probability distribution function selected for the model is different: given a valid probability distribution function, and given the systematic utilities of the alternatives, it is possible to evaluate the probabilistic statement associated with the choice of an alternative.

On a final note, before discussing other choice mechanisms. The simple example used here was a binomial choice situation, i.e., a situation with only two alternatives. This was done for convenience of exposition, and we will see how the ideas presented here generalize for situations with more than two alternatives, that is, for *multinomial* choice situations.

3.10 Other Choice Mechanisms

Utility maximization is only one of several plausible choice mechanisms. Utility functions assume that trade-offs among different attributes are possible; for example, the way the utility functions were formulated assumes that a decision-maker is willing to pay more for higher download speeds. While such trade-offs are plausible in many situations, other choice mechanisms could apply in other cases, as discussed by Ortuzar and Willumsen (2011, Fourth Edition: 241–243).

Consider as an example a user who is shopping for smartphones. This individual may have low tolerance for download speeds below a certain threshold, or may have a budget limit that prevents them from considering certain models. Some alternatives from the choice set may be eliminated or ranked based on some dominant attribute. This kind of choice mechanism is called *lexicographic* choice, or elimination by attributes.

Another plausible choice mechanism is a form of satisficing behavior. Again, a user shopping for a smartphone might find a model that does not maximize their utility, but that is otherwise satisfactory. For example, the decision-maker may consider that the additional time spent finding an even better model is not worth their while, so they stop the search at a sub-optimal point. Would this become a utility maximization situation if the cost of the search was incorporated as part of the utility?

Another choice mechanism seen recently in the literature is regret minimization (see Chorus 2010).

In addition to these various mechanisms, it is possible that a decision-maker deploys combinations of them for any one given choice situation: for example, a person might begin with a lexicographic search to reduce the number of alternatives they need to consider, followed by utility maximization or regret minimization. There has been much progress on these models, but for the time being utility maximization remains the most widely used approach for the analysis of discrete choices.

3.11 Exercises

1. Define utility.
2. Describe in your own words the behavior described by utility maximization.
3. What conditions are necessary for a function to be a valid probability distribution function?

   Consider the function shown in Fig. 3.6. This is called the triangle or tent function.
4. Show that the triangle function in the figure is a valid probability distribution function.
5. Next, consider the following utility functions for two alternatives, namely i and j:

$$U_i = V_i + \epsilon_i$$
$$U_j = V_j + \epsilon_j$$

   Assume that the difference between the error terms below follows the triangle distribution:

$$\epsilon_q = \epsilon_i - \epsilon_j$$

Parting from the assumption above, derive a binary choice model for the probability of selecting alternative j.

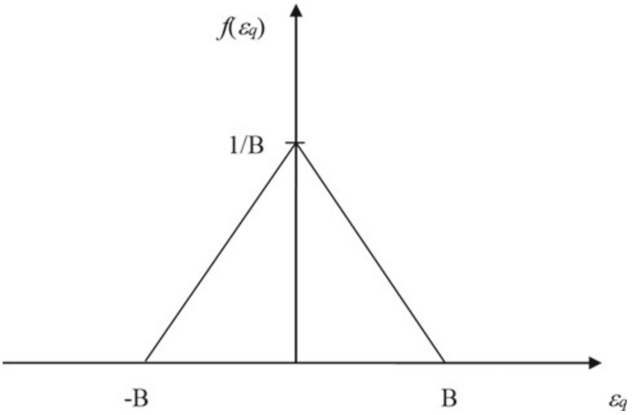


Fig. 3.6 Triangle function

References

Ben-Akiva, M., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and applications to travel demand*. Cambridge: The MIT Press.

Chorus, C. G. (2010). A new model of random regret minimization. *Eurpean Journal of Transport Infrastructure Research, 10*(2), 181–96.

Degenne, A., & Fors, M. (1999). *Introducing social networks*. Introducing statistical methods. London: SAGE Publications.

Hensher, D. A., Rose, J. M., Rose, J. M., & Greene, W. H. (2005). *Applied choice analysis: A primer*. Cambridge University Press.

Kirschenbaum, A. A. (2013). The cost of airport security: The passenger dilemma. *Journal of Air Transport Management, 30*, 39–45. https://doi.org/10.1016/j.jairtraman.2013.05.002.

Louviere, J. J., Hensher, D. A., & Swait, J. D. (2000). *Stated choice methods: Analysis and applications*. Cambridge University Press.

Maddala, G. S. (1983). *Limited-dependent and qualitative variables in econometrics*. Cambridge: Cambridge University Press.

Ortúzar, J. D., & Willumsen, L. G. (2011). *Modelling transport* (4th ed.). New York: Wiley.

Pàez, A., & Scott, D. M. (2007). Social influence on travel behavior: A simulation example of the decision to telecommute. *Environment and Planning A, 39*(3), 647–65.

Strogatz, S. (2019). *Infinite powers: How calculus reveals the secrets of the universe*. Eamon Dolan Books.

Webb, J. K., Murphy, M. T., Flambaum, V. V., Dzuba, V. A., Barrow, J. D., Churchill, C. W., et al. (2001). Further evidence for cosmological evolution of the fine structure constant. *Physical Review Letters, 87*(9), 091301.

Chapter 4
Logit

I believe that we do not know anything for certain, but everything probably.

— Christiaan Huygens

4.1 Modeling Choices

In Chap. 2, we introduced a fundamental conceptual framework to model choice-making behavior. This framework is based on the economic notion of *utility*, basically that which decision-makers wish to maximize when making choices. The concept of utility has many flaws—key among them is that it is not directly observable. If utility could be measured directly by an external observer (or analyst), behavior would seem deterministic. However, unlike Laplace's Demon, an external observer with only human capabilities has limited knowledge of the conditions under which choices are made, if for no other reason that they cannot possibly know the frame of mind of the decision-maker at the moment when choices are made.

A way to implement the conceptual framework under such conditions involved an acknowledgement that although the decision-maker tries to maximize their utility, some part of it will look random to the observer - therefore the term *random utility modeling*. Thus, while the analyst cannot make definitive statements about choices, it is possible to make probabilistic statements about the behavior of decision-makers. In other words, the analyst does not understand with complete certainty the choice process, but can quantify uncertainty in a fairly precise way.

Based on these concepts, Chap. 3 concluded by deriving a simple model for discrete choices, namely the linear probability model (see Ben-Akiva and Lerman 1985, 66–68). This model is useful for illustrative purposes. However, it suffers from an important limitation: the linear probabilities are a stepwise function, which makes their mathematical treatment unfun, and also imply that some outcomes are certain

A. Páez and G. Boisjoly, *Discrete Choice Analysis with R*, Use R!,
https://doi.org/10.1007/978-3-031-20719-8_4

(i.e., it can return probabilities of exactly one or exactly zero). This would preclude some behaviors, which is a somewhat arrogant assumption on the part of the analyst. A better approach would be to allow for any behavior, but instead assign very small probabilities to more extreme choices.

In this chapter, we will revisit the random utility terms in the probabilistic statement concerning choices, and we will derive an alternative to the linear probability model. This will be the *logit* model, one of the most popular models in discrete choice analysis for reasons that will be discussed below.

4.2 How to Use This Chapter

Remember that the source code used in this chapter is available. Throughout the notes, you will find examples of code in segments of text called *chunks*. This is an example of a chunk:

```
print("Hello, Prof. Train")
```

```
[1] "Hello, Prof. Train"
```

If you are working with RStudio you can type the chunks of code to experiment with them. As an alternative, you may copy and paste the source code into your `R` or RStudio console, or create a script/notebook to save the code and any experiments you may conduct.

4.3 Learning Objectives

In this chapter, you will learn about:

1. The Extreme Value Type I (EV Type I) distribution.
2. The binary logit model.
3. The multinomial logit model.
4. Properties of the logit model.

4.4 Suggested Readings

- Ben-Akiva, M., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and applications to travel demand*, **Chapters 4 and 5**. MIT Press.
- Hensher, D. A., Rose, J. M., & Greene, W. H (2005). *Applied choice analysis: A primer*, **Chapter 10**. Cambridge University Press.

- Louviere, J. J., Hensher, D. A., & Swait, J. D. (2000). *Stated choice methods: Analysis and application*, **Chapter 3, pp. 199–205**. Cambridge University Press.
- Ortuzar, J. D., Willumsen, & L. G. (2011). *Modelling transport*, 4th edn., **Chapter 7**. Wiley.
- Train, K. (2009). *Discrete choice methods with simulation*, 2nd edn., **Chapter 3**. Cambridge University Press.

4.5 Preliminaries

Load the packages used in this section:

```r
library(dplyr) # A Grammar of Data Manipulation
library(evd) # Functions for Extreme Value Distributions
library(ggplot2) # Create Elegant Data Visualisations Using the Grammar of Graphics
```

4.6 Once Again Those Random Terms

Recall that implementation of the probabilistic statement at the heart of a discrete choice model requires the analyst to make assumptions about the random utility terms. Previously, a number of probability distributions were explored, and one in particular (the uniform distribution) was used to derive a simple discrete choice model. But, is the uniform distribution an *adequate* choice for the purpose of modeling the random utility?

The uniform distribution (and some of the other stepwise distributions seen in Chap. 2) are useful to illustrate the concept of probability, and more specifically the need to calculate the area under the curve of the distribution. The area under the curve of the uniform distribution is simply the area of a rectangle, which makes this task extremely simple since it requires only the formula for the area of a rectangle. On the other hand, it precludes certain outcomes, which limits its practical usefulness.

The reality is that, since the utility is in principle unobservable, there is little theoretical support for any specific distribution of the random utility terms. For this reason, the choice of a specific distribution tends to be very pragmatic, and usually pays attention to its convenience for estimation purposes, i.e., for the task of retrieving parameters from a sample of observations. Another consideration is the ability of the distribution to represent different types of behavior, as will be seen.

The parameters that need to be retrieved via estimation include the parameters used in the systematic utility function V_{ij} (more on this later), as well as any other parameters needed by the distribution itself. For instance, the uniform distribution is defined by two parameters, a and b:

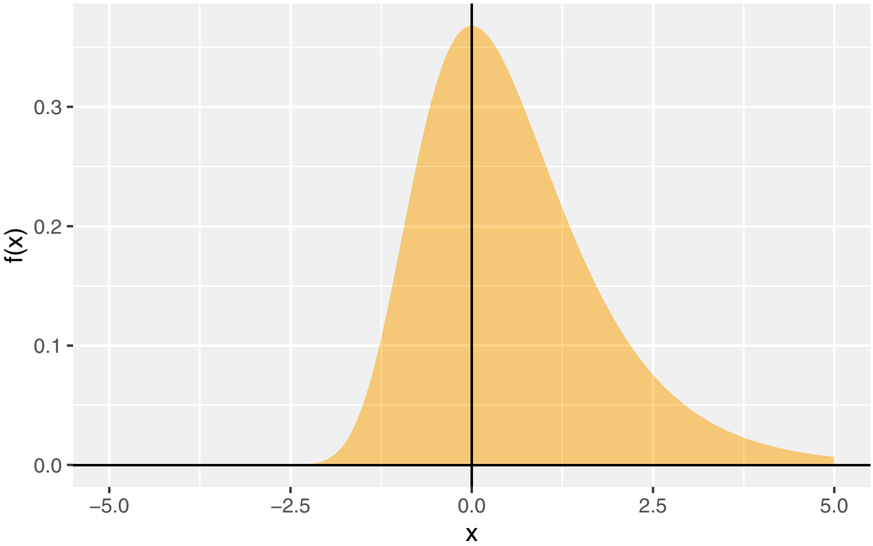


Fig. 4.1 Extreme value Type I distribution

$$f(x) = \begin{cases} 0 & x \leq b \\ \frac{1}{a-b} & b < x < a \\ 0 & x \geq a \end{cases}$$

These parameters define the *scale* of the distribution. The scale controls the shape of the distribution, in this case how wide or narrow it is. The greater the difference between a and b the greater the range of values with non-zero probability, but the lower the probability between any given constant interval of values of x. These two parameters also determine the *location* of the distribution (i.e., the mid-point of the rectangle). In this way, the uniform distribution is *located* at $\frac{a-b}{2}$. Other distributions also have parameters that determine their location and scale.

A convenient choice of distribution is the Extreme Value Type I probability distribution function (EV Type I). This function is defined as

$$f(x; \mu, \sigma) = e^{-(x+e^{-(x-\mu)/\sigma})}$$

The EV Type I distribution has two parameters, namely μ and σ, which determine the location (i.e., the center) and the scale of the distribution, respectively.

The shape of this distribution is shown in Fig. 4.1 with $\mu = 0$ and $\sigma = 1$:

```
# Define parameters for the distribution
# Location
mu <- 0
# Scale
sigma <- 1
```

```
# Create a data frame for plotting;
df <- data.frame(x =seq(from = -5,
                        to = 5,
                        by = 0.01)) %>%
  # The function `dgumbel()` is the EV Type I distribution
  mutate(f = dgumbel(x,
                     # Location parameter
                     loc = mu,
                     # Scale parameter
                     scale = sigma))

# Plot
ggplot(data = df,
       aes(x = x,
           y = f)) +
  geom_area(fill = "orange",
            alpha = 0.5) +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0) +
  ylab("f(x)")
```

If you are working with the R code, you can try changing the parameters to see how the function behaves (remember to adjust the starting and ending values of the sequence for the data frame if you change the center of the distribution!).

The EV Type I has a very interesting property: the difference of two EV Type I distributions follows the logistic distribution. In other words, if we let $X \sim \text{EVI}(\alpha_Y, \sigma)$ and $Y \sim \text{EVI}(\alpha_Y, \sigma)$, then:

$$X - Y \sim \text{Logistic}(\alpha_X - \alpha_Y, \sigma)$$

4.7 The Logit Model

As noted above the difference of two random variables that follow the EV Type I distribution is a new random variable that follows the logistic distribution. Recall that our probability statement regarding choices was (in general form):

$$P_i = P(\epsilon_j - \epsilon_i < V_i - V_j)$$

We can assume that the random utility terms ϵ follow the EV Type I distribution, in which case their difference (i.e., $\epsilon_n = \epsilon_j - \epsilon_i$) follows the logistic distribution. This distribution, in turn, is as follows:

$$f(x; \mu, \sigma) = \frac{e^{-(x-\mu)/\sigma}}{\sigma(1 + e^{-(x-\mu)/\sigma})^2}$$

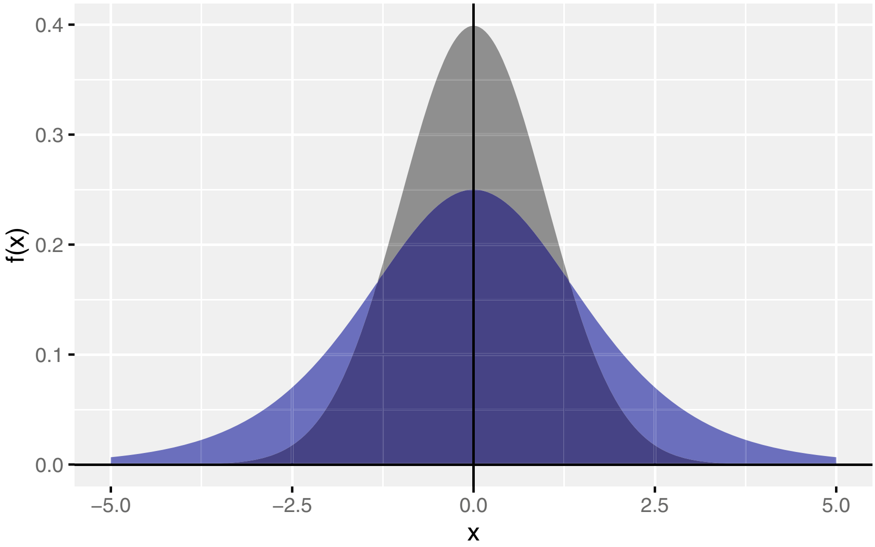


Fig. 4.2 Comparison of the logistic (blue) and normal (grey) distributions

While the EV Type I distribution is not symmetric, the shape of the logistic distribution is. The logistic distribution is, in fact, quite similar to the normal distribution but for the same location and scale parameters it has fatter tails (the two ends of the distribution), which means that for identical parameters the probability of extreme values is higher. This is illustrated in Fig. 4.2:

```
# Define parameters for the distribution
# Location
mu <- 0
# Scale
sigma <- 1

# Create a data frame for plotting
df <- data.frame(x =seq(from = -5,
                        to = 5,
                        by = 0.01)) %>%
  # Add columns with the values of the
  # `dlogis()` and `dnorm()` distributions
  mutate(logistic = dlogis(x,
                           # Location parameter
                           location = mu,
                           # Scale parameter
                           scale = sigma),
         normal = dnorm(x,
                        # The location parameter of the normal
                        # distribution is the mean
                        mean = mu,
                        # The scale parameter of the normal
                        # distribution is the standard deviation
                        sd = sigma))

# Plot
ggplot() +
  # Add geometric object of type area to plot the
  # logistic distribution
  geom_area(data = df,
            aes(x = x,
                y = logistic),
            # The fill color of the logistic distribution
            fill = "blue",
            alpha = 0.5) +
  # Add geometric object of type area to plot the
  # normal distribution
  geom_area(data = df,
            aes(x = x,
                y = normal),
            # The fill color of the normal distribution
            fill = "black",
            alpha = 0.5) +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0) +
  ylab("f(x)") # Label the y axis
```

Since the probability expression is given in terms of the difference of the random utilities, if we assume that the random terms ϵ follow the EV Type I distribution, their difference (i.e., $\epsilon_n = \epsilon_i - \epsilon_j$) follows the logistic distribution.

As before, to obtain a probability we need to calculate the area under the curve of the function. Unfortunately, this needs to be done by integration. Fortunately, this integral has an analytical solution, or so-called *closed form*:

$$F(x; \mu, \sigma) = \frac{1}{1 + e^{-(x-\mu)/\sigma}}$$

The analytical solution is the exact value of the integral. Accordingly, the probability expression in terms of the utilities becomes:

$$P_i = P(\epsilon_n < V_i - V_j) = \frac{1}{1 + e^{-(V_i - V_j - \mu)/\sigma}}$$

After some manipulation, the above expression can be rewritten as

$$P_i = P(\epsilon_n < V_i - V_j) = \frac{e^{V_i/\sigma}}{e^{V_i/\sigma} + e^{(V_j+\mu)/\sigma}}$$

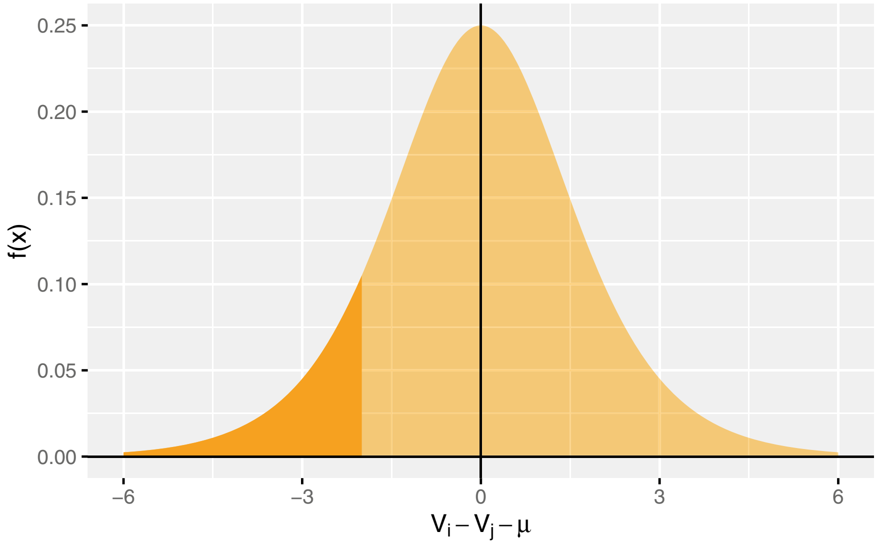


Fig. 4.3 Logit probability

The above is called the *logit probability* and the resulting model is called the *logit model*. As seen in Fig. 4.3, the probability of choosing alternative i is the area under the curve of the logistic distribution function to the left of $X = V_i - V_j - \mu$.

```
# Define parameters for the distribution
# Location
mu <- 0
# Scale
sigma <- 1

# Define an upper limit for calculating the probability;
# This equivalent to V_i - V_j.
# Negative values represent V_i < V_j, and positive values are V_j > V_i;
# when V_j = V_k, then X = 0:
x <- -2

# Create data frames for plotting
df <- data.frame(x =seq(from = -6 + mu,
                        to = 6 + mu,
                        by = 0.01)) %>%
  mutate(y = dlogis(x,
                    location = mu,
                    scale = sigma))
df_p <- data.frame(x =seq(from = -6,
                          to = x,
                          by = 0.01)) %>%
  mutate(y = dlogis(x,
                    location = mu,
                    scale = sigma))
```

```
# Plot distribution function and the area under the curve
ggplot(data = df,
       aes(x, y)) +
  geom_area(fill = "orange",
            alpha = 0.5) +
  geom_area(data = df_p,
            fill = "orange",
            alpha = 1) +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0) +
  xlab(expression(paste(V[i] - V[j] - mu))) +
  ylab("f(x)")
```

Try changing the upper limit in the figure above to explore the behavior of the logit probability. What is the probability of choosing j when $x = V_i - V_j = 0$? What is the probability of choosing j when $V_j >> V_i$? And when $V_j << V_i$? Is this as expected? Next, try changing the location parameter μ. What happens when you do this?

The cumulative distribution function is shown in Fig. 4.4. Notice that this function tends asymptotically to 0 when x tends to $-\infty$ and to 1 when x tends to ∞. This function never assigns values of exactly 0 or exactly 1.

```
# Define parameters for the distribution
# Location
mu <- 0
# Scale
sigma <- 1

# Create a data frame for plotting
df <- data.frame(x =seq(from = -5 + mu,
                        to = 5 + mu,
                        by = 0.01)) %>%
  mutate(f = plogis(x,
                    location = mu,
                    scale = 1))

# Plot the cumulative distribution function
logit_plot <- ggplot(data = df,
                     aes(x = x,
                         y = f)) +
  geom_line(color = "orange") +
  ylim(c(0, 1)) +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0)

logit_plot +
  xlab(expression(paste(V[i] - V[j] - mu))) +
  ylab(expression(paste(P[i])))
```

What happens when you change the value of the location parameter in the code used to create Fig. 4.4?

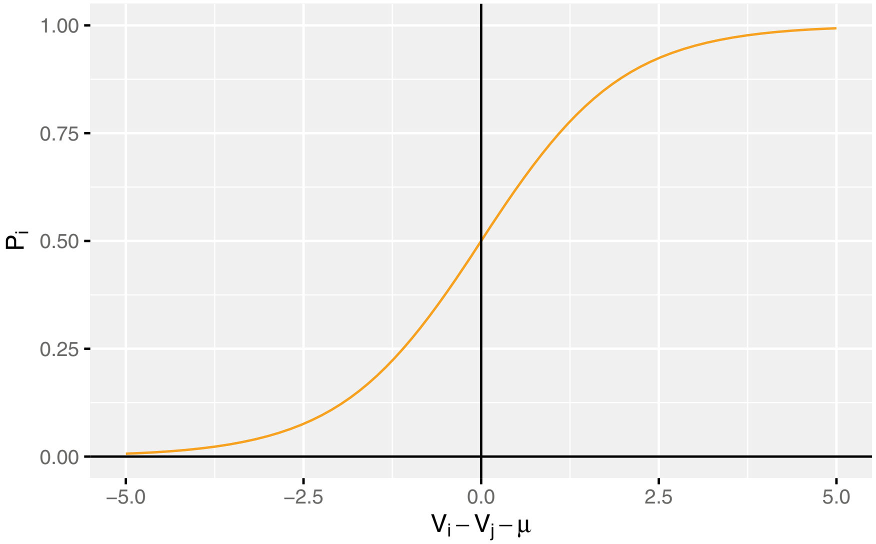


Fig. 4.4 Linear cumulative distribution function

The logit probability exhibits a shape usually called a *sigmoid* for its resemblance to the letter “s”. This shape is shared by most other discrete choice models—the uniform distribution in Chap. 2, for instance, resembled an angular letter “s”, whereas the linear and quadratic distribution functions started to display the non-linear aspect of the logit probability function. Sigmoid functions are of interest in many fields. The study of technology adoption is a case in point; new technologies are initially adopted slowly, then go through a rapid growth stage, before reaching saturation. Population growth is often represented by similar curves, with population growing slowly, then explosively, before reaching a carrying capacity limit.

In the case of discrete choice analysis, the shape of the function is interesting from a policy perspective. In the vast majority of cities in North America, for example, two main modes of transportation are cars and transit. However, the shares of transit tend to be very low, sometimes lower than 10% or even 5%. This suggests that the underlying probabilities of choosing transit at the individual level are very low too.

Suppose that the logit curve in Fig. 4.4 is for the probability of choosing transit. If the initial probability of choosing transit is low, even large increases in the utility of transit result in relatively modest gains in probability (see solid blue line in Fig. 4.5). If the starting probability of transit had been instead 0.5, an identical increase in the utility of transit would result in a much larger gain in the probability (see dashed red line in Fig. 4.5).

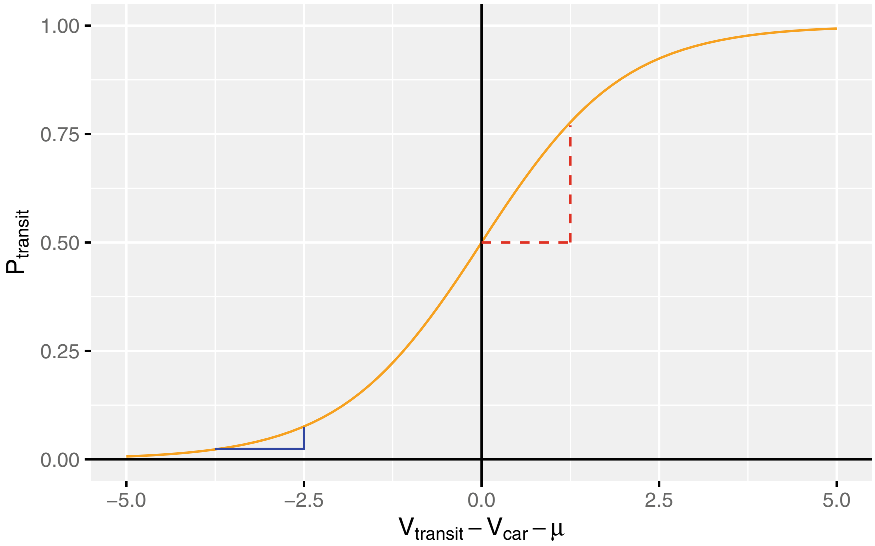


Fig. 4.5 Implication of the sigmoid shape

```
logit_plot +
  xlab(expression(paste(V[transit] - V[car] - mu))) +
  ylab(expression(paste(P[transit]))) +
  annotate("segment",
           x = -3.75, xend = -2.5,
           y = 0.024, yend = 0.024,
           colour = "blue",
           linetype = "solid") +
  annotate("segment",
           x = -2.5, xend = -2.5,
           y = 0.024, yend = 0.075,
           colour = "blue",
           linetype = "solid") +
  annotate("segment",
           x = 0, xend = 1.25,
           y = 0.5, yend = 0.5,
           colour = "red",
           linetype = "dashed") +
  annotate("segment",
           x = 1.25, xend = 1.25,
           y = 0.5, yend = 0.77,
           colour = "red",
           linetype = "dashed")
```

The implication is that when the penetration of an alternative (think transit, hybrid vehicles, clean energy, and other new technologies) is still low, the incentives needed to raise the probabilities need to be very strong even for modest gains. When penetration has increased, the incentives may be eased since their impact is now more than proportional, until reaching saturation, where again large gains in utility result in modest increases in the probability of adoption.

4.8 Now, About Those Parameters μ and σ . . .

Figure 4.3 above was created assuming that $\mu = 0$ and $\sigma = 1$. Can we really set these values in such an arbitrary fashion? The answer is no and yes.

In the case of the centering parameter μ, setting it arbitrarily to zero is not appropriate. If you experimented changing this parameter, you would have noticed that the distribution shifts on the x-axis. When $\mu = 0$, the distribution is centered at zero and $P_i = 0.5$ when $V_i = V_j$. When $\mu \neq 0$, P_i when $V_i = V_j$ is *not* 0.5. What does this mean?

Recall that the probability depends on the difference between the systematic utilities of the alternatives. As seen above, the logit probability is

$$P_i = P(\epsilon_n < V_i - V_j) = \frac{1}{1 + e^{-(V_i - V_j - \mu)/\sigma}}$$

Assume that we let one of the utility functions absorb μ, that is we let

$$V_j^* = V_j + \mu$$

or we let

$$V_i^* = V_i - \mu$$

It does not really matter which utility function we choose to absorb μ (the only thing that changes is the sign). For convenience, we will say that it is V_j, in which case the logit probability can be written as

$$P_i = P(\epsilon_n < V_i - V_j) = \frac{1}{1 + e^{-(V_i - V_j^*)/\sigma}}$$

The difference in other words depends on the value of μ. When μ is a large positive number, the effect is to increase the utility of alternative j (or conversely, since it would enter with a negative sign in V_j^*, it would decrease the utility of alternative i). When μ is a large negative number, the effect is to *increase* the utility of i (since in this case $V_i^* = V_i - (-\mu)$)—or alternatively to reduce the utility of j. For this reason we do not want to arbitrarily set the value of μ to zero, because this parameter contains information about the relative *systematic* differences between V_i and V_j. The utility function that does not contain the centering parameter μ is called the *reference* function.

For simplicity of presentation, we will drop the notation V^* and will assume henceforth that one of the utility functions has absorbed parameter μ.

Now, with respect to the scale parameter σ, this parameter is common to the two utility functions in the logit probability and, as it turns out, it *can* be arbitrarily set to one. Consider two utility functions as follows:

$$V_i - V_j$$

Multiplying (alternatively dividing) by a constant greater than zero changes the *magnitude* of their difference, since:

$$\theta(V_i - V_j) = \theta V_i - \theta V_j$$

In other words, multiplying two quantities by a positive constant changes the cardinality of the difference. If you are working with the R Notebook, you might want to try changing the value of `theta` below, keeping in mind that the value must be **greater than zero** (the dispersion parameter is never negative):

```r
V_j <- -4
V_k <- 8
theta <- 0.8

theta * V_j - theta * V_k
```

```
[1] -9.6
```

You will notice that the difference changes as you change the value of `theta`. But what about the sign?

Clearly, multiplying two quantities by a positive constant does not affect their *ordinality*. That is, if $V_i > V_j$ then it is always true that $\theta V_i > \theta V_j$. Recall the decision-making rule: an alternative is chosen if its utility is greater than that of the competing alternatives. The rule is purely ordinal, it does not matter if the difference between them is small or large—in other words, their cardinality is irrelevant. This is convenient because it allows us to simplify the logit probability as follows, by arbitrarily setting $\sigma = 1$:

$$P_i = P(\epsilon_n < V_i - V_j) = \frac{1}{1 + e^{-(V_i - V_j)}} = \frac{e^{V_i}}{e^{V_i} + e^{V_j}}$$

P_i above is the logit probability of choosing alternative i. In the binomial case, the probability of choosing j is simply:

$$P_j = 1 - P_i.$$

4.9 Multinomial Logit

The logit model in the preceding section was derived assuming a choice set with only two alternatives. This, of course, is very restrictive, and there are many situations where more than two alternatives are of interest. In general, we can express a multinomial choice situation as a case where the number of alternatives $J > 2$. Fortunately, a multinomial version of the logit model can be derived without much difficulty, and it also results in a closed-form expression, as follows:

$$P_i = \frac{e^{V_i}}{\sum_j^J e^{V_j}}$$

Notice that in this case there are $J - 1$ parameters μ that are absorbed by all but one of the utility functions. As before, it does not matter which utility is selected to act as the reference, since the signs (and magnitudes) of the centering parameters adjust accordingly. More on this later.

4.10 Properties of the Logit Model

The logit model is the workhorse of discrete choice analysis, in good measure because its closed form does not require numerical evaluation of the integrals involved in calculating probabilities (i.e., the "area under the curve", although in multinomial situations this actually is a volume under the surface!)

One important property of the logit model is the way it handles substitution patterns. Consider the ratio of odds for any two alternatives according to the multinomial logit model:

$$\frac{P_i}{P_k} = \frac{\frac{e^{V_i}}{\sum_j e^{V_j}}}{\frac{e^{V_k}}{\sum_j e^{V_j}}} = \frac{e^{V_i}}{e^{V_k}} = e^{V_i - V_k}$$

As seen above, the ratio of the odds of P_i to P_k depends only on the difference in the utilities of alternatives i and k but none of the other alternatives in the choice set. Furthermore, recall that the choice set is by design an exhaustive set of possible alternatives, and therefore the sum of the probabilities over this set is one:

$$P_1 + P_2 + \cdots + P_J = 1$$

The above means that if the probability of choosing one alternative, say j, increases, then the probabilities of choosing any of the other alternatives must decline. But since the ratio of odds for any two alternatives is independent of other alternatives in the choice set, the way the probabilities change depends on the change on the probability that triggered the adjustments. This property is called, quite fittingly, *independence from irrelevant alternatives* or IIA.

Suppose, for instance, that a choice set consists of three alternatives products, say margarine (m) by Naturally, and salted butter (sb) and low-sodium butter (lb) by Happy Farms. The initial probabilities of choosing these alternatives are as follows:

$$\begin{cases} P_m^0 = \frac{1}{3} \\ P_{sb}^0 = \frac{1}{3} \\ P_{lb}^0 = \frac{1}{3} \end{cases}$$

Next, suppose that a change in the attribute set of salted butter (sb), for instance a reduction in price, leads to an increase in the probability of choosing this product. Now the probability of choosing salted butter is

$$P_{sb}^1 = \frac{1}{2}$$

How do the other probabilities change? On the one hand, we know that the sum of the new probabilities must be one:

$$P_m^1 + P_{sb}^1 + P_{lb}^1 = 1$$

Since the attributes of margarine and low-sodium butter did not change, we know that their utilities remain unchanged, and therefore:

$$\frac{P_m^1}{P_{lb}^1} = \frac{\frac{1}{3}}{\frac{1}{3}} = 1$$

In other words, the probability of $P_m^1 = P_{lb}^1$. Substituting:

$$P_m^1 + P_{sb}^1 + P_{lb}^1 = 2P_m^1 + P_{sb}^1 = 1$$

Solving for $P_l^1 b$:

$$P_m^1 = \frac{1 - P_{sb}^1}{2} = \frac{1 - \frac{1}{2}}{2} = \frac{1}{4}$$

Therefore the new probabilities are:

$$\begin{cases} P_m^1 = \frac{1}{4} \\ P_{sb}^1 = \frac{1}{2} \\ P_{lb}^1 = \frac{1}{4} \end{cases}$$

Notice that the increase in probability of choosing salted butter draws proportionally from the other alternatives (i.e., low-sodium butter and margarine)—in fact, 12.5% from each. Does this result make sense? What is now the market share of Happy Farms-brand line of butter?

The property of Independence from Irrelevant Alternatives leads to proportional substitution patterns. Consider the following initial probabilities:

$$\begin{cases} P^0_m = 0.5 \\ P^0_{sb} = 0.3 \\ P^0_{lb} = 0.2 \end{cases}$$

The new probability of sb changes to $P^1_{sb} = 0.5$. Following the same logic:

$$\frac{P^1_m}{P^1_{lb}} = \frac{0.5}{0.2} = \frac{5}{2}$$

And:

$$P^1_m = \frac{5}{7}(1 - P^1_{sb}) = \left(\frac{5}{7}\right)\left(\frac{1}{2}\right) = \frac{5}{14} = 0.3571$$

So the final probabilities are:

$$\begin{cases} P^1_m = \frac{5}{14} = 0.3571 \\ P^1_{sb} = \frac{1}{2} = 0.5000 \\ P^1_{lb} = \frac{2}{14} = 0.1429 \end{cases}$$

Now, the increase in $P1_{sb}$ to 1/2 from $P^0_{sb} = 1/5$ is drawing *more* from P^0_m than from P^0_{lb}. However, the pattern of substitution is still proportional, as it can be verified:

$$\frac{P^1_m}{P^0_m} = \frac{\frac{5}{14}}{\frac{1}{2}} = \frac{10}{14}$$
$$\frac{P^1_{lb}}{P^0_{lb}} = \frac{\frac{2}{14}}{\frac{2}{10}} = \frac{10}{14}$$

Proportional substitution patterns are a consequence of the lack of correlation among the random utilities. The logit model considers that the alternatives are all independent. However, in this example, this condition is suspect: the two kinds of butter are more similar between them than either are to margarine. Indeed, if consumers choose butter for flavor, lowering the price of one kind is likely to draw *less* than proportionally from the probability of choosing margarine—and more than proportionally from the probability of the other kind of butter.

In this case, the correlation between the two kinds of butter is a consequence of a missing attribute—say flavor, or health, that is necessary to discriminate among the alternatives. In this way, the logit model can be seen as the ideal model—its closed form is a very attractive feature…as long as the systematic utilities are properly and completely specified. When this is not the case, the results can lead to unrealistic and even unreasonable substitution patterns. This issue suggests two possible courses of action:

1. Working to ensure that the systematic utility functions are properly and completely specified.
2. Modifying the modeling apparatus to accommodate correlations among the random utilities.

As will become clear in later chapters, much work in the field of discrete choice analysis has been concerned with the latter.

4.11 Revisiting the Systematic Utilities

Much of the discussion above has concentrated on the random utility; however, specifying the systematic utility is key.

Recall that the utility is a function of the attributes of the alternatives and possibly the attributes of the decision-makers to allow the model to capture heterogeneity in decision-making styles by individuals. The utility function is a convenient way of summarizing all those attributes. Think again of the example of purchasing a new phone (see Chap. 2), where the alternatives where "do-nothing" (N) or "new phone" (B). In that simple example, the utilities for decision-maker n were a function of three attributes, namely cost, speed, and income—to which we can add the random utility:

$$
\begin{aligned}
U_{n,\mathrm{N}} &= U(\mathrm{cost_N},\ \mathrm{speed_N},\ \mathrm{income}_n) = V(\mathrm{cost_N},\ \mathrm{speed_N},\ \mathrm{income}_n) + \epsilon_{n,\mathrm{N}}\\
U_{n,\mathrm{B}} &= U(\mathrm{cost_B},\ \mathrm{speed_B},\ \mathrm{income}_n) = V(\mathrm{cost_B},\ \mathrm{speed_B},\ \mathrm{income}_n) + \epsilon_{n,\mathrm{B}}
\end{aligned}
$$

A common way of specifying the systematic utility is as linear-in-parameters, something that will be familiar to users of regression analysis:

$$
\begin{aligned}
V(\mathrm{cost_N},\ \mathrm{speed_N},\ \mathrm{income}_n) &= \beta_1 \mathrm{cost_N} + \beta_2\ \mathrm{speed_N} + \beta_3\ \mathrm{income}_n\\
V(\mathrm{cost_B},\ \mathrm{speed_B},\ \mathrm{income}_n) &= \mu + \beta_1 \mathrm{cost_B} + \beta_2\ \mathrm{speed_B} + \beta_3\ \mathrm{income}_n
\end{aligned}
$$

Notice how the location parameter of the logistic function is absorbed by one of the utility functions!

The additive form of the utilities reflects a compensatory choice-making strategy: higher costs may be offset by higher speeds, for example. An important consideration is the way attributes enter the utility functions. Recall that one way of writing the logit probability was

$$
P_i = \frac{1}{1 + e^{-(V_i - V_j)}}
$$

This formulation makes it clear that the probability is a function of the differences between utilities (this remains true in the multinomial logit, even if it is not as clear to see). Now consider what happens when the differences in utility are calculated:

$$\begin{aligned} V_{n,\mathrm{N}} - V_{\mathrm{B}} &= \beta_1 \mathrm{cost}_{\mathrm{N}} + \beta_2 \ \mathrm{speed}_{\mathrm{N}} + \beta_3 \mathrm{income}_n - \mu - \beta_1 \mathrm{cost}_{\mathrm{B}} - \beta_1 \mathrm{speed}_{\mathrm{B}} - \beta_1 \mathrm{income}_n \\ &= \beta_1 (\mathrm{cost}_{\mathrm{N}} - \mathrm{cost}_{\mathrm{B}}) + \beta_2 (\ \mathrm{speed}_{\mathrm{N}} - \ \mathrm{speed}_{\mathrm{B}}) + \beta_3 (\ \mathrm{income}_n - \ \mathrm{income}_n) - \mu \end{aligned}$$

The income attribute vanishes!

It is useful to distinguish between attributes that vary across utility functions and those that do not. Level of service attributes, those that describe the alternatives, generally vary by utility function—indeed, it is those attributes that help a decision-maker discriminate between alternatives. In this instance, income is invariant across utility functions. Personal attributes of the decision-makers, in general, are invariant across utility functions.

The most common way of dealing with attributes that are constant across utility functions is to select one utility to act as a reference and set that attribute to zero there. This is illustrated below

$$V_{\mathrm{B}} = \mu + \beta_1 \mathrm{cost}_{\mathrm{B}} + \beta_2 \ \mathrm{speed}_{\mathrm{B}} + \beta_3 \ \mathrm{income}_n$$

The difference in utilities then becomes

$$V_{\mathrm{N}} - V_{\mathrm{B}} = \beta_1 (\mathrm{cost}_{\mathrm{N}} - \mathrm{cost}_{\mathrm{B}}) + \beta_2 (\ \mathrm{speed}_{\mathrm{N}} - \ \mathrm{speed}_{\mathrm{B}}) - \beta_3 (\ \mathrm{income}_n) - \mu$$

When the effect of income is positive (i.e., $\beta_3 > 0$) higher incomes reduce the probability of doing nothing, and when the effect of income is negative (i.e., $\beta_3 < 0$) higher incomes reduce the probability of buying a new phone. The effect of income is relative to the reference alternative. When there are more than two alternatives, the attribute can be entered in all but the reference utility, as shown next:

$$\begin{array}{lllllll} V_{\mathrm{N}} & = 0 & +0 & +\beta_1 \mathrm{cost}_{\mathrm{N}} & +\beta_2 \ \mathrm{speed}_{\mathrm{N}} & +\beta_3 (0) & +\beta_4 (0) \\ V_{\mathrm{uPhone}} & = \mu_{\mathrm{uPhone}} & +0 & +\beta_1 \mathrm{cost}_{\mathrm{uPhone}} & +\beta_2 \ \mathrm{speed}_{\mathrm{uPhone}} & +\beta_3 \ \mathrm{income}_n & +\beta_4 (0) \\ V_{\mathrm{zPhone}} & = 0 & +\mu_{\mathrm{zPhone}} & +\beta_1 \mathrm{cost}_{\mathrm{zPhone}} & +\beta_2 \ \mathrm{speed}_{\mathrm{zPhone}} & +\beta_3 (0) & +\beta_4 \ \mathrm{income}_n \end{array}$$

The above also illustrates how location parameters are absorbed by $J - 1$ utility functions.

Another way to introduce attributes that do not vary across utility functions is reminiscent of Casetti's expansion method (Casetti 1972). The expansion method is a systematic approach to introduce variable interactions that proceeds by defining an initial model whose coefficients are subsequently expanded using contextual variables. Suppose that the initial model is comprised of the utility functions with only level of service variables:

$$V_{\mathrm{N}} = 0 + \beta_1 \mathrm{cost}_{\mathrm{N}} + \beta_2 \ \mathrm{speed}_{\mathrm{N}}$$
$$V_{\mathrm{B}} = \mu + \beta_1 \mathrm{cost}_{\mathrm{B}} + \beta_2 \ \mathrm{speed}_{\mathrm{B}}$$

The coefficients are expanded by a contextual variable, in this case income:

$$\beta_1 = \beta_{11} + \beta_{12}\mathrm{income}_n \beta_2 = \beta_{21} + \beta_{22}\mathrm{income}_n$$

Substituting the expanded coefficients in the initial model:

$$V_{\mathrm{N}} = 0 + (\beta_{11} + \beta_{12}\mathrm{income}_n)\mathrm{cost}_{\mathrm{N}} + (\beta_{21} + \beta_{22}\mathrm{income}_n) \ \mathrm{speed}_{\mathrm{N}}$$
$$V_{\mathrm{B}} = \mu + (\beta_{11} + \beta_{12}\mathrm{income}_n)\mathrm{cost}_{\mathrm{B}} + (\beta_{21} + \beta_{22}\mathrm{income}_n) \ \mathrm{speed}_{\mathrm{B}}$$

The expanded model then becomes

$$V_{\mathrm{N}} = 0 + \beta_{11}\mathrm{cost}_{\mathrm{N}} + \beta_{12}\mathrm{income}_n \cdot \mathrm{cost}_{\mathrm{N}} + \beta_{21} \ \mathrm{speed}_{\mathrm{N}} + \beta_{22}\mathrm{income}_n \cdot \ \mathrm{speed}_{\mathrm{N}}$$
$$V_{\mathrm{B}} = \mu + \beta_{11}\mathrm{cost}_{\mathrm{B}} + \beta_{12}\mathrm{income}_n \cdot \mathrm{cost}_{\mathrm{B}} + \beta_{21} \ \mathrm{speed}_{\mathrm{B}} + \beta_{22}\mathrm{income}_n \cdot \ \mathrm{speed}_{\mathrm{B}}$$

The difference of the two utilities in turn is

$$\begin{aligned} V_{\mathrm{N}} - V_{\mathrm{B}} = & \\ & \beta_{11}(\mathrm{cost}_{\mathrm{N}} - \mathrm{cost}_{\mathrm{B}}) + \beta_{12}\mathrm{income}_n \cdot (\mathrm{cost}_{\mathrm{N}} - \mathrm{cost}_{\mathrm{B}}) \\ & + \beta_{21}(\ \mathrm{speed}_{\mathrm{N}} - \ \mathrm{speed}_{\mathrm{B}}) + \beta_{22}\mathrm{income}_n \cdot (\ \mathrm{speed}_{\mathrm{N}} - \ \mathrm{speed}_{\mathrm{B}}) - \mu \end{aligned}$$

Specifying the utility functions is more art than technique. We will return to this issue when we begin the practice of model estimation in Chap. 5.

4.12 Exercises

1. What do we mean when we say that the logit probability has a closed form?
2. Why is it that we can set the dispersion parameter in the logit probabilities to one?
3. Suppose that a choice set consists of two alternatives, travel by car (c) and travel by blue bus (bb). The utilities of these two modes are the same, that is

   $$V_c = V_{bb}$$

   What are the probabilities of choosing these two modes?
4. Suppose that the transit operator of the blue buses in Question 3 decides to introduce a new service, namely a red bus. This red bus is identical to the blue bus in every respect except the color. Under these new conditions, what are the logit probabilities of choosing these modes?
5. Discuss the results of introducing a new mode in the choice process above.

References

Ben-Akiva, M., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and applications to travel demand*. Cambridge: The MIT Press.

Casetti, E. (1972). Generating models by the expansion method: Applications to geographic research. *Geographical Analysis, 4*(1), 81–91.

Hensher, D. A., Rose, J. M., & Greene, W. H. (2005). *Applied choice analysis: A primer*. Cambridge University Press.

Louviere, J. J., Hensher, D. A., & Swait, J. D. (2000). *Stated choice methods: Analysis and applications*. Cambridge University Press.

Ortúzar, J. D., & Willumsen, L. G. (2011). *Modelling transport* (4th ed.). New York: Wiley.

Train, K. (2009). *Discrete choice methods with simulation* (2nd ed.). Cambridge: Cambridge University Press.

Chapter 5
Practical Issues in the Specification and Estimation of Discrete Choice Models

In theory, there is no difference between theory and practice. But in practice, there is.

— Benjamin Brewster

An ounce of practice is generally worth more than a ton of theory.

— E.F. Schumacher

5.1 Theory and Practice

In Chap. 3, we introduced a conceptual framework (a theory) to analyze individual decision-making behavior. This was followed in Chap. 4 by the necessary apparatus (based on probability theory) to implement the conceptual framework to analyze discrete choices. The two preceding chapters provide the conceptual and technical foundations of discrete choice analysis, and together offer an intuitive and elegant framework to study decision-making (and a powerful one too; Daniel McFadden was awarded the Sveriges Riksbank Prize in Economic Sciences (Nobel Prize) for his contributions to random utility modeling).

Although not described in detail in previous chapters, it is worthwhile to dwell for a moment on the history of the development of the logit model as a random utility model.

In his Nobel Lecture, McFadden (2001) recounts the path that led to the development of random utility models for discrete choices. Like most important discoveries, it is a meandering path. It began early in the twentieth century with a theory for economic behavior (i.e., utility) that considered heterogeneous preferences (remember, utility functions are specific to individuals) that in practice were difficult to estimate

A. Páez and G. Boisjoly, *Discrete Choice Analysis with R*, Use R!,
https://doi.org/10.1007/978-3-031-20719-8_5

empirically because of data limitations. Indeed, studies before the 1960s mostly considered aggregated demand with representative agents (i.e., archetypal consumers) to accommodate this limitation in data availability. It was only when individual-level data became more widely collected and within reach of researchers that it became possible to pay attention to the behavior of individual agents.

While economists were busy with models of aggregated demand, researchers in psychometrics and mathematical psychology, chiefly L.L Thurstone and R.D. Luce, were occupied providing the technical basis for modeling what Thurstone termed *Comparative Judgment* (in the sense of making a decision or forming an opinion). In particular, Luce introduced the axiom of Independence of Irrelevant Alternatives (discussed in Chap. 4). According to McFadden (2001, 353), this axiom "simplified experimental collection of choice data by allowing multinomial choice probabilities to be inferred from binomial choice experiments." J. Marschak was the first to introduce the work of Thurstone to econometrics in 1960, and also coined the term *Random Utility Maximizing* (RUM) that eventually prevailed over the comparative judgment terminology of Thurstone. McFadden's early contribution to this body of research was to develop an econometric version of Luce's model, with strict (i.e., systematic) utilities specified as functions of the attributes of the alternatives. This allowed researchers to link unobserved preference heterogeneity to a fully consistent description of the distribution of demand. Since the 1970s, discrete choice analysis has been a burgeoning area of research with a plethora of applications in economics, marketing, and travel behavior, among many other disciplines.

This brief story neatly illustrates the complex interplay between theory and practice.

Early attempts to study demand were limited due to practical considerations (i.e., the absence of data at the individual level). Once appropriate data became available, new studies continued to push the theoretical envelope. Indeed, theoretical questions have continued to inspire newer ways to collect data and novel modeling methods, and these in turn have helped us to refine our understanding of behavior. See as an example the work on decision-making in social situations (Akerlof 1997; Axhausen 2005; Páez and Scott 2007) which inspired the use of new data sources (Axhausen 2008), as well as novel modeling approaches (e.g., Dugundji and Walker 2005; Dugundji and Gulyas 2013; Kamargianni et al. 2014) and empirical work (e.g., Berg et al. 2009; Goetzke and Rave 2011; Matous 2017).

Now that the preceding chapters have armed us with the concepts and basic technical elements to implement random utility models, it is proper that we turn our attention to the more practical aspects of modeling. The best way to ensure that the concepts take hold, in our view, is to get our hands on a data set and struggle with the practicalities of cleaning and organizing data, specifying the utility functions (a task that is more art than science), and estimating models. These skills are mostly transferable to other modeling techniques to be introduced later in this book, so we will begin by applying them to the most fundamental of discrete choice models, namely the multinomial logit model.

5.2 How to Use This Chapter

Remember that the source code used in this chapter is available. Throughout the notes, you will find examples of code in segments of text called *chunks*. This is an example of a chunk:

```
print("Hats off to you, Prof. McFadden")
```

```
## [1] "Hats off to you, Prof. McFadden"
```

If you are working with RStudio you can type the chunks of code to experiment with them. As an alternative, you may copy and paste the source code into your `R` or RStudio console, or create a script/notebook to save the code and any experiments you may conduct.

5.3 Learning Objectives

In this chapter, you will learn about:

1. Specification of utility functions.
2. Maximum likelihood estimation.
3. Estimation of multinomial logit models.
4. McFadden's ρ^2.
5. The likelihood ratio test.

5.4 Suggested Readings

- Ben-Akiva, M., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and applications to travel demand*, **Chapters 4 and 5**. MIT Press.
- Hensher, D. A., Rose, J. M., & Greene, W. H. (2005). *Applied choice analysis: A primer*, **Chapter 10**. Cambridge University Press.
- Ortuzar, J. D., & Willumsen, L. G. (2011). Modelling transport, 4th ed., **Chapter 8**. Wiley.
- Train, K. (2009). Discrete choice methods with simulation, 2nd ed., **Chapter 3**. Cambridge University Press.

5.5 Preliminaries

Load the packages used in this section:

```r
library(discrtr) # A companion package for the book Introduction to Discrete Choice Analysis with `R`
library(dplyr) # A Grammar of Data Manipulation
library(ggplot2) # Create Elegant Data Visualisations Using the Grammar of Graphics
library(htmlwidgets) # HTML Widgets for R
library(kableExtra) # Construct Complex Table with 'kable' and Pipe Syntax
library(mlogit) # Multinomial Logit Models
library(plotly) # Create Interactive Web Graphics via 'plotly.js'
library(stargazer) # Well-Formatted Regression and Summary Statistics Tables
library(tidyr) # Tidy Messy Data
library(webshot2) # Take Screenshots of Web Pages
```

Load the data set used in this section:

```r
data("mc_commute_wide",
     package = "discrtr")
```

5.6 The Anatomy of Utility Functions

At the end Chap. 4 we took, for the first time, a closer look at the systematic utilities of discrete choice models. It is useful to think about the systematic utility functions in terms of their typical anatomy. As you will recall, we said that some variables vary across utility functions; these are typically the attributes that describe the various alternatives (e.g., their level of service and cost). The variables that describe the decision-maker do *not* vary by alternative. This has implications, as we saw in the preceding chapter, for how the variables enter the functions. Since the model works on the basis of *differences* between utilities, the attributes must actually measure different levels of something or else they vanish.

We can describe the utilities in terms of the way different variables are introduced in the utility functions. As before, we will assume that the location parameters of the distribution are absorbed by $J - 1$ utility functions (where J is the number of alternatives) in the form of alternative-specific constants.

Consider first variables that vary across alternatives for individual n. These variables can have a generic coefficient or they can have alternative-specific coefficients, as seen here:

$$\begin{array}{l} V_{n1} = \\ V_{n2} = \\ V_{n3} = \end{array}
\overbrace{\begin{array}{ll} 0 & +0 \\ \mu_2 & +0 \\ 0 & +\mu_3 \end{array}}^{\text{alternative-specific constants}}
\underbrace{\begin{array}{l} +\beta_1 x_{n1} \\ +\beta_1 x_{n2} \\ +\beta_1 x_{n3} \end{array}}_{\text{alternative vars. with generic coefficients}}
\overbrace{\begin{array}{lll} +\delta_1 w_{n1} & +0 & +0 \\ +0 & +\delta_2 w_{n2} & +0 \\ +0 & +0 & +\delta_3 w_{n3} \end{array}}^{\text{alternative vars. with specific coefficients}}$$

In many cases it makes sense to use generic coefficients. For instance, if the variable is cost, we might assume that one dollar is valued equally irrespective of which alternative it is spent on. In other cases, the use of alternative-specific coefficients might be informative. For instance, it is possible that time spent driving is seen as more expensive than time traveling by bus (where a traveler is free to do other things). Occasionally, as well, an attribute might be specific to an alternative: for instance, waiting time is often implicitly zero for travel by car and active modes of transportation (i.e., walking and cycling).

The differences of the utilities are as follows:

$$\begin{aligned}
V_{n2} - V_{n1} &= (\mu_2 - 0) &&+ \beta_1(x_{n2} - x_{n1}) &&+ (\delta_2 w_{n2} - \delta_1 w_{n1})\\
V_{n3} - V_{n1} &= (\mu_3 - 0) &&+ \beta_1(x_{i3} - x_{n1}) &&+ (\delta_3 w_{n3} - \delta_1 w_{n1})\\
V_{n3} - V_{n2} &= (\mu_3 - \mu_2) &&+ \beta_1(x_{i3} - x_{n2}) &&+ (\delta_3 w_{n3} - \delta_2 w_{n2})
\end{aligned}$$

Variables that vary across individuals but not by alternative can be introduced with alternative-specific coefficients for $J - 1$ alternatives, as follows:

$$\begin{array}{lllllllll}
 & \multicolumn{2}{c}{\overbrace{\hphantom{0\quad +0}}^{\text{alternative-specific constants}}} & & \multicolumn{3}{c}{\overbrace{\hphantom{+\delta_1 w_{n1}\quad +0\quad +0}}^{\text{alternative vars. with specific coefficients}}} & & \\
V_{n1} = & 0 & +0 & +\beta_1 x_{n1} & +\delta_1 w_{n1} & +0 & +0 & +0 & +0\\
V_{n2} = & \mu_2 & +0 & +\beta_1 x_{n2} & +0 & +\delta_2 w_{n2} & +0 & +\gamma_2 z_n & +0\\
V_{n3} = & 0 & +\mu_3 & +\beta_1 x_{n3} & +0 & +0 & +\delta_3 w_{n3} & +0 & +\gamma_3 z_n\\
 & & & \underbrace{\hphantom{+\beta_1 x_{n3}}}_{\text{alternative vars. with generic coefficients}} & & & & \multicolumn{2}{c}{\underbrace{\hphantom{+0\quad +\gamma_3 z_n}}_{\text{individual vars. with specific coefficients}}}
\end{array}$$

This ensures that the individual variables do not vanish. Following the example above, the differences of utilities are

$$\begin{aligned}
V_{n2} - V_{n1} &= (\mu_2 - 0) &&+ \beta_1(x_{n2} - x_{n1}) + (\delta_2 w_{n2} - \delta_1 w_{n1}) + (\gamma_2 - 0) z_n\\
V_{n3} - V_{n1} &= (\mu_3 - 0) &&+ \beta_1(x_{n3} - x_{n1}) + (\delta_3 w_{n3} - \delta_1 w_{n1}) + (\gamma_3 - 0) z_n\\
V_{n3} - V_{n2} &= (\mu_3 - \mu_2) &&+ \beta_1(x_{n3} - x_{n2}) + (\delta_3 w_{n3} - \delta_2 w_{n2}) + (\gamma_3 - \gamma_2) z_n
\end{aligned}$$

A different way of introducing individual-level variables is as part of an expansion of some coefficients, for example:

$$\begin{array}{lllllll}
 & \multicolumn{2}{c}{\overbrace{\hphantom{0\quad +0}}^{\text{alternative specific constants}}} & & \multicolumn{3}{c}{\overbrace{\hphantom{+\delta_1 w_{n1}\quad +0\quad +0}}^{\text{alternative vars. with specific coefficients}}}\\
V_{n1} = & 0 & +0 & +(\beta_{11} + \beta_{12} z_n) x_{n1} & +\delta_1 w_{n1} & +0 & +0\\
V_{n2} = & \mu_2 & +0 & +(\beta_{11} + \beta_{12} z_n) x_{n2} & +0 & +\delta_2 w_{n2} & +0\\
V_{n3} = & 0 & +\mu_3 & +(\beta_{11} + \beta_{12} z_n) x_{n3} & +0 & +0 & +\delta_3 w_{n3}\\
 & & & \underbrace{\hphantom{+(\beta_{11} + \beta_{12} z_n) x_{n3}}}_{\text{alternative vars. with generic coefficients}} & & &
\end{array}$$

The above expands to:

$$
\begin{matrix} V_{n1} = \\ V_{n2} = \\ V_{n3} = \end{matrix}
\quad
\overbrace{\begin{matrix} 0 & +0 \\ \mu_2 & +0 \\ 0 & +\mu_3 \end{matrix}}^{\text{alternative specific constants}}
\quad
\underbrace{\begin{matrix} +\beta_{11}x_{n1} & +\beta_{12}z_nx_{n1} \\ +\beta_{11}x_{n2} & +\beta_{12}z_nx_{n2} \\ +\beta_{11}x_{n3} & +\beta_{12}z_nx_{n3} \end{matrix}}_{\text{alternative vars. with generic coefficients}}
\quad
\overbrace{\begin{matrix} +\delta_1 w_{n1} & +0 & +0 \\ +0 & +\delta_2 w_{n2} & +0 \\ +0 & +0 & +\delta_3 w_{n3} \end{matrix}}^{\text{alternative vars. with specific coefficients}}
$$

And so the differences in utilities are

$$
\begin{aligned}
V_{n2} - V_{n1} &= (\mu_2 - 0) \quad + \beta_{11}(x_{n2} - x_{n1}) + \beta_{12}(z_nx_{n2} - z_nx_{n1}) + (\delta_2 w_{n2} - \delta_1 w_{n1})\\
V_{n3} - V_{n1} &= (\mu_3 - 0) \quad + \beta_{11}(x_{n3} - x_{n1}) + \beta_{12}(z_nx_{n3} - z_nx_{n1}) + (\delta_3 w_{n3} - \delta_1 w_{n1})\\
V_{n3} - V_{n2} &= (\mu_3 - \mu_2) + \beta_{11}(x_{n3} - x_{n2}) + \beta_{12}(z_nx_{n3} - z_nx_{n2}) + (\delta_3 w_{n3} - \delta_2 w_{n2})
\end{aligned}
$$

Understanding the anatomy of utility functions is essential to properly specify and estimate models.

5.7 Example: Specifying the Utility Functions

We are now ready to begin working with a practical example. Although briefly mentioned in question 3 of the exercise in Chap. 2, it is useful to highlight here that data tables for discrete choice analysis can have two shapes: *long* or *wide*.

In a data frame in wide format each row represents one decision-maker, and the information about the alternatives is spread: the same attribute can and often does appear in multiple columns, one for each alternative for which the attribute was measured. A table in *long* format is different in that each row represents an alternative: attributes appear only in one column, but the decision-maker will be repeated in multiple rows, one time for each alternative that they could choose from.

To illustrate the difference, we will use a data set that you encountered before in Chap. 2. This data set contains information about various modes of transportation used by students commuting to McMaster University in Canada (Whalen et al. 2013). The data set was loaded above as part of the preliminaries of this chapter. This table is *wide*. We can see this by displaying the first few rows of the data frame. First we select the columns with the respondents' `id`, their `choice`, and all variables that start with `time`; then we filter a selection of rows by decision-maker `id`:

```
example_wide <- mc_commute_wide %>%
  # Select columns from the table
  select(id, choice, starts_with("time")) %>%
  # Filter three decision-makers by their `id`
  # Here the symbol `|` is for "or", so this reads
  # filter rows with id == 566910139 OR id == 566873140 OR id == 566872636
  filter(id == 566910139 |
           id == 566873140 |
           id == 566872636)

example_wide
```

```
         id choice time.Cycle time.Walk time.HSR time.Car
1 566872636    HSR         NA  21.31439        5       NA
2 566873140    HSR         NA  12.78863       10        2
3 566910139   Walk   4.371118  15.00000       20        5
```

It can be seen that in this wide table the variable `time` (for travel time) appears in four columns. We also notice that this variable was not measured for every single alternative: this is because some users did not have a bicycle or car; for others school might have been too far to walk. When attributes are missing, this is interpreted as a situation where the alternative was not available to the decision-maker. Pay attention and you will see that decision-maker with `id` "566872636" only had two alternatives available to them, walk and HSR (public transportation). Decision-maker with `id` "566873140" had three alternatives available (walk, HSR, car), and decision-maker with `id` "566910139" had the full choice set of four alternatives.

We now use the same selection of variables and rows to see the difference with a table in long format. For this, we reshape the table with `pivot_longer()`:

```
example_wide %>%
  # `pivot_longer()` takes a wide table and makes it long
  # Here we pivot the columns with the `time` variable
  pivot_longer(cols = starts_with("time."),
               # There is a pattern to the names: time.ALTERNATIVE
               # The prefix is the name of the variable
               names_prefix = "time.",
               # The alternatives are placed in a new column called
               # "alternative"
               names_to = "alternative",
               # The values of the variables are consolidated in
               # a single column called "time"
               values_to = "time")
```

```
# A tibble: 12 x 4
          id choice alternative  time
       <dbl> <fct>  <chr>       <dbl>
 1 566872636 HSR    Cycle       NA   
 2 566872636 HSR    Walk        21.3 
 3 566872636 HSR    HSR          5   
 4 566872636 HSR    Car         NA   
 5 566873140 HSR    Cycle       NA   
 6 566873140 HSR    Walk        12.8 
 7 566873140 HSR    HSR         10   
 8 566873140 HSR    Car          2   
 9 566910139 Walk   Cycle        4.37
10 566910139 Walk   Walk        15   
11 566910139 Walk   HSR         20   
12 566910139 Walk   Car          5   
```

Notice how the same id (the identifier of the decision-maker) is repeated in four rows, but time now is a single column.

Package {mlogit}, which we will use to practice, uses *long* tables. Since wide tables are more common, the package includes a utility function to reshape the table. This is used below to convert mc_commute_wide into a long table. For this, we need to indicate which variables are *varying*, meaning that they vary by alternative. In our wide table, there are four variables that vary by alternative: travel time (time), access time (access: the time needed to reach an HSR bus stop), waiting time (wait: time spent waiting for an HSR bus), and number of transfers when using HSR (transfer). The latter three variables are specific to HSR and therefore are set to zero for the modes “Car”, Cycle”, and “Walk”.

```
example_long <- mc_commute_wide %>%
  # Filter three decision-makers by their `id`
  # Here the symbol `|` is for "or", so this reads
  # filter rows with id == 566910139 OR id == 566873140 OR id == 566872636
  filter(id == 566910139 |
           id == 566873140 |
           id == 566872636) %>%
  mlogit.data(shape="wide",
              # Name of column with the choices
              choice = "choice",
              # Numbers of columns with attributes that vary by alternative
              varying = 3:22)
```

The output of the function is an indexed data frame, which is in fact two tables: a main table with the data and a second table with the index. If we examine the main table (as in our preceding example), we see that instead of each row being an individual, each row is an alternative or choice situation:

```
data.frame(example_long) %>%
  # Select columns
  select(id,
         choice,
         alt,
         starts_with("time"),
         idx)
```

```
          id choice    alt      time    idx
1  566872636  FALSE    Car        NA  1:Car
2  566872636  FALSE Cycle         NA 1:ycle
3  566872636   TRUE    HSR  5.000000  1:HSR
4  566872636  FALSE   Walk 21.314387 1:Walk
5  566873140  FALSE    Car  2.000000  2:Car
6  566873140  FALSE Cycle         NA 2:ycle
7  566873140   TRUE    HSR 10.000000  2:HSR
8  566873140  FALSE   Walk 12.788632 2:Walk
9  566910139  FALSE    Car  5.000000  3:Car
```

```
10 566910139  FALSE Cycle  4.371118 3:ycle
11 566910139  FALSE   HSR 20.000000  3:HSR
12 566910139   TRUE  Walk 15.000000 3:Walk
```

Since there are four alternatives in this case, each row corresponds to the choice situation for an alternative for an individual (long tables generally have J rows per decision-maker). Looking jointly at the columns `choice` and `alt` we see that individual with `id` "566872636" chose HSR, individual with `id` "566910139" chose to walk, and so on. This table also contains a new index column (`idx`) that refers to the index table.

We can also look specifically at the index data frame:

```
print(data.frame(mc_commute_long$idx))
```

The index data frame contains two variables. The first one identifies the individual and the second one identifies the alternative.

The first step towards developing a choice model is to specify the utility functions for the desired model. Package {mlogit} uses package {Formula} to create the utility functions. This package creates objects that build upon the {Formula} package for multi-component formulas. As seen above, utility functions can potentially have multiple components, so this functionality is quite useful.

Formulas for use in the {mlogit} package are defined using three parts:

choice $\sim$ alternative-specific vars with generic coefficients |
individual-specific vars |
alternative-specific vars with specific coefficients

If we list all columns in the data frame, we can see what variables are available for this analysis:

```
colnames(example_long)
```

```
 [1] "id"                       "choice"
 [3] "parking"                  "vehind"
 [5] "gender"                   "age"
 [7] "shared"                   "family"
 [9] "child"                    "street_density"
[11] "sidewalk_density"         "LAT"
[13] "LONG"                     "PersonalVehComf_SD"
[15] "PersonalVehComf_D"        "PersonalVehComf_A"
[17] "PersonalVehComf_SA"       "Fun_SD"
[19] "Fun_D"                    "Fun_A"
[21] "Fun_SA"                   "ActiveNeigh_SD"
[23] "ActiveNeigh_D"            "ActiveNeigh_A"
[25] "ActiveNeigh_SA"           "UsefulTrans_SD"
```

```
[27] "UsefulTrans_D"             "UsefulTrans_A"
[29] "UsefulTrans_SA"            "BusComf_SD"
[31] "BusComf_D"                 "BusComf_A"
[33] "BusComf_SA"                "TravelAlone_SD"
[35] "TravelAlone_D"             "TravelAlone_A"
[37] "TravelAlone_SA"            "Shelters_SD"
[39] "Shelters_D"                "Shelters_A"
[41] "Shelters_SA"               "Community_SD"
[43] "Community_D"               "Community_A"
[45] "Community_SA"              "personal_veh_comfortable"
[47] "getting_there_fun"         "like_active_neighborhood"
[49] "commute_useful_transition" "buses_comfortable"
[51] "prefer_travel_alone"       "shelter_good_quality"
[53] "sense_community"           "numna"
[55] "alt"                       "available"
[57] "time"                      "access"
[59] "wait"                      "transfer"
[61] "chid"                      "idx"
```

The source table is documented and the variable definitions can be consulted like so:

```
?mc_commute_wide
```

We see that besides identifier variables `id` and `chid`, and the variable for `choice`, there are several variables that are specific to individual decision-makers. These are `parking` (availability of a parking pass), `vehind` (whether the decision-maker had individual access to a private vehicle), `gender`, `age`, `shared` (living in shared accommodations away from the family home), `family` (living at the family home), and `child` (respondent was responsible for at least one minor in the household). Furthermore, some variables relate to the physical environment of the place of residence (`street_density` and `sidewalk_density`), in addition to the coordinates of the place of residence (geocoded to the nearest major intersection or postal code centroid). These variables are also considered individual specific, as they relate to their home location. One variable is alternative specific, namely `time` (travel time in minutes). And, as noted before, three variables are specific to public transportation, namely `access` (access time to public transportation in minutes), `wait` (waiting time in minutes), and `transfer` (number of transfers when traveling by public transportation).

We will begin by defining a very simple formula that considers only travel time. We will save this object as `f1`:

```
# Function `mFormula()` is used to define multi-part formulas of the form:
# y ~ x | z | w, which in the notation used for the anatomy of utility functions is
# choice ~ alternative vars. with generic coefficients |
#          individual vars. with specific coefficients |
#          alternative vars. with specific coefficients
# In this formula time is one of x variables
f1 <- mFormula(choice ~ time)
```

The function `model.matrix` can be used to examine how the formula is applied to the data (we first convert the model matrix into a data frame and then use the head function to display the top rows):

```
# Pipe `f1` to next function
f1 %>%
  # Build the model matrix with data set `example_long`
  model.matrix(example_long)
```

```
  (Intercept):Cycle (Intercept):HSR (Intercept):Walk      time
1                 0               1               0  5.000000
2                 0               0               1 21.314387
3                 0               0               0  2.000000
4                 0               1               0 10.000000
5                 0               0               1 12.788632
6                 0               0               0  5.000000
7                 1               0               0  4.371118
8                 0               1               0 20.000000
9                 0               0               1 15.000000
```

Recall that the first decision-maker in this subset of the table only had two alternatives in their choice set: HSR and walk; accordingly the first two rows correspond to them. The model matrix does not include alternatives that were not part of the choice set. Furthermore, we see that the formula includes by default the alternative-specific coefficients. The utility functions of this decision-maker can be written as:

$$\begin{array}{rlll} V_{HSR} = & \overbrace{0 + \mu_{HSR} + 0 \phantom{+\mu_{walk}}}^{\text{alternative-specific constants}} & & + \beta_1 \text{time}_{HSR} \\ V_{walk} = & 0 + 0 \quad + \mu_{walk} & & \underbrace{+ \beta_1 \text{time}_{walk}}_{\text{alternative vars. with generic coefficients}} \end{array}$$

Compare these utility functions to the model matrix.

The second individual in the table had access to three alternatives, so there are three rows for them. The third and last individual in the reduced table had all four alternatives available, so the last four rows correspond to them. Their utility functions are as follows:

$$
\begin{array}{rclllll}
 & & \multicolumn{3}{c}{\overbrace{\hphantom{0 \; +\mu_{cycle} \; +\mu_{walk}}}^{\text{alternative-specific constants}}} & & \\
V_{car} & = & 0 & +0 & +0 & & +\beta_1 \text{time}_{car} \\
V_{cycle} & = & 0 & +\mu_{cycle} & +0 & & +\beta_1 \text{time}_{cycle} \\
V_{HSR} & = & 0 & +\mu_{HSR} & +0 & & +\beta_1 \text{time}_{HSR} \\
V_{walk} & = & 0 & +0 & +\mu_{walk} & & \underbrace{+\beta_1 \text{time}_{walk}}_{\text{alternative vars. with generic coefficients}}
\end{array}
$$

Let us define now a formula with an individual-specific variable, say sidewalk density at the place of residence, and call it `f2`:

```
# Function `mFormula()` is used to define multi-part formulas of the form:
# y ~ x | z | w, which in the notation used for the anatomy of utility functions is
# choice ~ alternative vars. with generic coefficients |
#          individual vars. with specific coefficients |
#          alternative vars. with specific coefficients
# In this formula `time` is one of x variables and `sidewalk_density` is one of z variables
f2 <- mFormula(choice ~ time | sidewalk_density)
```

The model matrix is now:

```
# Pipe `f2` to next function
f2 %>%
  # Build the model matrix with data set `example_long`
  model.matrix(example_long)
```

```
  (Intercept):Cycle (Intercept):HSR (Intercept):Walk      time
1                 0               1               0  5.000000
2                 0               0               1 21.314387
3                 0               0               0  2.000000
4                 0               1               0 10.000000
5                 0               0               1 12.788632
6                 0               0               0  5.000000
7                 1               0               0  4.371118
8                 0               1               0 20.000000
9                 0               0               1 15.000000
  sidewalk_density:Cycle sidewalk_density:HSR sidewalk_density:Walk
1                0.00000             22.63322               0.00000
2                0.00000              0.00000              22.63322
3                0.00000              0.00000               0.00000
4                0.00000             39.64003               0.00000
5                0.00000              0.00000              39.64003
6                0.00000              0.00000               0.00000
7               26.06793              0.00000               0.00000
8                0.00000             26.06793               0.00000
9                0.00000              0.00000              26.06793
```

The utility functions for the full choice set are as follows:

$$
\begin{array}{llllllll}
 & & \multicolumn{3}{c}{\overbrace{\hphantom{= 0 \quad +0 \quad +0}}^{\text{alternative specific constants}}} & & \multicolumn{2}{c}{\overbrace{\hphantom{0 \quad +0 \quad +0}}^{\text{individual vars with specific coefficients}}} \\
V_{n,car} & = 0 & +0 & +0 & +\beta_1 \mathrm{time}_{n,car} & 0 & +0 & +0 \\
V_{n,cycle} & = \mu_{\mathrm{cycle}} & +0 & +0 & +\beta_1 \mathrm{time}_{n,cycle} & \gamma_1 \mathrm{swd}_n & +0 & +0 \\
V_{n,HSR} & = 0 & +\mu_{\mathrm{HSR}} & +0 & +\beta_1 \mathrm{time}_{n,HSR} & 0 & +\gamma_2 \mathrm{swd}_n & +0 \\
V_{n,walk} & = 0 & +0 & +\mu_{\mathrm{walk}} & \underbrace{+\beta_1 \mathrm{time}_{n,walk}}_{\text{alternative vars. with generic coefficients}} & 0 & +0 & +\gamma_3 \mathrm{swd}_n
\end{array}
$$

Compare the model matrix and you will see that the values there encode the utility functions for each decision-maker.

Here, we try a different formula, where time has alternative-specific instead of generic coefficients, and call it `f3`:

```
f3 <- mFormula(choice ~ 0 | sidewalk_density | time)
```

Note that, since we do not define other alternative-specific variables with generic coefficients, we have to explicitly state that there are `0` such variables! This formula leads to the following model matrix:

```
# Pipe `f2` to next function
f3 %>%
  # Build the model matrix with data set `example_long`
  model.matrix(example_long)
```

```
  (Intercept):Cycle (Intercept):HSR (Intercept):Walk sidewalk_density:Cycle
1                 0               1               0                0.00000
2                 0               0               1                0.00000
3                 0               0               0                0.00000
4                 0               1               0                0.00000
5                 0               0               1                0.00000
6                 0               0               0                0.00000
7                 1               0               0               26.06793
8                 0               1               0                0.00000
9                 0               0               1                0.00000
  sidewalk_density:HSR sidewalk_density:Walk time:Car time:Cycle time:HSR
1             22.63322               0.00000        0   0.000000        5
2              0.00000              22.63322        0   0.000000        0
3              0.00000               0.00000        2   0.000000        0
4             39.64003               0.00000        0   0.000000       10
5              0.00000              39.64003        0   0.000000        0
6              0.00000               0.00000        5   0.000000        0
7              0.00000               0.00000        0   4.371118        0
8             26.06793               0.00000        0   0.000000       20
9              0.00000              26.06793        0   0.000000        0
  time:Walk
1   0.00000
2  21.31439
3   0.00000
4   0.00000
5  12.78863
6   0.00000
7   0.00000
8   0.00000
9  15.00000
```

The utility functions for this are

$$\begin{array}{llllllllllll}
 & & \multicolumn{3}{c}{\overbrace{\hspace{8em}}^{\text{alternative specific constants}}} & & & & \multicolumn{4}{c}{\overbrace{\hspace{20em}}^{\text{alternative vars. with generic coefficients}}} \\
V_{n,car} & = & 0 & +0 & +0 & +0 & +0 & +0 & +\beta_1\,\text{time}_{n,car} & +0 & +0 & +0 \\
V_{n,cycle} & = & \mu_{\text{cycle}} & +0 & +0 & +\gamma_1\,\text{swd}_n & +0 & +0 & +0 & +\beta_1\,\text{time}_{n,cycle} & +0 & +0 \\
V_{n,HSR} & = & 0 & +\mu_{\text{HSR}} & +0 & +0 & +\gamma_2\,\text{swd}_n & +0 & +0 & +0 & +\beta_1\,\text{time}_{n,HSR} & +0 \\
V_{n,walk} & = & 0 & +0 & +\mu_{\text{walk}} & +0 & +0 & +\gamma_3\,\text{swd}_n & +0 & +0 & +0 & +\beta_1\,\text{time}_{n,walk} \\
 & & & & & \multicolumn{3}{c}{\underbrace{\hspace{10em}}_{\text{individual vars with specific coefficients}}} & & & &
\end{array}$$

Given the utility functions, the multinomial logit probabilities for each alternative are

$$\begin{aligned}
P(car) &= \frac{e^{V_{car}}}{e^{V_{car}} + e^{V_{cycle}} + e^{V_{HSR}} + e^{V_{walk}}} \\
P(cycle) &= \frac{e^{V_{cycle}}}{e^{V_{car}} + e^{V_{cycle}} + e^{V_{HSR}} + e^{V_{walk}}} \\
P(HSR) &= \frac{e^{V_{HSR}}}{e^{V_{car}} + e^{V_{cycle}} + e^{V_{HSR}} + e^{V_{walk}}} \\
P(walk) &= 1 - P(car) - P(HSR) - P(cycle)
\end{aligned}$$

The values of the utility functions depend on the data but also on the coefficients, which we do not know *a priori*. Rather, these must be retrieved from the sample, as discussed next.

5.8 Estimation

Before we can calculate the choice probabilities, we need to somehow obtain coefficients for the utility functions. The process to do so is called *estimation*, and it involves the use of a statistical sample (subset of choice situations).

To estimate the coefficients of a model we need to define a criterion that we wish to satisfy with our choice of coefficients. Estimates can take an infinite number of values, after all, so our criterion must be optimal in some sense—in this way, once we estimate the coefficients, we can be satisfied that they are the best that could obtain for the model under consideration, given then inputs.

A common criterion used to estimate discrete choice models is the *likelihood*. So what is this likelihood? Previously we encountered probability distribution functions. These functions were defined by parameters (such as the location parameter and the dispersion parameter). Given the parameters, it is possible to calculate the probability of values for a variable x. A likelihood function is a similar concept, except that whereas in the probability functions the parameters were given, in a likelihood function the data are given and the parameters need to be obtained from the function.

The relevant likelihood function for the multinomial logit model is as follows:

$$L = \prod_{n=1}^{N} \prod_{j=1}^{J} P_{nj}^{y_{nj}}$$

Table 5.1 Toy data set to illustrate the likelihood function

Individual	Choice	yA	yB	xA	xB
1	A	1	0	5	4
2	A	1	0	2	5
3	B	0	1	5	2
4	A	1	0	1	6
5	B	0	1	4	1
6	B	0	1	3	4

where P_{nj} is the probability of decision-maker n selecting alternative j and y_{nj} is an indicator variable that takes the value of 1 if individual n chose alternative j and 0 otherwise. The effect of the indicator variable is to turn the probabilities on and off, since $P^0 = 1$ and $P^1 = P$. Notice that the likelihood function is bounded between 0 and 1, but in the case of the logit model it is never exactly zero nor one, since the logit probabilities never take any of those exact values.

We can explore the behavior of the likelihood function by means of a simple example. Consider a binomial logit model, that is, a model with only two alternatives in the choice set, say A and B. The likelihood function of this model is as follows:

$$L = \prod_{n=1}^{N} P_{nA}^{y_{nA}} P_{nB}^{y_{nB}} = \prod_{n=1}^{N} \left(\frac{e^{V_{nA}}}{e^{V_{nA}} + e^{V_{nB}}}\right)^{y_{nA}} \left(\frac{e^{V_{nB}}}{e^{V_{nA}} + e^{V_{nB}}}\right)^{y_{nB}}$$

The utility functions V_{nA} and V_{nB} depend on two things: (1) the data, which we know since we have a statistical sample; and (2) the coefficients, which we do not know.

To illustrate the likelihood function we will use a toy sample with six individuals, as shown in Table 5.1.

Based on this toy sample, we can specify the utility functions in this fashion:

$$\begin{aligned} V_{nA} &= 0 \ + \beta x_{nA} \\ V_{nB} &= \mu \ + \beta x_{nB} \end{aligned}$$

These utility functions are very similar to the first set of utility functions that we defined in the preceding section for the table with mode choices.

Next, the likelihood function for this toy sample can be written as a function of μ and β. In this way, it is possible to calculate an initial value of the likelihood function by setting μ and β to zero. We will call this “Experiment 1”:

```r
# Set the parameters:
mu <- 0
beta <- 0

# Calculate probabilities. Notice that these are the logit probabilities
# Individual 1
P1A <- (exp(beta * ts$xA[1])/
          (exp(beta * ts$xA[1]) + exp(mu + beta * ts$xB[1])))
P1B <- 1 - P1A
# Individual 2
P2A <- (exp(beta * ts$xA[2])/
          (exp(beta * ts$xA[2]) + exp(mu + beta * ts$xB[2])))
P2B <- 1 - P2A
# Individual 3
P3A <- (exp(beta * ts$xA[3])/
          (exp(beta * ts$xA[3]) + exp(mu + beta * ts$xB[3])))
P3B <- 1 - P3A
# Individual 4
P4A <- (exp(beta * ts$xA[4])/
          (exp(beta * ts$xA[4]) + exp(mu + beta * ts$xB[4])))
P4B <- 1 - P4A
# Individual 5
P5A <- (exp(beta * ts$xA[5])/
          (exp(beta * ts$xA[5]) + exp(mu + beta * ts$xB[5])))
P5B <- 1 - P5A
# Individual 6
P6A <- (exp(beta * ts$xA[6])/
          (exp(beta * ts$xA[6]) + exp(mu + beta * ts$xB[6])))
P6B <- 1 - P6A

# Calculate likelihood function as the product of all the probabilities
# Each probability is raised to ynj
L <-  P1A^ts$yA[1] * P1B^ts$yB[1] *
  P2A^ts$yA[2] * P2B^ts$yB[2] *
  P3A^ts$yA[3] * P3B^ts$yB[3] *
  P4A^ts$yA[4] * P4B^ts$yB[4] *
  P5A^ts$yA[5] * P5B^ts$yB[5] *
  P6A^ts$yA[6] * P6B^ts$yB[6]

# Create data frame to tabulate results:
df_experiment_1 <- data.frame(Individual = c(1, 2, 3, 4, 5, 6),
                              Choice = c("A", "A", "B", "A", "B", "B"),
                              PA = c(P1A, P2A, P3A, P4A, P5A, P6A),
                              PB = c(P1B, P2B, P3B, P4B, P5B, P6B))

# Display table
kable(df_experiment_1,
      "latex",
      digits = 4,
      booktabs = TRUE,
      align = c("l", "c", "c", "c")) %>%
  kable_styling(bootstrap_options = c("striped", "hover")) %>%
  footnote(general = paste("The value of the likelihood function in Example 1 is: ",
                           round(L, digits = 4)))
```

Individual	Choice	PA	PB
1	A	0.5	0.5
2	A	0.5	0.5
3	B	0.5	0.5
4	A	0.5	0.5
5	B	0.5	0.5
6	B	0.5	0.5

Note:
The value of the likelihood function in Example 1 is: 0.0156

As you can see, the logit probabilities when all coefficients are zero is 0.5. By setting the coefficients to zero we have defined what is called a *null model*. Since the variables vanish in this case, this model has no useful information to discriminate between the various alternatives, and concludes that they are all equally likely. The value of the likelihood function is a relatively small (positive) number (remember, the function is bounded between zero and one).

Next let us experiment with the coefficients, by giving them different values as follows (call this "Experiment 2"):

```
# Set the parameters:
mu <- 0.5
beta <- -0.5

# Calculate probabilities. Notice that these are the logit probabilities
# Individual 1
P1A <- (exp(beta * ts$xA[1])/
          (exp(beta * ts$xA[1]) + exp(mu + beta * ts$xB[1])))
P1B <- 1 - P1A
# Individual 2
P2A <- (exp(beta * ts$xA[2])/
          (exp(beta * ts$xA[2]) + exp(mu + beta * ts$xB[2])))
P2B <- 1 - P2A
# Individual 3
P3A <- (exp(beta * ts$xA[3])/
          (exp(beta * ts$xA[3]) + exp(mu + beta * ts$xB[3])))
P3B <- 1 - P3A
# Individual 4
P4A <- (exp(beta * ts$xA[4])/
          (exp(beta * ts$xA[4]) + exp(mu + beta * ts$xB[4])))
P4B <- 1 - P4A
# Individual 5
P5A <- (exp(beta * ts$xA[5])/
          (exp(beta * ts$xA[5]) + exp(mu + beta * ts$xB[5])))
P5B <- 1 - P5A
# Individual 6
P6A <- (exp(beta * ts$xA[6])/
          (exp(beta * ts$xA[6]) + exp(mu + beta * ts$xB[6])))
P6B <- 1 - P6A

# Calculate likelihood function as the product of all the probabilities
# Each probability is raised to ynj
L <-  P1A^ts$yA[1] * P1B^ts$yB[1] *
  P2A^ts$yA[2] * P2B^ts$yB[2] *
  P3A^ts$yA[3] * P3B^ts$yB[3] *
  P4A^ts$yA[4] * P4B^ts$yB[4] *
  P5A^ts$yA[5] * P5B^ts$yB[5] *
  P6A^ts$yA[6] * P6B^ts$yB[6]
```

```
# Create data frame to tabulate results:
df_experiment_2 <- data.frame(Individual = c(1, 2, 3, 4, 5, 6),
                              Choice = c("A", "A", "B", "A", "B", "B"),
                              PA = c(P1A, P2A, P3A, P4A, P5A, P6A),
                              PB = c(P1B, P2B, P3B, P4B, P5B, P6B))

# Display table
kable(df_experiment_2,
      "latex",
      digits = 4,
      booktabs = TRUE,
      align = c("l", "c", "c", "c")) %>%
  kable_styling(bootstrap_options = c("striped", "hover")) %>%
  footnote(general = paste("The value of the likelihood function in Example 2 is: ",
                           round(L, digits = 4)))
```

Individual	Choice	PA	PB
1	A	0.2689	0.7311
2	A	0.7311	0.2689
3	B	0.1192	0.8808
4	A	0.8808	0.1192
5	B	0.1192	0.8808
6	B	0.5000	0.5000

Note:
The value of the likelihood function in Example 2 is: 0.0672

Notice how changing the coefficients has two effects, as you would expect: the probabilities change and the value of the likelihood function changes too. Inspect the probabilities and the value of the likelihood function with the new coefficients. What do you notice? If you are working with the code, at this point you can try changing the coefficients. Can you improve the value of the likelihood function (i.e., increase it), or maybe even make it worse (i.e., decrease it)? (Why is an increase good)?

The likelihood function can be plotted as shown in Fig. 5.1. If you are working with the code, you can hover over the plot and see how the value of the likelihood changes as a function of μ and β.

```
# Create a grid to plot the likelihood function
mu = seq(from = -1, to = 1, by = 0.05)
beta = seq(from = -2, to = 0, by = 0.05)
coeffs <- expand.grid(mu, beta)

# Define the likelihood function
lkh <- function(mu = 0, beta = 0){
  ts <- data.frame(Individual = c(1, 2, 3, 4, 5, 6),
                   Choice = c("A", "A", "B", "A", "B", "B"),
                   yA = c(1, 1, 0, 1, 0, 0),
                   yB = c(0, 0, 1, 0, 1, 1),
                   xA = c(5, 2, 5, 1, 4, 3),
                   xB = c(4, 5, 2, 6, 1, 4))

  P1A <- (exp(beta * ts$xA[1])/
            (exp(beta * ts$xA[1]) + exp(mu + beta * ts$xB[1])))
  P1B <- 1 - P1A
  P2A <- (exp(beta * ts$xA[2])/
            (exp(beta * ts$xA[2]) + exp(mu + beta * ts$xB[2])))
  P2B <- 1 - P2A
  P3A <- (exp(beta * ts$xA[3])/
            (exp(beta * ts$xA[3]) + exp(mu + beta * ts$xB[3])))
  P3B <- 1 - P3A
  P4A <- (exp(beta * ts$xA[4])/
            (exp(beta * ts$xA[4]) + exp(mu + beta * ts$xB[4])))
  P4B <- 1 - P4A
  P5A <- (exp(beta * ts$xA[5])/
            (exp(beta * ts$xA[5]) + exp(mu + beta * ts$xB[5])))
  P5B <- 1 - P5A
  P6A <- (exp(beta * ts$xA[6])/
            (exp(beta * ts$xA[6]) + exp(mu + beta * ts$xB[6])))
  P6B <- 1 - P6A

  P1A^ts$yA[1] * P1B^ts$yB[1] *
    P2A^ts$yA[2] * P2B^ts$yB[2] *
    P3A^ts$yA[3] * P3B^ts$yB[3] *
    P4A^ts$yA[4] * P4B^ts$yB[4] *
    P5A^ts$yA[5] * P5B^ts$yB[5] *
    P6A^ts$yA[6] * P6B^ts$yB[6]
}

# Evaluate the likelihood function on the grid
L <- lkh(mu = coeffs$Var1, beta = coeffs$Var2)

L <- data.frame(mu = coeffs$Var1, beta = coeffs$Var2, L)
L <- xtabs(L ~ beta + mu, L) %>% # Convert to cross-tabulation matrix
  unclass() # Drop the xtabs class (plotly does not like it)

likelihood_plot <- plot_ly(z = ~L, x = ~mu, y = ~beta) %>%
  add_surface() %>%
  layout(scene = list(
    xaxis = list(title = "x-axis (mu)"),
    yaxis = list(title = "y-axis (beta)"),
    zaxis = list(title = "$z$-axis (L)")))
```

```
# This code displays the figure in the Rmd document
# but is not run for knitting to pdf
likelihood_plot
```

From Fig. 5.1, we can see that the approximate values of the coefficients that maximize the likelihood function are $\mu = 0.10$ and $\beta = -0.65$. If we use these coefficients to calculate the logit probabilities, we can compare the probabilities of Experiments 1 and 2:

```
# Approximate values that maximize the likelihood function.
mu <- 0.10
beta <- -0.65

# Calculate probabilities. Notice that these are the logit probabilities
# Individual 1
P1A <- (exp(beta * ts$xA[1])/
          (exp(beta * ts$xA[1]) + exp(mu + beta * ts$xB[1])))
P1B <- 1 - P1A
# Individual 2
P2A <- (exp(beta * ts$xA[2])/
          (exp(beta * ts$xA[2]) + exp(mu + beta * ts$xB[2])))
P2B <- 1 - P2A
```

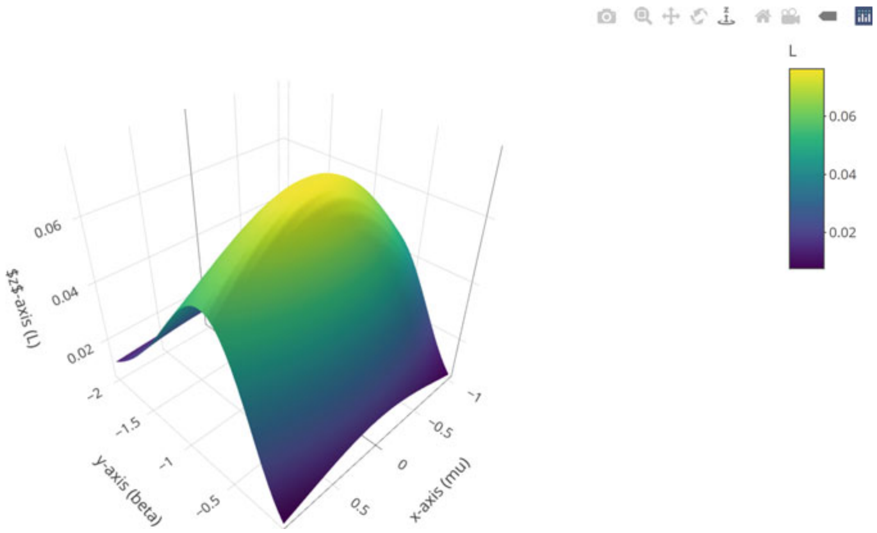


Fig. 5.1 Likelihood function for toy data set

```
# Individual 3
P3A <- (exp(beta * ts$xA[3])/
          (exp(beta * ts$xA[3]) + exp(mu + beta * ts$xB[3])))
P3B <- 1 - P3A
# Individual 4
P4A <- (exp(beta * ts$xA[4])/
          (exp(beta * ts$xA[4]) + exp(mu + beta * ts$xB[4])))
P4B <- 1 - P4A
# Individual 5
P5A <- (exp(beta * ts$xA[5])/
          (exp(beta * ts$xA[5]) + exp(mu + beta * ts$xB[5])))
P5B <- 1 - P5A
# Individual 6
P6A <- (exp(beta * ts$xA[6])/
          (exp(beta * ts$xA[6]) + exp(mu + beta * ts$xB[6])))
P6B <- 1 - P6A

# Calculate likelihood function as the product of all the probabilities
# Each probability is raised to ynj
L <-  P1A^ts$yA[1] * P1B^ts$yB[1] *
  P2A^ts$yA[2] * P2B^ts$yB[2] *
  P3A^ts$yA[3] * P3B^ts$yB[3] *
  P4A^ts$yA[4] * P4B^ts$yB[4] *
  P5A^ts$yA[5] * P5B^ts$yB[5] *
  P6A^ts$yA[6] * P6B^ts$yB[6]

# Create data frame to tabulate results:
df_approx_solution <- data.frame(Individual = c(1, 2, 3, 4, 5, 6),
                                 Choice = c("A", "A", "B", "A", "B", "B"),
                                 PA = c(P1A, P2A, P3A, P4A, P5A, P6A),
                                 PB = c(P1B, P2B, P3B, P4B, P5B, P6B))

# Join tables for displaying results
df <- df_experiment_1 %>%
  left_join(df_experiment_2,
            by = c("Individual", "Choice")) %>%
  left_join(df_approx_solution,
            by = c("Individual", "Choice"))

# Display table
kable(df,
      "latex",
      digits = 4,
      booktabs = TRUE,
      col.names = c("Individual", "Choice", "PA", "PB", "PA", "PB", "PA", "PB"),
      align = c("l", "c", "c", "c", "c", "c", "c", "c")) %>%
  kable_styling(latex_options = c("striped")) %>%
  add_header_above(c(" " = 1, " " = 1,
                     "Experiment 1" = 2,
                     "Experiment 2" = 2,
                     "Approx Max Likelihood" = 2))
```

		Experiment 1		Experiment 2		Approx Max Likelihood	
Individual	Choice	PA	PB	PA	PB	PA	PB
1	A	0.5	0.5	0.2689	0.7311	0.3208	0.6792
2	A	0.5	0.5	0.7311	0.2689	0.8641	0.1359
3	B	0.5	0.5	0.1192	0.8808	0.1141	0.8859
4	A	0.5	0.5	0.8808	0.1192	0.9589	0.0411
5	B	0.5	0.5	0.1192	0.8808	0.1141	0.8859
6	B	0.5	0.5	0.5000	0.5000	0.6341	0.3659

Maximizing the likelihood is a useful criterion to estimate the coefficients of the models, since this criterion provides the optimal probabilities of the right alternative being chosen. Mind you, this does not necessarily mean that those probabilities will be high—however, we can be certain that they will be the best for the model under consideration for the sample given.

In this toy example we “solved” the problem of maximizing the likelihood by hand. This is rather difficult, unfeasible even, in most applied situations with large samples and/or more than one variable. Fortunately, there are a number of numerical algorithms that can be used to maximize the likelihood. We will not discuss this issue in detail, but interested readers can consult [Train (2009); Sect. 3.7] for details. The {mlogit} package imports the package `maxLik` (Henningsen and Toomet 2011), which implements canonical algorithms including Newton-Raphson, the Berndt–Hall–Hall–Hausman (or BHHH), and the Broyden–Fletcher–Goldfarb–Shanno (or BFGS) algorithm.

In practice, the algorithms above do not maximize the likelihood function, but a transformation thereof, called the *log-likelihood*, which is obtained by taking the natural logarithm of the function, to give:

$$l = \sum_{n=1}^{N} \sum_{j=1}^{J} y_{nj} log(P_{nj})$$

Maximizing the log-likelihood function, instead of the likelihood, is easier from a computational standpoint and reduces numerical precision errors.

Since the likelihood function is bound between zero and one, the log-likelihood is bound between minus infinity and zero. The value of the maximized log-likelihood function provides a useful diagnostic to compare models, since higher values are indicative of a better model. Several statistical tests (such as the likelihood ratio) can be used to test the hypothesis that a model is a significant improvement over another, and are thus useful for model selection purposes. However, before discussing model diagnostics, we will see how multinomial logit models are estimated using {mlogit}.

5.9 Example: A Logit Model of Mode Choice

Coming back to the transportation mode choice data set, we had already defined some formulas (i.e., utility functions) that we can use to estimate a model. We need to reshape the full table from *wide* to *long*:

```
mc_commute_long <- mc_commute_wide %>%
  mlogit.data(shape="wide",
              # Name of column with the choices
              choice = "choice",
              # Numbers of columns with attributes that vary by alternative
              varying = 3:22)
```

The function to estimate a model is `mlogit()`. This function requires at least two arguments: an `mFormula` object and a data set. We can verify that the formulas we created above are of this class:

```
class(f1)
```

```
## [1] "mFormula" "Formula"  "formula"
```

```
class(f2)
```

```
## [1] "mFormula" "Formula"  "formula"
```

```
class(f3)
```

```
## [1] "mFormula" "Formula"  "formula"
```

The value (output) of the function can be named and saved to an object for further analysis or for further processing, post-estimation. We begin by estimating a model using the simplest of our formulas and displaying the output using the function `summary()`:

```
# Function `mlogit()` is used to estimate logit models
# It needs a multi-part formula and a data set in long form
model1 <- mlogit(f1,
                 mc_commute_long)

# Function `summary()` give the summary of data objects,
# including the output of model estimation algorithms
summary(model1)
```

```
Call:
mlogit(formula = choice ~ time, data = mc_commute_long, method = "nr")

Frequencies of alternatives:choice
     Car    Cycle      HSR     Walk 
0.204364 0.034182 0.244364 0.517091 

nr method
5 iterations, 0h:0m:0s 
g’(-H)^-1g = 0.000102 
successive function values within tolerance limits 

Coefficients :
                   Estimate Std. Error z-value  Pr(>|z|)    
(Intercept):Cycle  0.366898   0.201434  1.8214   0.06854 .  
(Intercept):HSR    0.707267   0.118104  5.9885 2.118e-09 ***
(Intercept):Walk   3.212834   0.184169 17.4451 < 2.2e-16 ***
time              -0.056494   0.005685 -9.9374 < 2.2e-16 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Log-Likelihood: -768.63
McFadden R^2:  0.50323 
Likelihood ratio test : chisq = 1557.2 (p.value = < 2.22e-16)
```

The output of the function includes the observed frequencies of alternatives in addition to information about the optimization procedure. For instance, the message “successive function values within tolerance limits” indicates that the algorithm converged normally.

The output also reports the estimated values of the coefficients, along with standard errors, z-values, and p-values. The null hypothesis associated with the coefficients is that they are zero. An analyst can reject the null hypothesis in any case, but small p-values indicate a low probability that the coefficient is zero—and therefore increase the confidence that by rejecting the null hypothesis the analyst is not mistakenly rejecting a true zero. In the present case, with p-values smaller than 0.0001, the null hypothesis can be comfortably rejected for every coefficient. The small p-values mean that it is highly unlikely that the coefficients are zero.

This simple model includes three alternative-specific constants and one alternative-specific variable with a generic coefficient. The signs of the coefficients are informative. Since the reference mode is “Car”, the positive values of the constants indicate that, other things being equal (that is, the variables considered in the model), car is

Table 5.2 Estimation results: Model 1

	Dependent variable:
	Choice
(Intercept):Cycle	0.367* (0.201)
(Intercept):HSR	0.707*** (0.118)
(Intercept):Walk	3.213*** (0.184)
time	−0.056*** (0.006)
Observations	1,375
R^2	0.503
Log likelihood	−768.633
LR test	1,557.233*** (df = 4)
Note :	*p<0.1; **p<0.05; ***p<0.01

the least preferred mode, followed by cycling, HSR, and then walk (which gives the highest utility at a constant value of time).

The negative coefficient for time indicates that time is a “cost” or “disutility”, in other words, the utility of traveling tends to decline with increasing travel times. This indicates that slower modes tend to have lower utilities, when all other variables are kept the same.

Finally, the maximized value of the log-likelihood function is reported, along with two diagnostics, McFadden Rˆ2 (in reality ρ^2) and a likelihood ratio test. We will come back to these diagnostics later.

The `summary()` function is useful in providing the relevant information, but the format is not necessarily appealing. Different packages are available in `R` to format output. Here, we use the {stargazer} package (which we first encountered in Chap. 2) to present the results of the model we estimated (see Table 5.2):

```
# Note: use chunk option results="asis" to display latex output in pdf
stargazer::stargazer(model1,
                     # Use type = "text", "latex". or "html" depending
                     # on the desired output
                     type ="latex",
                     header = FALSE,
                     single.row = TRUE,
                     title = "Estimation results: Model 1")
```

We will now estimate a new model using the second formula (Table 5.3):

```
# Note: use chunk option results="asis" to display latex output in pdf
model2 <- mlogit(f2,
                mc_commute_long)

stargazer::stargazer(model2,
                     # Use type = "text", "latex". or "html" depending
                     # on the desired output
                     type = "latex",
                     header = FALSE,
                     single.row = TRUE,
                     title = "Estimation results: Model 2")
```

Now there is an individual-specific variable in the model (i.e., sidewalk density). Only one of the three coefficients is significant at a conventional level of significance (i.e., $p < 0.05$), and the value is positive. Since the reference is "Car", a positive value indicates that higher sidewalk density is associated with an increase in the utility of walking with respect to the utility of using a car. The same is not true for "HSR" and "Cycle", whose coefficients for this attribute are not significantly different from "Car" (Table 5.3).

Note that it is possible to select the reference level for the utilities when estimating the model. For example, below we re-estimate the preceding model, but now using the utility of Walk as the reference (see Table 5.4):

```
# Note: use chunk option results="asis" to display latex output in pdf
model2 <- mlogit(f2,
                 mc_commute_long,
                 # Specify the alternative that acts as reference
                 reflevel = "Walk")

stargazer::stargazer(model2,
                     # Use type = "text", "latex". or "html" depending
                     # on the desired output
                     type = "latex",
                     header = FALSE,
                     single.row = TRUE,
                     title = "Estimation results: Model 2 (alternative: Walk)")
```

Notice that now two sidewalk coefficients are significant! While sidewalk density does not significantly change the utility of cycling with respect to walking, living in a place with high sidewalk density reduces the utility of "Car" and "HSR" *with*

Table 5.3 Estimation results: Model 2

	Dependent variable:
	Choice
(Intercept):Cycle	−0.306 (0.454)
(Intercept):HSR	0.676*** (0.215)
(Intercept):Walk	2.345*** (0.312)
time	−0.056*** (0.006)
sidewalk_density:Cycle	0.027 (0.017)
sidewalk_density:HSR	0.002 (0.009)
sidewalk_density:Walk	0.035*** (0.011)
Observations	1,375
R^2	0.509
Log likelihood	−760.418
LR test	1,573.662*** (df = 7)
Note :	*p<0.1; **p<0.05; ***p<0.01

Table 5.4 Estimation results: Model 2 (alternative: Walk)

	Dependent variable:
	Choice
(Intercept):Car	−2.345*** (0.312)
(Intercept):Cycle	−2.651*** (0.419)
(Intercept):HSR	−1.669*** (0.230)
time	−0.056*** (0.006)
sidewalk_density:Car	−0.035*** (0.011)
sidewalk_density:Cycle	−0.008 (0.015)
sidewalk_density:HSR	−0.033*** (0.008)
Observations	1,375
R^2	0.509
Log likelihood	−760.418
LR test	1,573.662*** (df = 7)
Note:	*p<0.1; **p<0.05; ***p<0.01

respect to walking. Note that when using car as the reference, it is not possible to conclude whether the coefficients for walking and HSR are significantly different (Table 5.4).

The value of the maximized log-likelihood and other model diagnostics are identical irrespective of which mode is selected as reference. In essence, the models are the same, but they provide a different perspective on how some coefficients relate to each other across alternatives.

We can visually explore how the probability of choosing different modes varies with sidewalk density. To do this we will first summarize the sidewalk density variable:

```
summary(mc_commute_long$sidewalk_density)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   18.19   22.63   24.18   35.70   59.41
```

We then copy the data frame used to estimate the model, but only enough rows to explore sidewalk densities in the range between 0 and 60, in intervals of 5. Therefore we need to copy 52 rows from the long table (thirteen levels of sidewalk density times four alternatives):

```
mc_commute_predict <- mc_commute_long[1:52,]
```

Finally, we replace the sidewalk density variable using values from 0 to 60, with an interval of 5. Since each alternative is a row, we need to create a sequence of replicated values as follows:

```
# Function `rep()` repeats the values in the argument a designated
# number of times; here, the values in the sequence 0 to 60 in intervals
# of 5 are repeated four times each (once for each alternative)
mc_commute_predict$sidewalk_density <- rep(seq(from = 0,
                                               to = 60,
                                               by = 5),
                                           each = 4)
```

We can examine the results of simulating the sidewalk density:

```
mc_commute_predict %>%
  data.frame() %>%
  select(sidewalk_density) %>%
  slice_head(n = 8)
```

```
  sidewalk_density
1                0
2                0
3                0
4                0
5                5
6                5
7                5
8                5
```

The prediction data frame now includes the range of sidewalk densities that we are interested in.

The setup for the simulation essentially amounts to: what is the probability of choosing each mode for a trip that originates in an area with a sidewalk density of 5? Of 10? For the simulation, we also need to set the values of other variables of interest. Since the model has the variable `time` we also need to set it to a desired value, for instance, the median trip duration (which we calculate after removing missing values):

```
median(mc_commute_predict$time,
       na.rm = TRUE)
```

```
[1] 10
```

The median trip length is 10 min, so we replace the current values in the prediction data frame:

```
mc_commute_predict$time <- 10
```

Next, we examine the relevant variables of the prediction data frame:

```
mc_commute_predict %>%
  data.frame() %>%
  select(time, sidewalk_density) %>%
  summary()
```

```
      time     sidewalk_density
 Min.   :10    Min.   : 0
 1st Qu.:10    1st Qu.:15
 Median :10    Median :30
 Mean   :10    Mean   :30
 3rd Qu.:10    3rd Qu.:45
 Max.   :10    Max.   :60
```

```
mc_commute_predict %>%
  data.frame() %>%
  select(time, sidewalk_density) %>%
  slice_head(n = 8)
```

```
  time sidewalk_density
1   10                0
2   10                0
3   10                0
4   10                0
5   10                5
6   10                5
7   10                5
8   10                5
```

Once we have a prediction matrix, we can forecast the probabilities for the given inputs using function `predict()` function and model object `model2`:

```
probs <- predict(model2,
                 newdata = mc_commute_predict)
```

The value (output) of `predict` is a matrix that contains the probability for thirteen levels of sidewalk density (i.e., 0, 5, 10, $\cdots$, 60), and four modes ("Walk", "Cycle", "HSR", "Car").

```
print(probs)
```

```
         Walk        Car      Cycle        HSR
1  0.7381288 0.07072471 0.05208792 0.13905856
2  0.7646590 0.06149222 0.05191414 0.12193469
3  0.7887761 0.05323772 0.05152105 0.10646509
4  0.8105461 0.04591523 0.05093563 0.09260301
5  0.8300785 0.03946495 0.05018525 0.08027133
6  0.8475135 0.03381834 0.04929656 0.06937162
7  0.8630106 0.02890248 0.04829466 0.05979230
8  0.8767385 0.02464350 0.04720262 0.05141540
9  0.8888677 0.02096925 0.04604115 0.04412190
10 0.8995646 0.01781113 0.04482852 0.03779578
11 0.9089873 0.01510533 0.04358057 0.03232680
12 0.9172834 0.01279350 0.04231084 0.02761229
13 0.9245881 0.01082299 0.04103075 0.02355815
```

As discussed in Chap. 2, it is much easier to identify patterns visually. Notice that the table with probabilities is wide: there is a column for each mode, but the contents of the table are all one single type of measurement, the probability. To facilitate plotting, we add the sidewalk density values and then reshape the table to long form, so that each row is a single probability for a mode-sidewalk density combination (which should be a 52 × 3 matrix) as follows:

```
probs <- data.frame(sidewalk_density = seq(from = 0,
                                           to = 60,
                                           by = 5),
                    probs) %>%
  # Pivot longer all columns _except_ `sidewalk_density`
  pivot_longer(cols = -sidewalk_density,
               # The column names become a new column called "Mode"
               names_to = "Mode",
               # The values are gathered into a single column called
               # "Probability"
               values_to = "Probability")
```

By pivoting the table with the probabilities we reshape it (make it longer), so that now the data frame has one column with the sidewalk density, one with the mode and one column with the probability.

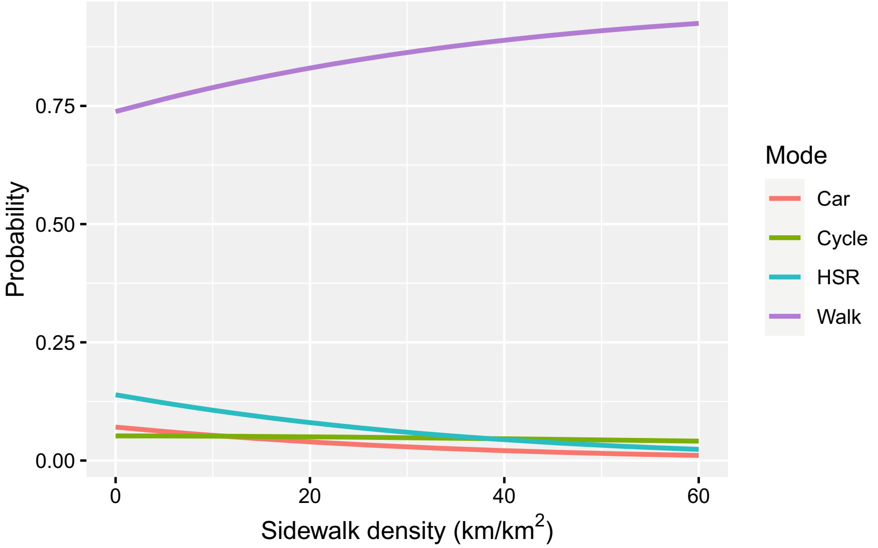
```
probs %>%
  slice_head(n = 8)
```

```
# A tibble: 8 x 3
  sidewalk_density Mode   Probability
             <dbl> <chr>        <dbl>
1                0 Walk        0.738
2                0 Car         0.0707
3                0 Cycle       0.0521
```

```
4                0 HSR       0.139
5                5 Walk      0.765
6                5 Car       0.0615
7                5 Cycle     0.0519
8                5 HSR       0.122
```

We can then plot the probabilities, using different colors for each mode. This is an example of *post-estimation* exploratory data analysis, and our objective is to use visualization tools to help us understand the behavior of the model:

```
probs %>%
  # Create ggplot object; map `sidewalk_density` to the y-axis
  # `Probability` to the x-axis, and the color of geometric
  # objects to `Mode`
  ggplot(aes(x = sidewalk_density,
             y = Probability,
             color = Mode)) +
  # Add geometric object of type line with size = 1
  geom_line(size = 1) +
  labs(y="Probability",
       x = expression("Sidewalk density (km/km"^2*")"))
```



We can see that the probability of walking (for a trip that takes the median duration of the observations in the sample, i.e., 10 min) tends to increase as the density of sidewalks increases. The probability of using the three other modes tends to decrease with sidewalk density, but more rapidly for HSR than for car or cycling. Keep in mind that these probabilities were simulated after setting travel time to the median travel time in the sample, i.e., 10 min. The probability of walking is very high when trips are relatively short; the effect of higher sidewalk density is to make it easier to choose walking.

5.10 Comparing Models: McFadden's ρ^2

The log-likelihood reported in the summary of the model is useful as a measure of goodness of fit. Recall that the likelihood is bounded between 0 and 1, and therefore the log-likelihood is bounded at the upper end by 0 (it is minus infinity at the lower end). We also know that higher values of the likelihood represent better fits.

One simple diagnostic to compare the fit of models is McFadden's ρ^2. This summary diagnostic is defined as follows:

$$\rho^2 = 1 - \frac{l^*}{l_0}$$

where l^* is the value of the maximized log-likelihood and l_0 is the value of the log-likelihood of a null model. A null model could be as follows, for a decision-maker n with J_n alternatives in their choice set:

$$\begin{aligned} V_{n1} &= 0 \\ \cdots & \\ V_{nj} &= 0 \\ \cdots & \\ V_{nJ_n} &= 0 \end{aligned}$$

This set of systematic utility functions is an admission of complete ignorance about the factors that influence choices; with nothing but the random utilities to go, the logit probabilities become

$$\begin{aligned} P_{n1} &= \tfrac{1}{J_n} \\ \cdots & \\ P_{nj} &= \tfrac{1}{J_n} \\ \cdots & \\ P_{nJ_n} &= \tfrac{1}{J_n} \end{aligned}$$

This model is sometimes called "Equally Likely" (Ortúzar and Willumsen 2011, Fourth Edition:281). Since the model lacks information about the alternatives, it concludes that the chances of choosing any of the alternatives are equiprobable. The likelihood of this model can be used as the benchmark that gives no information at all about the choice process.

An alternative null model includes only the constants (Table 5.5):

Table 5.5 Estimation results: Market shares model (Null Model)

	Dependent variable:
	Choice
(Intercept):Cycle	-1.788^{***} (0.158)
(Intercept):HSR	0.179^{**} (0.081)
(Intercept):Walk	0.928^{***} (0.070)
Observations	1,375
R^2	0.000
Log likelihood	$-1{,}547.249$
LR test	0.000 (df = 3)
Note:	*p<0.1; **p<0.05; ***p<0.01

$$
\begin{array}{l}
V_{n1} = 0\\
\cdots\\
V_{nj} = \mu_j\\
\cdots\\
V_{nJ} = \mu_J
\end{array}
$$

This model is called “Market Shares” (Ortúzar and Willumsen 2011, Fourth Edition:281) and introducing at least a systematic component has the property that the model replicates the observed shares in the sample. The next chunk of code specifies and estimates a Market Shares model

```
f0 <- mFormula(choice ~ 1)

model0 <- mlogit(f0,
                 mc_commute_long)

stargazer::stargazer(model0,
                     # Use type = "text", "latex". or "html" depending
                     # on the desired output
                     type = "latex",
                     header = FALSE,
                     single.row = TRUE,
                     title = "Estimation results: Market Shares Model (Null Model)")
```

The log-likelihood of the Market Shares model is –1547.249. The log-likelihood of `model2` is –760.418. Accordingly, McFadden’s ρ^2 for `model2` is (compare the R^2 reported in the table for this model) (Table 5.5):

```
1 - as.numeric(model2$logLik)/as.numeric(model0$logLik)
```

```
[1] 0.5085353
```

When a fully specified model is not very informative, its log-likelihood will tend to the log-likelihood of the null model, because in this case l^*/l_0 tends to one and therefore ρ^2 tends to zero. If the maximized log-likelihood of the model tends to 0 (the upper limit for the log-likelihood function), ρ^2 tends to one.

Although ρ^2 is bounded between zero and one, just like the coefficient of determination R^2 in regression analysis, its interpretation is *not* the same as for R^2. Whereas R^2 is interpreted as the proportion of variance explained by the model, ρ^2 lacks such an interpretation. Also, the values of ρ^2 tend to be lower, and values of 0.4 are conventionally considered very good fits (Ortúzar and Willumsen 2011, Fourth Edition:282). The main utility of McFadden's ρ^2 is as a quick way of comparing the fit of different models relative to an uninformative benchmark, rather than assessing the fit against an absolute *best* value of goodness of fit.

5.11 Comparing Models: The Likelihood Ratio Test

Another way to compare models is by means of the likelihood ratio test. This test compares the log-likelihood of two models to assess whether they are significantly different. The test follows the χ^2 distribution with degrees of freedom equal to the difference in the number of coefficients between the two models. The test requires a base model and a full model, and the base model must *nest* within the full model. Nesting in this sense means that the full model must be reducible to the base model. This can be done by restricting some coefficients so that the two models become identical.

For example, consider the utility functions of `model2`:

$$\begin{array}{llll} V_{n,car} & = 0 & +\beta_1 \text{time}_{i\text{Car}} & +0 \\ V_{n,cycle} & = \mu_{cycle} & +\beta_1 \text{time}_{n,cycle} & +\gamma_1 \text{swd}_n \\ V_{n,HSR} & = \mu_{HSR} & +\beta_1 \text{time}_{n,HSR} & +\gamma_2 \text{swd}_n \\ V_{n,walk} & = \mu_{walk} & +\beta_1 \text{time}_{n,walk} & +\gamma_3 \text{swd}_n \end{array}$$

We can convert this model into `model1` by setting $\gamma_1 = \gamma_2 = \gamma_3 = 0$:

$$\begin{array}{lll} V_{n,car} & = 0 & +\beta_1 \text{time}_{i\text{Car}} \\ V_{n,cycle} & = \mu_{cycle} & +\beta_1 \text{time}_{n,cycle} \\ V_{n,HSR} & = \mu_{HSR} & +\beta_1 \text{time}_{n,HSR} \\ V_{n,walk} & = \mu_{walk} & +\beta_1 \text{time}_{n,walk} \end{array}$$

In this way, `model1` "nests" in `model2`.

In the summary of the models, the likelihood ratio test is reported. The test reported in the output of the model is against the null model, that is, a model with no variables at all. This is the least informative of all models. When two non-null models need to be compared, the `lrtest` function implements the likelihood ratio test for two inputs, which are two {mlogit} models, as follows:

```
lrtest(model1,
       model2)
```

```
  Likelihood ratio test

  Model 1: choice ~ time
  Model 2: choice ~ time | sidewalk_density
    #Df  LogLik Df  Chisq Pr(>Chisq)
  1   4 -768.63
  2   7 -760.42  3 16.429  0.0009258 ***
  ---
  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notice that the number of degrees of freedom (Df) is 3: this is because there are three parameters (in this case individual-specific) in `model2` that are not present in `model1`. The null hypothesis of the test is that the log-likelihood of the two models is not different, in other words, that the alternate model is not an improvement over the base model.

In the present case, the very small *p*-value leads us to reject the null hypothesis, and the conclusion is that `model2`, which includes sidewalk density, is a significant improvement over `model1`. In other words, we gain information over `model1` with `model2`.

5.12 Exercises

1. In the example in this chapter we estimated the probabilities of choosing different modes by sidewalk density setting travel time to the in-sample median. Use `model2` to calculate the probability of choosing different modes by in-sample median sidewalk density but now for travel times of 20, 30, and 40 min. Discuss the results.
2. Estimate a model using formula `f3` (call it `model3`). Discuss the output of this model.
3. Use `model3` to calculate the probability of choosing different modes by in-sample median sidewalk density but now for travel times of 20, 30, and 40 min. Discuss the results.
4. In the general case, what is the value of the log-likelihood of the null (Equally Likely) model?
5. Use the likelihood ratio test to compare `model3` to `model2`? Discuss the results. What restrictions would you need to impose in `model3` to obtain `model2`?

References

Akerlof, G. A. (1997). Social distance and social decisions. *Econometrica, 65*(5), 1005–1027. https://ISI:A1997XT90600001.

Axhausen, K. W. (2005). Social networks and travel: Some hypotheses. In K. Donaghy, S. Poppelreuter, & G. Rudinger (Eds.), *Social aspects of sustainable transport: Transatlantic perspectives*, pp. 90–108. Aldershot: Ashgate Publishing.

Axhausen, K. W. (2008). Social networks, mobility biographies, and travel: Survey challenges. *Environment and Planning B-Planning and Design, 35*(6), 981–96.

Ben-Akiva, M., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and applications to travel demand*. Cambridge: The MIT Press.

Dugundji, E. R. & Gulyas, L. (2013). Structure and emergence in a nested logit model with social and spatial interactions. *Computational and Mathematical Organization Theory, 19*(2), 151–203. https://doi.org/10.1007/s10588-013-9157-y.

Dugundji, E. R., & Walker, J. L. (2005). Discrete choice with social and spatial network interdependencies—An empirical example using mixed generalized extreme value models with field and panel effects. *Transportation Research Record, 1921,* 70–78.

Goetzke, F., & Rave, T. (2011). Bicycle use in Germany: Explaining differences between municipalities with social network effects. *Urban Studies, 48*(2), 427–37. https://doi.org/10.1177/0042098009360681.

Henningsen, A., & Toomet, O. (2011). maxLik: A package for maximum likelihood estimation in R. *Computational Statistics, 26*(3), 443–58. https://doi.org/10.1007/s00180-010-0217-1.

Hensher, D. A., Rose, J. M., Rose, J. M., & Greene, W. H. (2005). *Applied choice analysis: A primer*. Cambridge University Press.

Kamargianni, M., Ben-Akiva, M., & Polydoropoulou, A. (2014). Incorporating social interaction into hybrid choice models. *Transportation, 41*(6), 1263–1285. https://doi.org/10.1007/s11116-014-9550-5.

Matous, P. (2017). Complementarity and substitution between physical and virtual travel for instrumental information sharing in remote rural regions: A social network approach. *Transportation Research Part a-Policy and Practice, 99*, 61–79. https://doi.org/10.1016/j.tra.2017.02.010.

Ortúzar, J. D., & Willumsen, L. G. (2011). *Modelling transport* (4th ed.). New York: Wiley.

Páez, A., & Scott, D. M. (2007). Social influence on travel behavior: A simulation example of the decision to telecommute. *Environment and Planning A, 39*(3), 647–65.

Train, K. (2009). *Discrete choice methods with simulation* (2nd ed.). Cambridge: Cambridge University Press.

Van den Berg, P., Arentze, T. A., & Timmermans, H. J. (2009). Size and composition of ego-centered social networks and their effect on geographic distance and contact frequency. *Transportation Research Record, 2135*, 1–9. ISI:000274591300002.

Whalen, K. E., Páez, A., & Carrasco, J. A. (2013). Mode choice of university students commuting to school and the role of active travel. *Journal of Transport Geography, 31*, 132–42. https://doi.org/10.1016/j.jtrangeo.2013.06.008.

Chapter 6
Behavioral Insights from Choice Models

Men's actions are the best guides to their thoughts.
— John Locke, An Essay Concerning Human Understanding—Volume I

Prediction is not proof.
— King and Kraemer, Models, facts, and the policy process: the political ecology of estimated truth

Human behavior is incredibly pliable, plastic.

— Philip Zimbardo

6.1 Inferring and Forecasting Behavior

In Chap. 5 we covered some important practical aspects around the estimation of the multinomial logit model. Many of them transfer to other kinds of discrete choice models as well. Before exploring other models we will take the opportunity, armed as we are with the practical skills to estimate the multinomial logit model, to see how discrete choice models can be used to understand preferences and to infer behavior.

In the preceding chapters we made the assumption that an observer/analyst cannot possibly know the state of mind of the decision-maker, and argued that a model is a way to infer preferences and trade-offs based on what people do (or say they *would* do, in the case of *stated preferences*[1]). Once this has been achieved, what is the model good for?

In addition to providing a plausible description of the decision-making process of interest, a model can be used to examine how behavior *might* change if the conditions

[1] Stated preferences elicit information about the preferences of decision-makers in carefully controlled experimental settings; see Louviere et al. (2000).

A. Páez and G. Boisjoly, *Discrete Choice Analysis with R*, Use R!,
https://doi.org/10.1007/978-3-031-20719-8_6

of the decision-making situation changed. McFadden, in one of the seminal papers on discrete choice analysis (see McFadden 1974), was concerned with travel demand forecasting. The basis for forecasting demand was a model of mode choice that helped to tease out the factors that influenced the choice between car and bus as a mode for commuting to work. Once that this model was estimated, McFadden was interested in the level of demand of a new mode, a rail system called BART (Bay Area Rail Transit). This system had not been built yet, and so obtaining estimates of the level of demand for this mode was an important part of planning and policy analysis. In other words, McFadden was interested in how behavior *might* change with the introduction of a new mode of transportation. What would be the demand for the new mode at a certain fare? If fares changed? If waiting times for BART were longer or shorter? And so on.

With practice you should become familiar with this process of inquiry, which consists of two main steps:

1. Estimate and select a plausible model for the behavior of interest.
2. Analyze scenarios: what would happen if?

This process is valuable in several ways.

First, it helps to clarify issues that are of policy interest (King and Kraemer 1993, 365). Modelers need to document their assumptions, modeling strategies, choices regarding the use of data and model selection, and so on, which results in a systematic approach of simplification (and remember, all models are wrong, but some are useful).

Secondly, the modeling process, and subsequently the use of models, help to enforce discipline in analysis and discourse. The interpretation of models is to some extent limited by their mechanics, and the results should reflect this. For example, incremental changes in the inputs should result in plausible changes in the predictions. Making predictions outside of the calibration range of the model (beyond the range of values of variables used to estimate the model) will likely result in predictions that are implausibly out of bounds. As King and Kraemer note (1993) “the results of radical changes are unlikely to be predicted accurately by models based on the performance under the status quo” (p. 365).

And **thirdly**, when used to analyze scenarios, models provide cautionary advice on *what not to do* by revealing the possible consequences of a bad intervention or policy. In this way, models can help inform planners and policy-makers about the range of possible outcomes and whether these outcomes are in a sense in some “acceptable range” (King and Kraemer 1993, pp. 365–66). In other words, models help to probe the limits of the plasticity of human behavior.

We will explore the preceding issues in this chapter, and we will do this by looking at different ways to derive behavioral insights based on models of choices.

6.2 How to Use This Note

As usual, the source code used in this chapter is available, which means that the chunks of code like the following, can be used to repeat the examples, and to extend the experiments:

```
# Function `writeLines()` is similar to `print()` but allows for line breaks
writeLines("Are these the shadows of the things that Will be,
           or are they shadows of the things that May be only?")
```

```
Are these the shadows of the things that Will be,
           or are they shadows of the things that May be only?
```

If you are working with RStudio you can type the chunks of code to experiment with them. As an alternative, you may copy and paste the source code into your `R` or RStudio console, or create a script/notebook to save the code and any experiments you may conduct.

6.3 Learning Objectives

In this chapter, you will learn about:

1. The meaning of the coefficients of a model.
2. The concept of marginal effects.
3. The concept of elasticity.
4. The concept of willingness to pay.
5. How to simulate outcomes.

6.4 Suggested Readings

- Ben-Akiva, M., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and applications to travel demand* (Chap. 5, pp. 111–113). MIT Press. .
- Hensher, D. A., Rose, J. M., & Greene, W. H. (2005). *Applied choice analysis: A primer*, (Chap. 11). Cambridge University Press.
- King, J. L., & Kraemer, K. L. (1993). *Models, facts, and the policy process: The political ecology of estimated truth*. In Goodchild, M., Parks, B.O., & Stayaert, L.T. (Eds.) *Environmental modelling with GIS*. Oxford University Press.
- Ortuzar, J. D., & Willumsen, L. G. (2011). *Modelling Transport*, 4th edn. (Chap. 2, pp. 43–44). Wiley.
- Train, K. (2009).*Discrete Choice Methods with Simulation*, 2nd edn. (Chap. 2, pp. 29–31). Cambridge University Press.

6.5 Preliminaries

Load the packages used in this section:

```
library(dplyr) # A Grammar of Data Manipulation
library(evd) # Functions for Extreme Value Distributions
library(ggplot2) # Create Elegant Data Visualisations Using the Grammar of Graphics
library(kableExtra) # Construct Complex Table with 'kable' and Pipe Syntax
library(mlogit) # Multinomial Logit Models
library(tidyr) # Tidy Messy Data
```

Load the data set used in this section (from the `mlogit` package):

```
data("Heating")
```

This data set is in wide format, with each row corresponding to an individual decision-maker. It includes choices by a sample of consumers in California with respect to different heating systems. In-depth analysis of this data set was conducted by Train and Croissant (2012) in a document available here.

Five heating systems are considered in this choice problem:

- Gas Central (gc)
- Gas Room (gr)
- Electric Central (ec)
- Electric Room (er)
- Heat Pump (hp)

These heating systems differ in terms of their installation cost (ic) and annual operation cost (oc). See the following table with the median installation cost, annual operation costs, and proportion of choices in the sample:

```
Proportion <- Heating %>%
  # Group the rows by the value of `depvar`
  group_by(depvar) %>%
  # Count the number of cases of each outcome in `depvar`
  summarise(no_rows = n())

df <- data.frame(System = c("Gas Central",
                            "Gas Room",
                            "Electric Central",
                            "Electric Room",
                            "Heat Pump"),
                 Installation = c(median(Heating$ic.gc),
                                  median(Heating$ic.gr),
                                  median(Heating$ic.ec),
                                  median(Heating$ic.er),
                                  median(Heating$ic.hp)),
                 Operation = c(median(Heating$oc.gc),
                               median(Heating$oc.gr),
                               median(Heating$oc.ec),
                               median(Heating$oc.er),
                               median(Heating$oc.hp)),
                 Proportion = Proportion$no_rows/900)
```

```
df %>%
  kable() %>%
  kable_styling()
```

System	Installation	Operation	Proportion
Gas Central	778.505	172.105	0.6366667
Gas Room	924.305	154.110	0.1433333
Electric Central	824.840	480.055	0.0711111
Electric Room	989.700	430.665	0.0933333
Heat Pump	1046.550	220.845	0.0555556

In addition, the data set also includes some information about the consumers, including income, age of household head, number of rooms in the house, and region in California:

```
Heating %>%
  # Select columns 13 to 16
  select(13:16) %>%
  summary()
```

```
     income           agehed          rooms            region
 Min.   :2.000   Min.   :20.00   Min.   :2.000   valley:177
 1st Qu.:3.000   1st Qu.:30.00   1st Qu.:3.000   scostl:361
 Median :5.000   Median :45.00   Median :4.000   mountn:102
 Mean   :4.641   Mean   :42.94   Mean   :4.424   ncostl:260
 3rd Qu.:6.000   3rd Qu.:55.00   3rd Qu.:6.000
 Max.   :7.000   Max.   :65.00   Max.   :7.000
```

The data set is in “wide” format, which means that there is one record per decision-making unit (i.e. per household). Package {mlogit} works with data in “long” format, but it is easier to work with the data frames in their wide format for the purpose of post-estimation analysis. As we saw before in Chap. 5, package {mlogit} includes a function for changing the format. We will call this function when required:

```
H <- Heating %>%
  mlogit.data(shape = "wide",
              # Name of column with the choices
              choice = "depvar",
              # Numbers of columns with attributes that vary by alternative
              varying = c(3:12))
```

In the above, the argument `varying` lets the function know that the variables in the columns 3 through 12 in the data frame are alternative specific, and therefore vary across utility functions. Once that the format of the table has been changed from “wide” to “long”, the number of rows changes from 900 (the number of decision

Table 6.1 Estimation results: model 1 (wide data as input)

	Dependent variable
	Depvar
(Intercept):er	0.537*** (0.192)
(Intercept):gc	2.119*** (0.134)
(Intercept):gr	0.864*** (0.164)
(Intercept):hp	0.123 (0.232)
ic	−0.002*** (0.001)
Observations	900
R^2	0.004
Log Likelihood	−1,018.514
LR Test	7.420 (df = 5)

Note—*p<0.1; **p<0.05; ***p<0.01

makers) to 4500: this is the number of decision-makers (900) times the number of alternatives (5). There is accordingly one row per choice situation.

Before estimating an initial model, we need to define the utility functions that we wish to estimate. Since there are five alternatives, we define the following five functions (with the electric central system as the reference level):

$$\begin{array}{rcll} V_{\text{ec}} &=& 0 & +\beta_1 \text{ic.ec} \\ V_{\text{er}} &=& \beta_{er} & +\beta_1 \text{ic.er} \\ V_{\text{gc}} &=& \beta_{gc} & +\beta_1 \text{ic.gc} \\ V_{\text{gr}} &=& \beta_{gr} & +\beta_1 \text{ic.gr} \\ V_{\text{hp}} &=& \beta_{hp} & +\beta_1 \text{ic.hp} \end{array}$$

These functions include only the installation cost of the systems (ic). The `mlogit` function can be used to estimate this model (call this Model 1), using the electric central system ("ec") as the reference level (Table 6.1):

```r
model1 <- mlogit(depvar ~ ic,
                 Heating,
                 shape = "wide",
                 choice = "depvar",
                 reflevel = "ec",
                 varying = c(3:7))

stargazer::stargazer(model1,
                     type = "latex",
                     header = FALSE,
                     single.row = TRUE,
                     title = "Estimation results: Model 1 (wide data as input)")
```

We can verify that this is identical to estimating the model when the input is the table in long format (Table 6.2):

Table 6.2 Estimation results: model 1 (long data as input)

	Dependent variable
	Depvar
(Intercept):er	0.537*** (0.192)
(Intercept):gc	2.119*** (0.134)
(Intercept):gr	0.864*** (0.164)
(Intercept):hp	0.123 (0.232)
ic	−0.002*** (0.001)
Observations	900
R^2	0.004
Log Likelihood	−1,018.514
LR Test	7.420 (df = 5)

Note—*p<0.1; **p<0.05; ***p<0.01

```
model1 <- mlogit(depvar ~ ic,
                 H,
                 reflevel = "ec")

stargazer::stargazer(model1,
                     type = "latex",
                     header = FALSE,
                     single.row = TRUE,
                     title = "Estimation results: Model 1 (long data as input)")
```

6.6 The Meaning of the Coefficients

The coefficients of a discrete choice model are informative because they relate to the implicit preferences (prices) of the characteristics of an alternative; in this way, they reflect the effect of various variables on the probabilities of selecting alternatives. For this reason, they can be used to understand how decision-makers trade off different attributes.

The first thing to note is the sign of the coefficients. In the case of the binomial and multinomial logit model, the signs of the coefficients are informative because they tell us something about how the utility is affected by an attribute (either of the alternatives or the decision-maker). A positive sign indicates that the utility increases as the attribute increases. For instance, when considering mobile phones, we would expect speed to increase the utility of the alternatives. On the other hand, we would expect the price of the devices to decrease their utility: the higher the cost, the lower the utility derived from an alternative. In this case, the expected sign of the coefficient for price would be negative.

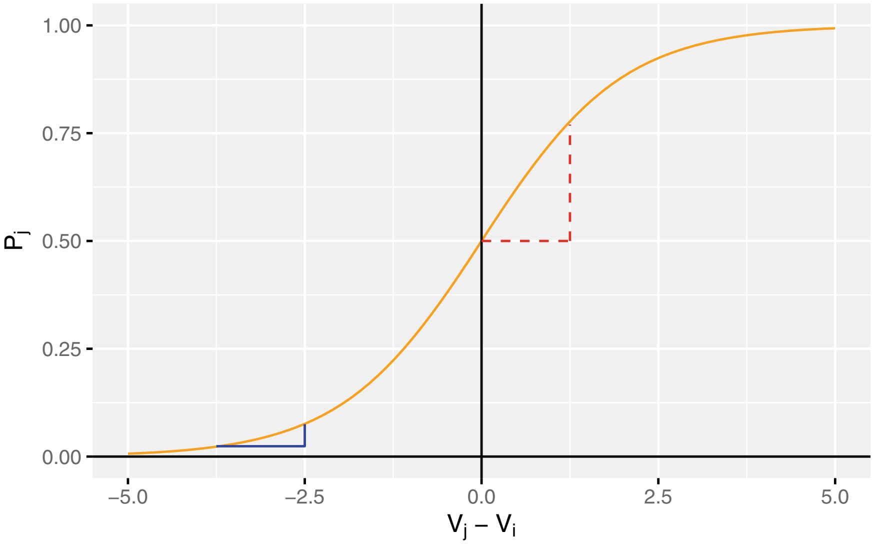


Fig. 6.1 The logit probability is not linear on the variables

Inspecting the results of Model 1 above, we notice that the sign of the coefficient for installation costs is negative; this indicates that the utility of a system tends to decrease as the installation cost increases. Not surprisingly, since central gas systems have the lowest installation costs, they are also the most popular, whereas heat pumps, with the highest installation costs, are the least popular.

If you are familiar with linear regression analysis, you are probably used to thinking of the coefficients as representing a marginal effect, that is, the impact on the outcome of changing the relevant attribute by one unit. This interpretation does not hold in the case of discrete choice models, and so beyond a qualitative assessment of the signs of the coefficients, their magnitudes are not directly interpretable (Fig. 6.1).

To understand why this is so, we must recall that underlying a logit model is a sigmoid relationship between the *differences in attributes* among utility functions and the choice probabilities. As discussed in Chap. 3, the effect on the probability of choosing an alternative, given a change on some attribute, is not constant along the logit probability curve, and in fact depends on the initial value of the variable, as seen in Fig. 4.5).

Since the coefficients are not directly interpretable, a number of different techniques have been devised to assess the behavioral implications of changes in one or more of the attributes. We will cover some relevant techniques next.

6.7 Marginal Effects

A marginal effect is a summary measure of the amount of change in a dependent variable y when an independent variable x_k changes by one unit:

$$M^{y}_{x_k} = \frac{\partial y}{\partial x_k}$$

In the case of linear models, the function that describes y is linear, and therefore the derivatives with respect to x_k are constants and collapse to the corresponding coefficients. This is not so in the case of discrete choice models, where the dependent variable that we are interested in is the probability of choosing alternative j, which is *not* linear a linear function of y. Louviere et al. (2000, pp. 58–59) show that, given a discrete choice model, the marginal effect is:

$$M^{P_{in}}_{x_{jnk}} = \beta_{jk}(\delta_{ij} - P_{jn})$$

where P_{jn} is the probability of decision-maker n selecting alternative j and β_{jk} is the coefficient that corresponds to variable x_{jnk}. This measures the change in probability as a result of a one-unit change in x_{jnk}, where k represents the attribute of interest.

Two cases result from this expression:

1. When $i = j$ then $\delta_{ii} = 1$ (this is called a direct marginal effect)
2. When $i \neq j$ then $\delta_{ij} = 0$ (this is called a cross-marginal effect)

We discuss these two cases next, but first, to continue with the example, we need to estimate the probability of choosing different systems at different levels of the installation costs. We use different levels of the installation cost because the effects will not be the same at different starting points in the logit curve!

We simulate three situations: when the costs are at their minimum, mean, and maximum. To do this, we begin by copying the input data frame. Here we copy one row from the input data frame for each of the levels that we wish to simulate:

```
ic_min <- Heating[1,]
ic_mean <- Heating[1,]
ic_max <- Heating[1,]
```

Next, we define the following vectors to retrieve the minimum, mean, and maximum installation costs for each heating system:

```
min_cost <- Heating %>%
  select(starts_with("ic")) %>%
  summarise(across(.cols = everything(),
                   min))
mean_cost <- Heating %>%
  select(starts_with("ic"))%>%
  summarise(across(.cols = everything(),
                   mean))
max_cost <- Heating %>%
  select(starts_with("ic"))%>%
  summarise(across(.cols = everything(),
                   max))
```

We now replace the cost of installation with these vectors:

```
ic_min[3:7] <- min_cost
ic_mean[3:7] <- mean_cost
ic_max[3:7] <- max_cost
```

If we quickly examine the data frame with the maximum installation costs:

```
head(ic_max)
```

```
  idcase depvar  ic.gc ic.gr  ic.ec  ic.er  ic.hp  oc.gc  oc.gr  oc.ec oc.er
1      1     gc 1158.9  1344 1230.5 1496.3 1679.8 199.69 151.72 553.34 505.6
   oc.hp income agehed rooms region
1 237.88      7     25     6 ncostl
```

We can see that we have simulated a data set that includes the maximum installation cost of every system. We did not modify any of the other variables (they are not part of the model, so they are not relevant for the simulation).

Given the different values of installation cost (at minimum, mean, and maximum), we can predict the probabilities as follows (notice that predicting with an `mlogit` model requires the new data to be in long format, so we change the wide format by means of `mlogit.data`):

```
p_model1_ic_min <- predict(model1,
                           newdata = mlogit.data(ic_min,
                                                 shape = "wide",
                                                 choice = "depvar",
                                                 varying = 3:7))
p_model1_ic_mean <- predict(model1,
                            newdata = mlogit.data(ic_mean,
                                                  shape = "wide",
                                                  choice = "depvar",
                                                  varying = 3:7))
p_model1_ic_max <- predict(model1,
                           newdata = mlogit.data(ic_max,
                                                 shape = "wide",
                                                 choice = "depvar",
                                                 varying = 3:7))
```

Note that the probabilities and the costs are not in the same order, so they need to be rearranged in the next table:

```
min_cost
```

```
    ic.gc  ic.gr  ic.ec  ic.er  ic.hp
1 431.83 574.94 469.61 546.82 532.32
```

```
p_model1_ic_min
```

```
        ec         er         gc         gr         hp
0.06956900 0.10451552 0.61683645 0.13830486 0.07077416
```

The probabilities at their corresponding costs are summarized below:

```
data.frame(System = c("Electric Central",
                      "Electric Room",
                      "Gas Central",
                      "Gas Room",
                      "Heat Pump"),
           ic_min = ic_min %>%
             # Select installation costs in the same order as the probabilities
             select(ic.ec, ic.er, ic.gc, ic.gr, ic.hp) %>%
             t() %>%
             as.numeric() %>%
             round(1),
           p_min = p_model1_ic_min %>%
             as.numeric() %>%
             round(3),
           ic_mean = ic_mean %>%
             # Select installation costs in the same order as the probabilities
             select(ic.ec, ic.er, ic.gc, ic.gr, ic.hp) %>%
             t() %>%
             as.numeric() %>%
             round(1),
           p_mean = p_model1_ic_mean %>%
             as.numeric() %>%
             round(3),
           ic_max = ic_max %>%
             # Select installation costs in the same order as the probabilities
             select(ic.ec, ic.er, ic.gc, ic.gr, ic.hp) %>%
             t() %>%
             as.numeric()  %>%
             round(1),
           p_max = p_model1_ic_max %>%
             as.numeric() %>%
             round(3)) %>%
  kable(col.names = c("System",
                      "Cost",
                      "Probability",
                      "Cost",
                      "Probability",
                      "Cost",
                      "Probability"),
        digits = 3) %>%
  kable_styling() %>%
  add_header_above(c(" " = 1,
                     "Minimum Cost" = 2,
                     "Mean Cost" = 2,
                     "Maximum Cost" = 2))
```

	Minimum Cost		Mean Cost		Maximum Cost	
System	Cost	Probability	Cost	Probability	Cost	Probability
Electric Central	469.6	0.070	824.5	0.071	1230.5	0.072
Electric Room	546.8	0.105	983.9	0.093	1496.3	0.078
Gas Central	431.8	0.617	776.8	0.639	1158.9	0.672
Gas Room	574.9	0.138	921.8	0.143	1344.0	0.140
Heat Pump	532.3	0.071	1046.5	0.055	1679.8	0.038

Of the different systems, the probability of choosing Electric Central changes the least as the cost of installation changes from the minimum cost to the mean cost of such system, and to the maximum cost. The probability of choosing Heat Pump is similar to Electric Central at its lowest installation cost, but changes substantially as we go to the maximum cost; it is true too that these systems tend to be more expensive. The probability of choosing Gas Central increases as its installation cost goes up, but this is in comparison to the maximum installation cost of every other system! Clearly, trying to understand the probabilities as the cost of every system changes is not straightforward. To understand the response of changing the installation cost by one unit at each of the three levels of cost above, *while other things remain constant*, we calculate the direct marginal effects, as shown next.

6.7.1 Direct Marginal Effects

The direct marginal effect is defined as follows:

$$M^{P_{in}}_{x_{ink}} = \beta_{ik}(1 - P_{in})$$

This measure is useful to answer the question:

> How much would the probability of choosing alternative i change if its attribute k changed by one unit?

For example, considering the gas central system, the question would be:

> How much would the probability of choosing gas centra heating system change if its installation cost changed by one unit?

Based on the values summarized above, we can calculate the direct marginal effect of the gas central system at the three levels of installation costs of interest as:

```
-0.00168108 * (1 - 0.617)
```

```
[1] -0.0006438536
```

```
-0.00168108 * (1 - 0.639)
```

```
[1] -0.0006068699
```

```
-0.00168108 * (1 - 0.672)
```

```
[1] -0.0005513942
```

The values above indicate that the probabilities of choosing a gas central system would decrease by approximately 0.064%, 0.061%, 0.055% if the installation cost increased by one dollar from the minimum, mean, and maximum installation cost, respectively.

6.7.2 Cross-Marginal Effects

We know that as the probability of choosing one alternative changes, the other probabilities change as well in response. The cross-marginal effect, which is defined as follows, is useful to investigate this effect:

$$M^{P_{in}}_{x_{jnk}} = -\beta_{jk} P_{jn}$$

This measure is useful to answer the question:

> How much would the probability of choosing alternative i change if attribute k of alternative j changed by one unit?

Or, alternatively, in terms of the present example:

> How much would the probability of choosing system [other than gas central heating] change if the installation cost of the gas central heating system changed by one unit?

Based on the values summarized above, we can calculate the cross-marginal effect of the electric central system at the min, mean, max installation costs as:

```
-(-0.00168108 * 0.617)
```

```
[1] 0.001037226
```

```
-(-0.00168108 * 0.639)
```

```
[1] 0.00107421
```

```
-(-0.00168108 * 0.672)
```

```
[1] 0.001129686
```

The values above indicate that the probabilities of choosing an electric central heating system would *increase* by approximately 0.103%, 0.107%, 0.113% if the cost of installing a gas central heating system increased by one dollar from the minimum, mean, and maximum installation cost, respectively.

6.8 Elasticity

An alternative way to explore the way a variable responds to changes in another variable is by means of the elasticity. The elasticity is a concept from economics that is useful to summarize the way a dependent variable y changes in response to changes in an independent variable x_k. This is defined as follows:

$$E^y_{x_k} = \frac{\partial y}{\partial x_k}\frac{x_k}{y}$$

It can be seen that the elasticity takes the marginal effect and makes it relative to the values of y and x_k, in effect producing a unit-less summary measure that is interpreted as the percentage change in y when attribute x_i changes by 1%.

Following a similar logic as above, there are two cases of the elasticity: direct-point elasticity and cross-point elasticity. These will be discussed next.

6.8.1 Direct-Point Elasticity

Direct-point elasticity is calculated at a point value of $x_i nk$ (attribute k of alternative i for decision-maker n) with respect to the probability of selecting alternative i. This elasticity is useful when our question about behavior is as follows:

> How much would the probability of choosing alternative i change if its attribute k changed by one percent?

In terms of the present example, for the gas central system the question would be:

> How much would the probability of choosing a gas central heating system change if its installation cost changed by one percent?

The direct-point elasticity for a discrete choice model is given by:

$$E_{x_{ink}}^{P_{in}} = \beta_{ik} x_{ink} (1 - P_{in})$$

where P_{in} is the probability of decision-maker n selecting alternative i, given a variable x_{ink} and its corresponding coefficient β_{ik}.

Based on the values summarized above, we can calculate the direct-point elasticity of the gas central system at the minimum, mean, and maximum installation costs as:

```
-0.00168108 * 431.8 * (1 - 0.617)
```

```
[1] -0.278016
```

```
-0.00168108 * 776.8 * (1 - 0.639)
```

```
[1] -0.4714165
```

```
-0.00168108 * 1158.9 * (1 - 0.672)
```

```
[1] -0.6390108
```

These values indicate that the probability of choosing the gas central system with the lowest installation cost declines by approximately 0.278% when the installation cost increases by 1%. When the installation cost is the mean and the maximum, the probability of selecting the electric central system declines by approximately 0.471% and 0.639% respectively when the cost of installation increases by 1%.

Notice that the elasticity tends to give greater values than the marginal effect. This is because a one-percent change in installation costs represents a much larger amount of change in the variable than a one dollar change.

6.8.2 *Cross-Point Elasticity*

Another useful measure of elasticity is the *cross-point elasticity*. This is a useful tool that responds to the question:

> How much would the probability of choosing alternative *i* change if attribute *k* of alternative *j* changed by one percent?

Or, for example:

> How much would the probability of choosing a system [other than gas central heating] change if the installation cost of the gas central heating system changed by one percent?

Again, Louviere et al. (2000, pp. 58–59) show that, given a discrete choice model, the cross-point elasticity is given by:

$$E^{P_{in}}_{x_{jnk}} = -\beta_{jk} x_{jnk} P_{jn}$$

where P_{in} is the probability of decision-maker n selecting alternative i, given a variable x_{jnk} and its corresponding coefficient $\beta_j k$.

The cross-point elasticities of gas central heating at the three levels of interest of installation cost are:

```
-(-0.00168108 * 431.8 * 0.617)
```

```
[1] 0.4478743
```

```
-(-0.00168108 * 776.8 *  0.639)
```

```
[1] 0.8344464
```

```
-(-0.00168108 * 1158.9 * 0.672)
```

```
[1] 1.309193
```

In other words, the probability of choosing a system other than gas central system increases by approximately 0.448%, 0.834%, 1.309% when the cost of installing a gas central system goes up by 1% from the minimum, mean, and maximum base installation costs.

6.9 Calculating Elasticities Based on an `Mlogit` Model

Fortunately, all the effects discussed above can be easily calculated once that a model has been estimated using the `mlogit` function for `effects`. This is illustrated next.

6.9.1 Computing the Marginal Effects

The marginal effects can be computed by means of the `effects` function. This function takes four arguments, as follows: an `mlogit` model, the name of the covariate (or attribute) that we wish to examine, the type of effect, and the data for calculating the effects.

For the marginal effects, the "type" of the effect is "(r)elative" for the probability and "(a)bsolute"" for the covariate. In the case of the marginal effects at the minimum values of cost we use `model1` (our `mlogit` model output object), indicate the covariate of interest (`ic` for installation cost), the type of effect ("ra"), and the input data (again, notice that the input data needs to be in long format; this is because the function needs to make predictions using the model):

```
effects(model1,
        # Calculate the marginal effects with respect to attribute "ic"
        covariate = "ic",
        # Type of effects to compute: relative for probability, absolute for attribute
        type = "ra",
        data = mlogit.data(ic_min,
                           shape = "wide",
                           choice = "depvar",
                           varying = 3:7))
```

```
              ec            er            gc            gr            hp
ec -0.0015641293  0.0001169511  0.0001169511  0.0001169511  0.0001169511
er  0.0001756990 -0.0015053814  0.0001756990  0.0001756990  0.0001756990
gc  0.0010369517  0.0010369517 -0.0006441287  0.0010369517  0.0010369517
gr  0.0002325016  0.0002325016  0.0002325016 -0.0014485788  0.0002325016
hp  0.0001189771  0.0001189771  0.0001189771  0.0001189770 -0.0015621033
```

The values on the diagonal of the table above are the direct marginal effects, and the values off the diagonal are the cross-marginal effects.

The marginal effects at the mean values (sometimes called *MEM*, i.e., marginal effects at the mean) are:

```
effects(model1,
        covariate = "ic",
        type = "ra",
        data = mlogit.data(ic_mean,
                           shape = "wide",
                           choice = "depvar",
                           varying = 3:7))
```

```
              ec            er            gc            gr            hp
ec -1.562008e-03  1.190728e-04  1.190728e-04  1.190728e-04  0.0001190728
er  1.558057e-04 -1.525275e-03  1.558056e-04  1.558056e-04  0.0001558056
gc  1.073548e-03  1.073548e-03 -6.075323e-04  1.073548e-03  0.0010735481
gr  2.399663e-04  2.399663e-04  2.399663e-04 -1.441114e-03  0.0002399663
hp  9.268762e-05  9.268762e-05  9.268762e-05  9.268762e-05 -0.0015883927
```

And the marginal effects at the maximum values are:

```
effects(model1,
        covariate = "ic",
        type = "ra",
        data = mlogit.data(ic_max,
                           shape = "wide",
                           choice = "depvar",
                           varying = 3:7))
```

```
              ec            er            gc           gr           hp
ec -1.560758e-03  1.203225e-04  1.203225e-04 0.0001203225 0.0001203225
er  1.316513e-04 -1.549429e-03  1.316514e-04 0.0001316513 0.0001316514
gc  1.129256e-03  1.129256e-03 -5.518241e-04 0.0011292563 0.0011292563
gr  2.359411e-04  2.359411e-04  2.359412e-04 -0.0014451392 0.0002359412
hp  6.390908e-05  6.390909e-05  6.390909e-05 0.0000639091 -0.0016171713
```

You can verify that, with some small rounding error, they correspond to the values we calculated previously.

6.9.2 *Computing the Elasticities*

As well, the direct-point and cross-point elasticities can be calculated using the `effects` function, however in this case the type of the effects is *relative* both for the probability and for the covariate (type =“rr”).

```
effects(model1,
        covariate = "ic",
        type = "rr",
        data = mlogit.data(ic_min,
                           shape = "wide",
                           choice = "depvar",
                           varying = 3:7))
```

```
            ec          er          gc          gr          hp
ec -0.73453075  0.05492140  0.05492140  0.05492140  0.05492140
er  0.09607572 -0.82317265  0.09607573  0.09607572  0.09607572
gc  0.44778685  0.44778685 -0.27815410  0.44778684  0.44778685
gr  0.13367447  0.13367447  0.13367448 -0.83284588  0.13367448
hp  0.06333387  0.06333387  0.06333387  0.06333386 -0.83153884
```

The values on the diagonal of the table above are the direct-point elasticities, whereas other values are the cross-point elasticities.

The effects can be calculated at various levels of the covariate of interest, for instance the mean and max:

```
effects(model1,
        covariate = "ic",
        type = "rr",
        data = mlogit.data(ic_mean,
                           shape = "wide",
                           choice = "depvar",
                           varying = 3:7))
```

```
            ec          er          gc          gr          hp
ec -1.28794317  0.09818068  0.09818067  0.09818069  0.09818068
er  0.15330155 -1.50076056  0.15330153  0.15330154  0.15330154
gc  0.83396070  0.83396069 -0.47194728  0.83396070  0.83396070
gr  0.22119376  0.22119375  0.22119375 -1.32837610  0.22119376
hp  0.09699586  0.09699586  0.09699586  0.09699586 -1.66222324
```

```
effects(model1,
        covariate = "ic",
        type = "rr",
        data = mlogit.data(ic_max,
                           shape = "wide",
                           choice = "depvar",
                           varying = 3:7))
```

```
           ec         er         gc         gr         hp
ec -1.9205126  0.1480568  0.1480568  0.1480568  0.1480568
er  0.1969899 -2.3184107  0.1969899  0.1969899  0.1969899
gc  1.3086951  1.3086951 -0.6395089  1.3086951  1.3086951
gr  0.3171049  0.3171049  0.3171049 -1.9422671  0.3171049
hp  0.1073545  0.1073545  0.1073545  0.1073545 -2.7165243
```

6.10 A Note About Attributes in Dummy Format

The attribute used above for the example was installation cost, a continuous variable. In many cases, the attributes of alternatives or decision-makers are continuous variables, but not always. In the data set about heating systems, for example, there is a dummy variable that indicates the region of residence of the decision-maker:

```
summary(Heating$region)
```

```
valley scostl mountn ncostl
   177    361    102    260
```

As can be seen, there are four regions. In the sample, 177 respondents lived in the region labeled as “valley”, 361 in the south coastal (“scostl”), 102 in the mountain (“mountn”), and 260 in north coastal (“ncostl”).

The marginal effects and elasticities discussed above are not appropriate for use when it comes to dummy variables. The reason for this is that marginal changes are not meaningful (e.g., what does it mean to increase region “mountain” by one unit or by 1%?)

When variable x_{ink} is a dummy variable, the marginal effect must be calculated as follows:

$$M_{x_{nk}}^{P_{in}} = P_{in}(x_{nk} = 1) - P_{in}(x_{nk} = 0)$$

In other words, the marginal effect is the difference in the probability of choosing i when the dummy variable is 1 versus when it is 0. In this case, there is no direct or cross marginal effect, as the attribute does not vary according to the alternatives (Table 6.3).

To illustrate this, we will estimate a second model that uses the variable with the regions (call this Model 2):

```
model2 <- mlogit(depvar ~ ic | region,
                 Heating,
                 shape = "wide",
                 choice = "depvar",
                 reflevel = "ec",
                 varying = c(3:7))

stargazer::stargazer(model2,
                     header = FALSE,
                     single.row = TRUE,
                     title = "Estimation results: Model 2")
```

Notice that none of the regional coefficients is significant, so we would likely not include these variables in the model. However, for the sake of the example, we will calculate the marginal effect of the dummy variables at the mean of installation cost.

We had already created a matrix with the mean of the installation cost. To add the four regions, we will copy the row four times, and then add the regions:

```
ic_mean_region <- ic_mean %>%
  mutate(count = 4) %>%
  uncount(count)
ic_mean_region$region <- c("valley",
                           "scostl",
                           "mountn",
                           "ncostl")
head(ic_mean_region)
```

Table 6.3 Estimation results: Model 2

	Dependent variable
	Depvar
(Intercept):er	0.600 (0.378)
(Intercept):gc	2.030*** (0.295)
(Intercept):gr	0.896*** (0.343)
(Intercept):hp	0.281 (0.422)
ic	−0.002*** (0.001)
regionscostl:er	0.050 (0.447)
regionscostl:gc	0.074 (0.362)
regionscostl:gr	0.103 (0.414)
regionscostl:hp	−0.145 (0.501)
regionmountn:er	−0.022 (0.591)
regionmountn:gc	−0.086 (0.478)
regionmountn:gr	0.036 (0.546)
regionmountn:hp	−0.039 (0.655)
regionncostl:er	−0.338 (0.494)
regionncostl:gc	0.232 (0.384)
regionncostl:gr	−0.330 (0.455)
regionncostl:hp	−0.399 (0.554)
Observations	900
R^2	0.009
Log Likelihood	−1,012.524
LR Test	19.400 (df = 17)

Note—*p<0.1; **p<0.05; ***p<0.01

```
      idcase depvar    ic.gc    ic.gr    ic.ec    ic.er    ic.hp  oc.gc  oc.gr
1...1      1     gc 776.8266 921.7702 824.5435 983.928 1046.481 199.69 151.72
1...2      1     gc 776.8266 921.7702 824.5435 983.928 1046.481 199.69 151.72
1...3      1     gc 776.8266 921.7702 824.5435 983.928 1046.481 199.69 151.72
1...4      1     gc 776.8266 921.7702 824.5435 983.928 1046.481 199.69 151.72
       oc.ec oc.er  oc.hp income agehed rooms region
1...1 553.34 505.6 237.88      7     25     6 valley
1...2 553.34 505.6 237.88      7     25     6 scostl
1...3 553.34 505.6 237.88      7     25     6 mountn
1...4 553.34 505.6 237.88      7     25     6 ncostl
```

We then calculate the probabilities associated with each region and heating system. Notice that for each region, the sum of the probabilities is equal to 1.

```
p_region_ic_mean <- data.frame(Region = c("valley",
                                          "scostl",
                                          "mountn",
                                          "ncostl"),
                               predict(model2,
                                       newdata = mlogit.data(ic_mean_region,
                                                             shape = "wide",
                                                             choice = "depvar",
                                                             varying = 3:7),
                                       outcome = FALSE))
p_region_ic_mean
```

```
  Region         ec         er        gc        gr         hp
1 valley 0.06893601 0.06868371 0.7167905 0.1032603 0.04232956
2 scostl 0.07709246 0.10537150 0.5832442 0.1664375 0.06785436
3 mountn 0.07330246 0.10244901 0.6044788 0.1526602 0.06710956
4 ncostl 0.06922558 0.10166470 0.6145323 0.1597530 0.05482440
```

The first row contains the probabilities of the different systems for valley (the reference region), and then for each of the three other regions.

The marginal effects of changing from the valley to other regions are:

```
data.frame (Effect = c("valley to scostl",
                       "valley to mountn",
                       "valley to ncostl"),
            rbind (p_region_ic_mean[2, 2:6] - p_region_ic_mean[1, 2:6],
                   p_region_ic_mean[3, 2:6] - p_region_ic_mean[1, 2:6],
                   p_region_ic_mean[4, 2:6] - p_region_ic_mean[1, 2:6]))
```

```
            Effect           ec         er         gc         gr         hp
2 valley to scostl 0.0081564496 0.03668779 -0.1335463 0.06317722 0.02552480
3 valley to mountn 0.0043664503 0.03376531 -0.1123117 0.04939991 0.02478000
4 valley to ncostl 0.0002895745 0.03298099 -0.1022582 0.05649277 0.01249484
```

6.11 Willingness to Pay and Discount Rate

Another interesting question of policy relevance is the rate at which consumers are willing to trade off one attribute for another. In terms of the present example, this question could be:

> How much are consumers willing to pay in increased installation costs in exchange for lower annual operating costs?

To answer this question, we first need an appropriate choice model, that is, one that includes both installation *and* annual operation costs. The following utility functions are one possible model specification (with electric central as the reference level):

Table 6.4 Estimation results: model 3

	Dependent variable
	Depvar
(Intercept):er	0.195 (0.204)
(Intercept):gc	0.052 (0.466)
(Intercept):gr	−1.351*** (0.507)
(Intercept):hp	−1.659*** (0.448)
ic	−0.002** (0.001)
oc	−0.007*** (0.002)
Observations	900
R^2	0.014
Log Likelihood	−1,008.229
LR Test	27.990*** (df = 6)

Note—*p<0.1; **p<0.05; ***p<0.01

$$\begin{array}{rcll}
V_{\text{ec}} & = & 0 & +\beta_1\text{ic.ec} + \beta_2\text{oc.ec}\\
V_{\text{er}} & = & \beta_{er} & +\beta_1\text{ic.er} + \beta_2\text{oc.er}\\
V_{\text{gc}} & = & \beta_{gc} & +\beta_1\text{ic.gc} + \beta_2\text{oc.gc}\\
V_{\text{gr}} & = & \beta_{gr} & +\beta_1\text{ic.gr} + \beta_2\text{oc.gr}\\
V_{\text{hp}} & = & \beta_{hp} & +\beta_1\text{ic.hp} + \beta_2\text{oc.hp}
\end{array}$$

The model can be estimated again using these constants (Table 6.4). The option `reflevel` is used to select the alternative that will work as reference (call this Model 3; notice that the columns that vary need to be updated to include the new alternative-related variable):

```r
model3 <- mlogit(depvar ~ ic + oc,
                 Heating,
                 shape = "wide",
                 choice = "depvar",
                 reflevel = "ec",
                 varying = c(3:12))

stargazer::stargazer(model3,
                     header = FALSE,
                     single.row = TRUE,
                     title = "Estimation results: Model 3")
```

Now, how can this model be used to understand consumer preferences? Recall that the question is the trade-off between the cost of installation relative to operation.

The coefficients of the model provide useful information. Suppose that we would like to know the rate at which consumers prefer to trade one aspect of the good for another, without compromising the utility they derive from it. In effect we would like to know how changes in one attribute relate to changes in the other as long as the utility remains unchanged. In the present case, we want to know how much consumers are willing to pay in installation if it reduced the cost of operation, while maintaining the utility (i.e., the change in utility is zero):

$$\partial U = 0 = \beta_1 \partial ic + \beta_2 \partial oc$$

It follows then that:

$$-\frac{\partial ic}{\partial oc} = \frac{\beta_2}{\beta_1}$$

The ratio of the coefficients represents the *willingness to pay*. In this example, since:

$$-\frac{\partial ic}{\partial oc} = \frac{\beta_2}{\beta_1} = \frac{-0.0069}{-0.0015} = 4.56$$

The willingness to pay is an additional 4.56 dollars in installation cost per every dollar saved in operation cost per year. The discount rate is:

$$r = \frac{1}{4.56} = 0.219 = 21.9\%$$

This information can be used to assess the behavior of consumers.

6.12 Simulating Market Changes

To the extent that random utility models can capture consumer preferences, they are useful to understand patterns of substitution. Once a model has been estimated, simulating market changes (e.g.: increase in cost, reduction in quality, etc.) involves creating a new data matrix to which the model can be applied. We will now take a look at two examples.

6.12.1 Incentives

Heat pumps are on average more expensive to install than other heating systems, but they are also more energy efficient. Suppose then that the government, which has perhaps carbon emission targets that it wishes to meet, is analyzing a policy to encourage the adoption of heat pumps. The policy is to offer a rebate of 15% on the installation cost of heat pumps. As a consequence of this policy, consumers who install a heat pump and apply for the rebate pay only 85% of the cost of installation.

To simulate this scenario, we copy the input data frame after simulating the rebate, replacing the cost of installation as follows:

```
H_rebate <- Heating %>%
  mutate(ic.hp = 0.85 * ic.hp)
```

We can calculate the market shares of the “do nothing” (with the Heating data set) and “rebate” (with the H_rebate data set) policies and compare their shares (which are the mean values of the predictions):

```
data.frame(Policy = c("Do nothing", "15% rebate"),
           rbind(apply(predict(model3,
                               newdata = mlogit.data(Heating,
                                                     shape = "wide",
                                                     choice = "depvar",
                                                     varying = c(3:12))),
                       2,
                       mean),
                 apply(predict(model3,
                               newdata = mlogit.data(H_rebate,
                                                     shape = "wide",
                                                     choice = "depvar",
                                                     varying = c(3:12))),
                       2,
                       mean)))
```

```
      Policy         ec         er        gc        gr         hp
1 Do nothing 0.07111111 0.09333333 0.6366667 0.1433333 0.05555556
2 15% rebate 0.07009125 0.09199196 0.6272957 0.1412092 0.06941196
```

In this example, we see that the market share (or mean probability of selecting a heat pump) increases with the 15% rebate, while the market shares of other heating systems slightly decrease. This is in line with the policy objectives of the rebate on the installation of heat pumps. The magnitude of the change is informative to assess whether the rebate can achieve the policy objective.

6.12.2 Introduction of a New System

Now suppose that a more efficient electric system is developed using newer technologies. The cost of installation is more expensive due to the cost of the new technology (cost of installation is $200 dollars higher than the electric central system). On the other hand, the cost of operation is only 75% that of the electric central systems. The preceding analysis suggests that consumers are willing to spend more in installation costs in exchange for savings in operation costs. What would the market penetration of this new system be?

To simulate this situation, we begin by creating a model matrix based on the output of Model 3:

```
X <- model.matrix(model3)
head(X)
```

```
  (Intercept):er (Intercept):gc (Intercept):gr (Intercept):hp      ic     oc
1              0              0              0              0  859.90 553.34
2              1              0              0              0  995.76 505.60
3              0              1              0              0  866.00 199.69
4              0              0              1              0  962.64 151.72
5              0              0              0              1 1135.50 237.88
6              0              0              0              0  796.82 520.24
```

Then, we create a new alternative by copying the attributes of electric central. In other words, we create a new matrix in which we keep only the rows associated with electric central:

```
alt <- index(H)$alt
Xn <- X[alt == "ec",]
head(Xn)
```

```
   (Intercept):er (Intercept):gc (Intercept):gr (Intercept):hp     ic     oc
1               0              0              0              0 859.90 553.34
6               0              0              0              0 796.82 520.24
11              0              0              0              0 719.86 439.06
16              0              0              0              0 761.25 483.00
21              0              0              0              0 858.86 404.41
26              0              0              0              0 693.74 398.22
```

Next, we modify the attributes to reflect the attributes of the new system (+$200 to ic and 0.75 of oc):

```
Xn[, "ic"] <- Xn[, "ic"] + 200
Xn[, "oc"] <- Xn[, "oc"] * 0.75
head(Xn)
```

```
   (Intercept):er (Intercept):gc (Intercept):gr (Intercept):hp      ic       oc
1               0              0              0              0 1059.90 415.0050
6               0              0              0              0  996.82 390.1800
11              0              0              0              0  919.86 329.2950
16              0              0              0              0  961.25 362.2500
21              0              0              0              0 1058.86 303.3075
26              0              0              0              0  893.74 298.6650
```

We also want to identify the unique choice ids (in other words, the unique identifiers combining the household number and type of heating system), which we will add as row names to the new system.

```
chid <- index(H)$chid
head(chid, 12)
```

```
 [1] 1 1 1 1 1 2 2 2 2 2 3 3
```

```
unchid <- unique(index(H)$chid)
head(unchid, 12)
```

```
 [1]  1  2  3  4  5  6  7  8  9 10 11 12
```

```
rownames(Xn) <- paste(unchid, 'new', sep = ".")
chidb <- c(chid, unchid)
head(Xn)
```

```
      (Intercept):er (Intercept):gc (Intercept):gr (Intercept):hp      ic
1.new              0              0              0              0 1059.90
2.new              0              0              0              0  996.82
3.new              0              0              0              0  919.86
4.new              0              0              0              0  961.25
5.new              0              0              0              0 1058.86
6.new              0              0              0              0  893.74
            oc
1.new 415.0050
2.new 390.1800
3.new 329.2950
4.new 362.2500
5.new 303.3075
6.new 298.6650
```

After this, we can join the new system to the model matrix and sort by choice id:

```
X <- rbind(X, Xn)
X <- X[order(chidb), ]
head(X,15)
```

```
      (Intercept):er (Intercept):gc (Intercept):gr (Intercept):hp      ic
1                  0              0              0              0  859.90
2                  1              0              0              0  995.76
3                  0              1              0              0  866.00
4                  0              0              1              0  962.64
5                  0              0              0              1 1135.50
1.new              0              0              0              0 1059.90
6                  0              0              0              0  796.82
7                  1              0              0              0  894.69
8                  0              1              0              0  727.93
9                  0              0              1              0  758.89
10                 0              0              0              1  968.90
2.new              0              0              0              0  996.82
11                 0              0              0              0  719.86
12                 1              0              0              0  900.11
13                 0              1              0              0  599.48
           oc
1     553.340
2     505.600
3     199.690
4     151.720
5     237.880
```

```
1.new 415.005
6     520.240
7     486.490
8     168.660
9     168.660
10    199.190
2.new 390.180
11    439.060
12    404.740
13    165.580
```

Alas, the `predict()` function expects only the original alternatives in the inputs, and cannot handle a new alternative. For this reason we need to calculate the probabilities manually. The following chunk of code calculates the expression $e^{X\beta}$ and the sum, which are needed to compute the logit probabilities:

```
exp_Xb <- as.numeric(exp(X %*% coef(model3))) # vectors
head(exp_Xb)
```

```
[1] 0.005573519 0.007677778 0.069066744 0.020487382 0.006319992 0.010796680
```

```
# tapply does the sum of th exp_Xb for each chidb
sum_exp_Xb <- as.numeric(tapply(exp_Xb,
                                sort(chidb),
                                sum))
```

This is the vector of logit probabilities:

```
P <- exp_Xb / sum_exp_Xb[sort(chidb)]
```

We now convert to a matrix of logit probabilities, so that each row is the choice probabilities for a household:

```
P <- data.frame(matrix(P,
                       ncol = 6,
                       byrow = TRUE))
P <- transmute(P,
               ec = P[, 1],
               er = P[, 2],
               gc = P[, 3],
               gr = P[, 4],
               hp = P[, 5],
               new = P[, 6])
```

We can verify that the sum of the probabilities for each household is 1:

```
summary(rowSums(P))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      1       1       1       1       1       1
```

The estimated penetration of the new system is the average probability of households choosing this system:

```
apply(P, 2, mean)
```

```
        ec         er         gc         gr         hp        new 
0.06311578 0.08347713 0.57145108 0.12855080 0.04977350 0.10363170 
```

The new technology is estimated to have a penetration rate of approximately 10.4%. If we compare these estiated penetration rates to the original market shares of the systems:

```
apply(fitted(model3,
             outcome = FALSE),
      2,
      mean)
```

```
        ec         er         gc         gr         hp 
0.07111111 0.09333333 0.63666667 0.14333333 0.05555556 
```

Can you discern the patterns of substitution here? Are these patterns of substitution reasonable?

6.13 Simulating Individual-Level Outcomes

The simulation techniques discussed above aggregate the individual-level probabilities to give market-level shares. Sometimes, however, we might be curious about the individual level outcomes.

Let us return to Model 3. We saw before that the estimated probabilities can be retrieved from the model using function `predict()` or manually. The probabilities may be converted to predicted alternative using heuristics. These are the predicted probabilities according to Model 3:

```
p_model3 <- fitted(model3,
                   outcome = FALSE) %>%
  data.frame()
```

As we just saw, these can be used to obtain the predicted market shares by calculating the mean probabilities:

```
apply(p_model3,
      2,
      mean)
```

```
        ec         er         gc         gr         hp 
0.07111111 0.09333333 0.63666667 0.14333333 0.05555556 
```

We could use a heuristic rule to predict the chosen alternative, say by assigning the outcome to the alternative with the highest probability. This heuristic is implemented in this chunk of code:

```
o_model3 <- p_model3 %>%
  # Group by row
  rowwise() %>%
  # Find the maximum value by row
  mutate(max_p = max(c(ec, er, gc, gr, hp))) %>%
  ungroup() %>%
  # Find the column that matches the highest probability
  transmute(outcome = case_when(max_p == ec ~ "ec",
                                max_p == er ~ "er",
                                max_p == gc ~ "gc",
                                max_p == gr ~ "gr",
                                max_p == hp ~ "hp"))
```

How frequently are the various alternatives chosen, according to this heuristic?

```
table(o_model3)
```

```
outcome
 gc 
900 
```

The heuristic returns something implausible, namely that all respondents choose only gas central system. We know that this system is the most popular, and the logit probabilities are consistently high for this alternative. In fact, they are the highest probabilities in every row! But high probability is not destiny, and we see that trying to predict individual level outcomes can be problematic, especially when one alternative has consistently high logit probabilities (i.e., greater than 0.5), or when some alternatives have consistently low probabilities, since the heuristic will never pick them up as the chosen alternative.

It is important to keep this in mind, because predicting individual-level outcomes may give the impression that the model performs poorly, when the market-level predictions may in fact be quite good.

6.14 Exercises

1. What is the difference between a marginal effect and an elasticity?
2. Why is it inappropriate to calculate the elasticity of a dummy variable?
3. Use Model 3 in this chapter and calculate the marginal effects and the elasticities for operating cost at the mean of all variables.
4. Use Model 3 in this chapter to calculate the rebate needed to reach a 10% penetration rate of heat pumps.

Estimate a new model that extends Model 3 by introducing the age of the household head. Use the electric room system ("er") as the reference level.

5. Use the likelihood ratio test to compare your new model to Model 3. Discuss the results.
6. Is the ratio of the coefficient of installation (or operation) cost to the coefficient of age of household head meaningful? Explain.

References

King, J. L., & Kraemer, K. L. (1993). Models, facts, and the policy process: The political ecology of estimated truth. In M. Goodchild, B. O. Parks, & L. T. Steyaert (Eds.), *Environmental modelling with GIS*. Oxford: Oxford University Press.

Louviere, J. J., Hensher, D. A., & Swait, J. D. (2000). *Stated choice methods: Analysis and applications*. Cambridge: Cambridge University Press.

McFadden, D. (1974). The measurement of urban travel demand. *Journal of Public Economics, 3*(4), 303–28.

Train, K., & Croissant, Y. (2012). *Kenneth Train's Exercises Using the Mlogit Package for r*. CiteSeerX: Journal Article.

Chapter 7
Non-proportional Substitution Patterns I: Generalized Extreme Value Models

Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away.

— Antoine de Saint-Exupéry, Airman's Odyssey

Perfection is a stick with which to beat the possible.

— Rebecca Solnit, Hope in the Dark

The maxim, 'Nothing prevails but perfection,' may be spelled PARALYSIS.

—Winston S. Churchill

7.1 The Limits of Perfection

The multinomial logit model is the workhorse of discrete choice analysis. As seen in the preceding chapters, it is a model that is intuitive, and moreover, its closed analytical form makes it simple and convenient to estimate.

When originally developed by Luce (1959; cited in Train 2009), the logit model implicitly carried the axiom of Independence of Irrelevant Alternatives. In Chap. 4, we noted that this property of the logit model implies *proportional substitution patterns*, which means that the choice probabilities of substitutes change in proportion to a constant when the attributes of an alternative change.

At the end of Chap. 6, we asked whether the patterns of substitution were sensible after a new heating system was introduced. Let us revisit the model estimated there to answer this question. In the original choice problem, there were five different heating system, namely Gas Central (gc), Gas Room (gr), Electric Central (ec), Electric Room (er), and Heat Pump (hp).

A. Páez and G. Boisjoly, *Discrete Choice Analysis with R*, Use R!,
https://doi.org/10.1007/978-3-031-20719-8_7

We will begin this example by loading the following packages:

```
library(dplyr) # A Grammar of Data Manipulation
library(ggplot2) # Create Elegant Data Visualisations Using the Grammar of Graphics
library(kableExtra) # Construct Complex Table with 'kable' and Pipe Syntax
library(mlogit) # Multinomial Logit Models
```

In addition, we also need to load the data set used for the example (from the `mlogit` package)

```
data("Heating",
     package = "mlogit")
```

The data set is in “wide” form, which means that there is one record per decision-making unit (i.e., per household). We proceed to change the shape of the table to “long” format (Table 7.1)

```
H <- mlogit.data(Heating,
                 shape = "wide",
                 choice = "depvar",
                 varying = c(3:12))
```

Now we are ready to estimate a multinomial logit model (this was called Model 3 in Chap. 6)

```
model3 <- mlogit(depvar ~ ic + oc,
               Heating,
               shape = "wide",
               choice = "depvar",
               reflevel = "ec",
               varying = c(3:12))

stargazer::stargazer(model3,
                     header = FALSE,
                     single.row = TRUE,
                     title = "Estimation results: Model 3")
```

To explore the patterns of substitution according to this model, we will simulate the adoption rates of the systems after removing one system at the time.

To simulate this situation, we begin by retrieving the model matrix used to estimate Model 3

```
X <- model.matrix(model3)
head(X)
```

```
  (Intercept):er (Intercept):gc (Intercept):gr (Intercept):hp      ic     oc
1              0              0              0              0  859.90 553.34
2              1              0              0              0  995.76 505.60
3              0              1              0              0  866.00 199.69
4              0              0              1              0  962.64 151.72
5              0              0              0              1 1135.50 237.88
6              0              0              0              0  796.82 520.24
```

Table 7.1 Estimation results: model 3

	Dependent variable
	Depvar
(Intercept):er	0.195 (0.204)
(Intercept):gc	0.052 (0.466)
(Intercept):gr	−1.351*** (0.507)
(Intercept):hp	−1.659*** (0.448)
ic	−0.002** (0.001)
oc	−0.007*** (0.002)
Observations	900
R^2	0.014
Log Likelihood	−1,008.229
LR Test	27.990*** (df = 6)

Note—*p<0.1; **p<0.05; ***p<0.01

Based on this, we can create 5 new model matrices, each of which contains one fewer system (in effect we are removing an alternative; imagine that one particular system has been phased out or discontinued in each case):

```r
alt <- index(H)$alt
Xmec <- X[alt != "ec",]
Xmer <- X[alt != "er",]
Xmgc <- X[alt != "gc",]
Xmgr <- X[alt != "gr",]
Xmhp <- X[alt != "hp",]
```

After doing this, we also want to identify the unique choice ids to reduce the number of choice situations for each decision-maker (from five alternatives to four)

```r
# Unique identifiers by decision-maker
chid <- index(H)$chid
# Remove the fifth identifier for each decision-maker
chid <- chid[-seq(1, length(chid), 5)]
```

With these preparations, we are now ready to calculate the expression $e^{X\beta}$ and the sum, which are needed to compute the logit probabilities[1]

```r
# After removing ec
exp_Xb_mec <- as.numeric(exp(Xmec %*% coef(model3)))
sum_exp_Xb_mec <- as.numeric(tapply(exp_Xb_mec, sort(chid), sum))
P_mec <- exp_Xb_mec / sum_exp_Xb_mec[sort(chid)]

# After removing er
```

[1] The function `predict()` expects all the original alternatives used for estimation, but since we are altering the choice set, we need to calculate the probabilities manually.

```
exp_Xb_mer <- as.numeric(exp(Xmer %*% coef(model3)))
sum_exp_Xb_mer <- as.numeric(tapply(exp_Xb_mer, sort(chid), sum))
P_mer <- exp_Xb_mer / sum_exp_Xb_mer[sort(chid)]

# After removing gc
exp_Xb_mgc <- as.numeric(exp(Xmgc %*% coef(model3)))
sum_exp_Xb_mgc <- as.numeric(tapply(exp_Xb_mgc, sort(chid), sum))
P_mgc <- exp_Xb_mgc / sum_exp_Xb_mgc[sort(chid)]

# After removing gr
exp_Xb_mgr <- as.numeric(exp(Xmgr %*% coef(model3)))
sum_exp_Xb_mgr <- as.numeric(tapply(exp_Xb_mgr, sort(chid), sum))
P_mgr <- exp_Xb_mgr / sum_exp_Xb_mgr[sort(chid)]

# After removing hp
exp_Xb_mhp <- as.numeric(exp(Xmhp %*% coef(model3)))
sum_exp_Xb_mhp <- as.numeric(tapply(exp_Xb_mhp, sort(chid), sum))
P_mhp <- exp_Xb_mhp / sum_exp_Xb_mhp[sort(chid)]
```

Now we can convert the vector of logit probabilities to a matrix, in such a way that each row contains the choice probabilities for a single decision-making unit (i.e., a household)

```
# After removing ec
P_mec <- data.frame(matrix(P_mec, ncol = 4, byrow = TRUE))
P_mec <- transmute(P_mec,
                   # Remove this alternative from the choice set
                   ec = NA,
                   er = P_mec[, 1],
                   gc = P_mec[, 2],
                   gr = P_mec[, 3],
                   hp = P_mec[, 4])

# After removing er
P_mer <- data.frame(matrix(P_mer, ncol = 4, byrow = TRUE))
P_mer <- transmute(P_mer,
                   ec = P_mer[, 1],
                   # Remove this alternative from the choice set
                   er = NA,
                   gc = P_mer[, 2],
                   gr = P_mer[, 3],
                   hp = P_mer[, 4])

# After removing gc
P_mgc <- data.frame(matrix(P_mgc, ncol = 4, byrow = TRUE))
P_mgc <- transmute(P_mgc,
                   ec = P_mgc[, 1],
                   er = P_mgc[, 2],
                   # Remove this alternative from the choice set
                   gc = NA,
                    gr = P_mgc[, 3],
                    hp = P_mgc[, 4])

# After removing gr
P_mgr <- data.frame(matrix(P_mgr, ncol = 4, byrow = TRUE))
P_mgr <- transmute(P_mgr,
                   ec = P_mgr[, 1],
                   er = P_mgr[, 2],
                   gc = P_mgr[, 3],
                   # Remove this alternative from the choice set
                   gr = NA,
                   hp = P_mgr[, 4])

# After removing hp
P_mhp <- data.frame(matrix(P_mhp, ncol = 4, byrow = TRUE))
P_mhp <- transmute(P_mhp,
                   ec = P_mhp[, 1],
                   er = P_mhp[, 2],
                   gc = P_mhp[, 3],
                   gr = P_mhp[, 4],
                   # Remove this alternative from the choice set
                   hp = NA)
```

Given the above, it is possible to summarize the choice probabilities in the form of adoption rates (which are simply the means of the probabilities for each alternative). The table below shows the original adoption rates (when all alternatives were present) and for the different situations of interest, after removing each alternative

```
df <- data.frame(Alternative = c("None", "ec", "er", "gc", "gr", "hp" ),
  rbind(apply(fitted(model3,
                     outcome = FALSE),
              2, mean),
        apply(P_mec, 2, mean),
        apply(P_mer, 2, mean),
        apply(P_mgc, 2, mean),
        apply(P_mgr, 2, mean),
        apply(P_mhp, 2, mean))
)

df %>%
  kable(col.names = c("Alternative Removed",
                      "ec",
                      "er",
                      "gc",
                      "gr",
                      "hp"),
        digits = 3) %>%
  kable_styling()
```

Alternative Removed	ec	er	gc	gr	hp
None	0.071	0.093	0.637	0.143	0.056
ec	NA	0.101	0.685	0.154	0.060
er	0.079	NA	0.702	0.158	0.061
gc	0.193	0.253	NA	0.400	0.154
gr	0.083	0.109	0.743	NA	0.065
hp	0.075	0.099	0.674	0.152	NA

Examine for a moment the patterns of substitution when alternatives are removed. In general, the probability of choosing any of the remainder alternatives increases by the same percentage (recall the elasticities), which means that the alternatives that had a higher probability of being chosen to begin with will increase more. Consider, for instance, what happens when Gas Central is removed. Would you say that this pattern of substitution is sensible?

At issue is the possibility that there are hidden correlations between *some* alternatives. This does not mean that the method is flawed; instead, this means that the model was not properly specified. The logit model is the ideal modeling alternative, as long as the utilities are reasonably well-specified. Unfortunately, for us, there is no way to say, a priori, whether some alternatives are correlated given a model specification.

For instance, in the present example, it is plausible to think that central systems (i.e., Electric Central, Gas Central, Heat Pump) are more similar between them than room systems (i.e., Electric Room, Gas Room). If the specification of the model is not complete and some attribute common to central systems is missing, we would expect the patterns of substitution between those systems to be stronger than between central and room systems. On the other hand, it is possible that the source of energy makes some systems more similar between them, for example, Electric Central, Electric Room, and Heat Pump, whereas Gas Central and Gas Room might be more similar between them.

A way to capture these "hidden" similarities between modes is by specifying a branching structure whereby alternatives thought to share common but missing attributes are grouped. Three examples of choice structures are shown in Fig. 7.1, including a multinomial choice structure, such as in the model above, and two plausible nested choice structures.

The intuition behind the nested choice structures is that decision-makers decide first whether they want an electric- or gas-based system, before deciding which

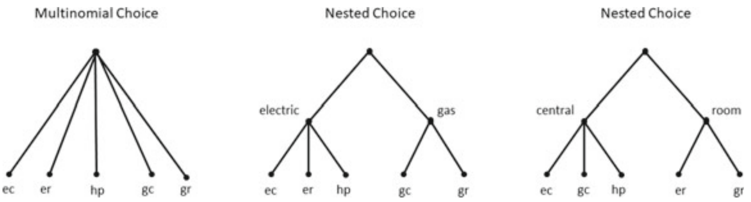


Fig. 7.1 Three examples of choice structures

system specifically; or, as an alternative, that decision-makers decide whether they want a room or central system, before deciding on a specific energy source. It is important to highlight that if the systems were highly differentiated through the specification of the utility functions, the nesting structure would be unnecessary.

The question of whether the patterns of substitution should be non-proportional is an empirical one, and the objective of this chapter is to introduce a class of models that are useful to capture non-proportional substitution patterns based on the Extreme Value distribution used to derive the multinomial logit model.

7.2 How to Use This Note

Every chunk of code, including the following, is available for practice purposes:

```
print("Do NOT believe a word Gwyneth Paltrow says. Repeat. DO NOT BELIEVE HER.")
```

```
[1] "Do NOT believe a word Gwyneth Paltrow says. Repeat. DO NOT BELIEVE HER."
```

If you are working with RStudio you can type the chunks of code to experiment with them. As an alternative, you may copy and paste the source code into your `R` or RStudio console, or create a script/notebook to save the code and any experiments you may conduct.

7.3 Learning Objectives

In this chapter, you will learn about:

1. The Generalized Extreme Value approach to derive models.
2. The nested logit model.
3. Properties of the nested logit model.
4. Patterns of substitution with the nested logit model.

7.4 Suggested Readings

- Ben-Akiva, M., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and applications to travel demand* (Chap. 5, pp. 126–128 and Chap. 10, pp. 285–299). MIT Press.
- Hensher, D. A., Rose, J. M., & Greene, W. H (2005). *Applied choice analysis: A primer* (Chaps. 13 and 14). Cambridge University Press.

- Louviere, J. J., Hensher, D. A., & Swait, J. D. (2000). *Stated choice methods: Analysis and application* (Chap. 6). Cambridge University Press.
- Ortuzar, J .D., & Willumsen L. G. (2011). *Modelling transport*, 4th edn. (Chap. 2, pp. 43-44 and). Wiley.
- Train, K. (2009). *Discrete choice methods with simulation*, 2nd edn. (Chap. 4). Cambridge University Press.

7.5 Generalized Extreme Value Models: A Recipe for Deriving Discrete Choice Models

McFadden's Generalized Extreme Value model [or GEV, for short; McFadden (1978)] is a general approach to derive discrete choice models. In this sense, it is not a single model, but rather a recipe for deriving discrete choice models based on the Extreme Value distribution. The definition of the model is discussed here, along with a preliminary example (see Ben-Akiva and Lerman 1985, pp. 126–27). Then, we will see how the GEV recipe can be used to derive models with nested structures such as shown in Fig. 7.1.

For simplicity, we will postulate that

$$y_j = e^{V_j} \text{ , for } j = 1, 2, \dots, J$$

where J is the number of alternatives. Notice that

$$y_1 \geq 0, y_2 \geq 0, \dots, y_J \geq 0$$

We will define a function G as follows:

$$G(y_1, y_2, \dots, y_J)$$

Function G must satisfy the following conditions:

1. G is non-negative.
2. G is a homogeneous function of degree one. This means that multiplying every y_j by a factor α is identical to multiplying G by α

$$G(\alpha y_1, \alpha y_1, \dots, \alpha y_{J_n}) = \alpha G(y_1, y_2, \dots, y_{J_n})$$

3. When any one of y_i grows without bound, G does too

$$\lim_{y_j \to \infty} G(y_1, y_2, \dots, y_{J_n}) = \infty, \text{ for all } y_j$$

4. The cross-partial derivatives of the function G change signs in a specific way, alternating between non-positive and non-negative for higher order derivatives

$$G_j = \tfrac{\partial G}{\partial y_j} \geq 0, \text{ for all } j$$

$$G_{jl} = \tfrac{\partial G}{\partial y_j \partial y_l} \leq 0, \text{ for all } l \neq j$$

$$G_{jlm} = \tfrac{\partial G}{\partial y_j \partial y_l \partial y_m} \geq 0, \text{ for any } m \neq l \neq j$$

Any function that satisfies all these conditions leads to the following choice probability, where G_i corresponds to the first derivative of the function G:

$$P_i = y_i \frac{G_i}{G}$$

Train (2009) notes that the conditions above lack any evident behavioral intuition. This means that any function G that satisfies them is a legitimate discrete choice probability. On the other hand, the lack of evident intuitions means that someone who is trying to develop a model also lacks guidance in terms of how to translate a desired correlation structure into a function G, a limitation that research by Daly and Bierlaire (2003) has aimed to address.

For the above reason, we will not be concerned with the particular meaning of the conditions, but rather will see how the GEV framework has been employed to derive various discrete choice models.

We begin in this section with the following function:

$$G = \sum_{j=1}^{J} y_j$$

with

$$y_j = e^{V_j}$$

This function satisfies the conditions to be a GEV model, since

1. G is always non-negative since y_j is non-negative for every j.
2. The function is homogeneous of degree one, since

$$G(\alpha y_1, \alpha y_1, \ldots, \alpha y_J) = \sum_{j=1}^{J} (\alpha y_j) = \alpha \sum_{j=1}^{J} y_j = \alpha G(y_1, y_1, \ldots, y_J)$$

3. G grows unbounded if any y_j grows unbounded; and
4. The derivatives of this function satisfy the condition since

$$\tfrac{\partial G}{\partial y_j} = 1$$

$$\tfrac{\partial G}{\partial y_j \partial y_l} = 0$$

The first partial derivative is 1, which satisfies the non-negativity condition. All higher order derivatives are zero, which are non-positive and non-negative as required.

Given this function, we can substitute in the choice probability expression

$$P_i = y_i \frac{G_i}{G} = \frac{y_i}{\sum_{j=1}^{J} y_j}$$

And since

$$\frac{\partial G}{\partial y_j} = 1$$

it follows then that

$$P_i = \frac{y_i G_i}{\sum_{j=1}^{J_n} y_j} = \frac{e^{V_i}}{\sum_{j=1}^{J_n} e^{V_j}}$$

which is simply the multinomial logit model.

As this illustrates, the multinomial logit is actually a particular case of a GEV model. Next, we will see how a model with nesting structures is derived using the same approach.

7.6 Nested Logit Model

The nested logit model is derived following the GEV modeling approach by assuming that the J alternatives are uniquely allocated to one class of alternatives, called a nest. There are S nests B_s ($s = 1, 2, 3, \ldots, S$).

Based on the above, we can define the following function:

$$G = \sum_{s=1}^{S} \Big(\sum_{j \in B_s} y_j^{1/\lambda_s} \Big)^{\lambda_s}$$

with

$$y_j = e^{V_j}$$

We can verify that this function satisfies the conditions to be a GEV model

1. G is always non-negative since y_j is non-negative for every j.
2. The function is homogeneous of degree one, since

$$\begin{aligned}
G(\alpha y_1, \alpha y_2, \ldots, \alpha y_J) &= \textstyle\sum_{s=1}^{S} \left(\sum_{j \in B_s} (\alpha y)_j^{1/\lambda_s} \right)^{\lambda_s} \\
&= \textstyle\sum_{s=1}^{S} \left(\sum_{j \in B_s} \alpha_j^{1/\lambda_s} y_j^{1/\lambda_s} \right)^{\lambda_s} \\
&= \textstyle\sum_{s=1}^{S} \left(\sum_{j \in B_s} \alpha_j^{1/\lambda_s} y_j^{1/\lambda_s} \right)^{\lambda_s} \\
&= \alpha \textstyle\sum_{s=1}^{S} \left(\sum_{j \in B_s} y_j^{1/\lambda_s} \right)^{\lambda_s} \\
&= \alpha G(y_1, y_2, \ldots, y_J)
\end{aligned}$$

3. G grows unbounded if any y_j grows unbounded *as long as* $0 \leq \lambda_s \leq 1$ for all s; and
4. The first derivative of this function for j in nest B_t is

$$\begin{aligned}
\tfrac{\partial G}{\partial y_i} &= \lambda_t \left(\textstyle\sum_{j \in B_t} y_j^{1/\lambda_t} \right)^{\lambda_t - 1} \tfrac{1}{\lambda_t} y_i^{1/\lambda_t - 1} \\
&= y_i^{1/\lambda_t - 1} \left(\textstyle\sum_{j \in B_t} y_j^{1/\lambda_t} \right)^{\lambda_t - 1}
\end{aligned}$$

This derivative is always non-negative, since y_j is non-negative for all j.

The second partial derivative is

$$\frac{\partial G}{\partial y_i \partial y_l} = y_i^{1/\lambda_t - 1} \frac{\partial}{\partial y_l} \left(\sum_{j \in B_t} y_j^{1/\lambda_t} \right)^{\lambda_t - 1}$$

When y_l is not part of nest B_t the second derivative is zero, which satisfies the non-positive condition.

When y_l is part of nest B_t the second derivative is

$$\begin{aligned}
\tfrac{\partial G}{\partial y_i \partial y_l} &= y_i^{1/\lambda_t - 1} \tfrac{\partial}{\partial y_l} \left(\textstyle\sum_{j \in B_t} y_j^{1/\lambda_t} \right)^{\lambda_t - 1} \\
&= (\lambda_t - 1) y_i^{1/\lambda_t - 1} \left(\textstyle\sum_{j \in B_t} y_j^{1/\lambda_t} \right)^{\lambda_t - 2} \tfrac{1}{\lambda_t} y_l^{1/\lambda_t - 1} \\
&= \tfrac{\lambda_t - 1}{\lambda_t} (y_i y_l)^{1/\lambda_t - 1} \left(\textstyle\sum_{j \in B_t} y_j^{1/\lambda_t} \right)^{\lambda_t - 2}
\end{aligned}$$

The term $\frac{\lambda_s - 1}{\lambda_s}$ is at most zero as long as $0 \leq \lambda_s \leq 1$; under this condition, the function satisfies the non-positive condition again. Higher order derivatives satisfy the required conditions.

Since G is a valid function for the GEV recipe, we have that

$$
\begin{aligned}
P_i &= y_i \frac{y_i^{1/\lambda_t - 1}\left(\sum_{j\in B_t} y_j^{1/\lambda_t}\right)^{\lambda_t - 1}}{\sum_{s=1}^{S}\left(\sum_{j\in B_s} y_j^{1/\lambda_s}\right)^{\lambda_s}}\\
&= \frac{y_i^{\lambda_t/\lambda_t} y_i^{1/\lambda_t - 1}\left(\sum_{j\in B_t} y_j^{1/\lambda_t}\right)^{\lambda_t - 1}}{\sum_{s=1}^{S}\left(\sum_{j\in B_s} y_j^{1/\lambda_s}\right)^{\lambda_s}}\\
&= \frac{y_i^{1/\lambda_t}\left(\sum_{j\in B_t} y_j^{1/\lambda_t}\right)^{\lambda_t - 1}}{\sum_{s=1}^{S}\left(\sum_{j\in B_s} y_j^{1/\lambda_s}\right)^{\lambda_s}}
\end{aligned}
$$

Replacing $y_j = e^{V_j}$ gives the expression for the choice probability of the nested model

$$
P_i = \frac{e^{V_i/\lambda_t}\left(\sum_{j\in B_t} e^{V_j/\lambda_t}\right)^{\lambda_t - 1}}{\sum_{s=1}^{S}\left(\sum_{j\in B_s} e^{V_j/\lambda_s}\right)^{\lambda_s}}
$$

Superficially, the probability of the nested model resembles the multinomial probability, but the differences are important. One of the examples of nesting in Fig. 7.1 can help to unpack the probability function of the nested model.

Suppose that there are $S = 2$ nests, namely central ("c") and room ("r"). The probability of choosing an alternative in the nest "central", say Electric Central, is

$$
P_{ec} = \frac{e^{V_{ec}/\lambda_c}(e^{V_{ec}/\lambda_c} + e^{V_{gc}/\lambda_c} + e^{V_{hp}/\lambda_c})^{\lambda_c - 1}}{(e^{V_{ec}/\lambda_c} + e^{V_{gc}/\lambda_c} + e^{V_{hp}/\lambda_c})^{\lambda_c} + (e^{V_{er}/\lambda_r} + e^{V_{gr}/\lambda_r})^{\lambda_r}}
$$

The probability of choosing another alternative in the nest "central", say Gas Central, is

$$
P_{gc} = \frac{e^{V_{gc}/\lambda_c}(e^{V_{ec}/\lambda_c} + e^{V_{gc}/\lambda_c} + e^{V_{hp}/\lambda_c})^{\lambda_c - 1}}{(e^{V_{ec}/\lambda_c} + e^{V_{gc}/\lambda_c} + e^{V_{hp}/\lambda_c})^{\lambda_c} + (e^{V_{er}/\lambda_r} + e^{V_{gr}/\lambda_r})^{\lambda_r}}
$$

Notice that the ratio of probabilities is

$$
\begin{aligned}
\frac{P_{ec}}{P_{gc}} &= \frac{\frac{e^{Vec/\lambda c}(e^{Vec/\lambda c} + e^{Vgc/\lambda c} + e^{Vhp/\lambda c})^{\lambda c - 1}}{(e^{Vec/\lambda c} + e^{Vgc/\lambda c} + e^{Vhp/\lambda c})^{\lambda c} + (e^{Ver/\lambda r} + e^{Vgr/\lambda r})^{\lambda r}}}{\frac{e^{Vgc/\lambda c}(e^{Vec/\lambda c} + e^{Vgc/\lambda c} + e^{Vhp/\lambda c})^{\lambda c - 1}}{(e^{Vec/\lambda c} + e^{Vgc/\lambda c} + e^{Vhp/\lambda c})^{\lambda c} + (e^{Ver/\lambda r} + e^{Vgr/\lambda r})^{\lambda r}}}\\
&= \frac{e^{Vec/\lambda c}}{e^{Vgc/\lambda c}}\\
&= \frac{e^{Vec}}{e^{Vgc}}
\end{aligned}
$$

As you can see, the ratio of probabilities is independent from other alternatives: this is the same as the multinomial logit model. In fact, substitution patterns *within* a nest are proportional, as they are in the case of the multinomial logit. This is not true when the ratio of probabilities is for two alternatives in different nests. For example, the probability of choosing an alternative in the nest for room systems, say “er”, is

$$P_{er} = \frac{e^{V_{er}/\lambda_r}(e^{V_{ec}/\lambda_r} + e^{V_{gr}/\lambda_r} + e^{V_{hp}/\lambda_r})^{\lambda_r-1}}{(e^{V_{ec}/\lambda_c} + e^{V_{gc}/\lambda_c} + e^{V_{hp}/\lambda_c})^{\lambda_c-1} + (e^{V_{er}/\lambda_r} + e^{V_{gr}/\lambda_r})^{\lambda_r}}$$

The ratio of probabilities for electric central and electric room is

$$\begin{aligned}\frac{P_{ec}}{P_{er}} &= \frac{\frac{e^{Vec/\lambda c}(e^{Vec/\lambda c}+e^{Vgc/\lambda c}+e^{Vhp/\lambda c})^{\lambda c-1}}{(e^{Vec/\lambda c}+e^{Vgc/\lambda c}+e^{Vhp/\lambda c})^{\lambda c}+(e^{Ver/\lambda r}+e^{Vgc/\lambda r})^{\lambda r}}}{\frac{e^{Ver/\lambda r}(e^{Ver/\lambda r}+e^{Vgr/\lambda r}+e^{Vhp/\lambda r})^{\lambda r-1}}{(e^{Vec/\lambda c}+e^{Vgc/\lambda c}+e^{Vhp/\lambda c})^{\lambda c}+(e^{Ver/\lambda r}+e^{Vgr/\lambda r})^{\lambda r}}} \\ &= \frac{e^{Vec/\lambda c}(e^{Vec/\lambda c}+e^{Vgc/\lambda c}+e^{Vhp/\lambda c})^{\lambda c-1}}{e^{Ver/\lambda r}(e^{Ver/\lambda r}+e^{Vgr/\lambda r}+e^{Vhp/\lambda r})^{\lambda r-1}}\end{aligned}$$

The ratio of probabilities is no longer independent from other alternatives! In fact, the ratio of odds depends on the alternatives in each of the two nests in the model. This property leads to non-proportional substitution patterns.

7.7 Properties of the Nested Logit Model

As seen above, the nested logit model behaves as the multinomial model within a nest (with proportional substitution), but between nests the patterns of substitution are not necessarily proportional. All alternatives within the same nest are assumed to be perceived as perfect substitutes, which is why there is a stronger preference for them relative to alternatives in different nests, which are seen as imperfect substitutes.

An intuitive way to think about the nested logit is in terms of *marginal* and *conditional* probabilities.

Suppose that we decomposed the systematic utility of the alternatives in the following way:

$$V_j = Z_j + W_s$$

where Z_j are attributes specific to the alternatives, whereas W_s are attributes specific to nest s, that is, those that are constant *within* the nest. For example, imagine that central heating systems receive a flat subsidy of \$100. Since the alternatives within the nest all share this attribute, and the model operates based on the differences in utilities, this subsidy is not useful to discriminate between alternatives in the “central” nest. However, the subsidy has potentially the effect of making alternatives in the “central” nest more attractive than alternatives in the “room” nest.

Based on the above decomposition, we can rewrite the choice probabilities as follows:

$$
\begin{aligned}
P_i &= \frac{e^{(Z_i+W_t)/\lambda_t}\left(\sum_{j\in B_t} e^{(Z_j+W_t)/\lambda_t}\right)^{-1}\left(\sum_{j\in B_t} e^{(Z_j+W_t)/\lambda_t}\right)^{\lambda_t}}{\sum_{s=1}^{S}\left(\sum_{j\in B_s} e^{(Z_j+W_s)/\lambda_s}\right)^{\lambda_s}}\\
&= \frac{e^{(Z_i+W_t)/\lambda_t}}{\sum_{j\in B_s} e^{(Z_j+W_t)/\lambda_t}}\frac{\left(\sum_{j\in B_t} e^{(Z_j+W_t)/\lambda_t}\right)^{\lambda_t}}{\sum_{s=1}^{S}\left(\sum_{j\in B_s} e^{(Z_j+W_s)/\lambda_s}\right)^{\lambda_s}}
\end{aligned}
$$

Let us now take each term of the probability above in turn to unpack them. The first term, which we are going to call the *conditional* probability, is as follows:

$$
P_{i|t} = \frac{e^{(Z_i+W_t)/\lambda_t}}{\sum_{j\in B_t} e^{(Z_j+W_t)/\lambda_t}}
$$

We can rearrange this term as follows:

$$
\begin{aligned}
P_{i|t} &= \frac{e^{(Z_i+W_s)/\lambda_t}}{\sum_{j\in B_t} e^{(Z_j+W_t)/\lambda_t}}\\
&= \frac{e^{Z_i/\lambda_t}e^{W_t/\lambda_t}}{\sum_{j\in B_t} e^{Z_j/\lambda_s}e^{W_t/\lambda_t}}\\
&= \frac{e^{Z_i/\lambda_t}e^{W_t/\lambda_t}}{e^{W_t/\lambda_t}\left(\sum_{j\in B_t} e^{Z_j/\lambda_t}\right)}\\
&= \frac{e^{Z_i/\lambda_t}}{\sum_{j\in B_t} e^{Z_j/\lambda_t}}
\end{aligned}
$$

The expression above is the probability of choosing alternative *i* *conditional* on choosing nest t.

Now the second term, which we will call the marginal probability, refers to the probability of choosing the nest *t*

$$
\begin{aligned}
P_t &= \frac{\left(\sum_{j\in B_t} e^{(Z_j+W_t)/\lambda_t}\right)^{\lambda_t}}{\sum_{s=1}^{S}\left(\sum_{j\in B_s} e^{(Z_j+W_s)/\lambda_s}\right)^{\lambda_s}}\\
&= \frac{\left(\sum_{j\in B_t} e^{Z_j/\lambda_t}e^{W_t/\lambda_t}\right)^{\lambda_t}}{\sum_{s=1}^{S}\left(\sum_{j\in B_s} e^{(Z_j+W_s)/\lambda_s}\right)^{\lambda_s}}\\
&= \frac{\left(e^{W_t/\lambda_t}\sum_{j\in B_t} e^{Z_j/\lambda_t}\right)^{\lambda_t}}{\sum_{s=1}^{S}\left(e^{W_s/\lambda_s}\sum_{j\in B_s} e^{Z_j/\lambda_s}\right)^{\lambda_s}}\\
&= \frac{e^{W_t}\left(\sum_{j\in B_t} e^{Z_j/\lambda_t}\right)^{\lambda_t}}{\sum_{s=1}^{S}e^{W_s}\left(\sum_{j\in B_s} e^{Z_j/\lambda_s}\right)^{\lambda_s}}
\end{aligned}
$$

For the last step, we make use of the following property of logarithms:

$$e^x b^c = e^{x+c\ln b}$$

Therefore

$$P_t = \frac{e^{W_t+\lambda_t \ln\left(\sum_{j\in B_t} e^{Z_j/\lambda_t}\right)}}{\sum_{s=1}^{S} e^{W_s+\lambda_s \ln\left(\sum_{j\in B_s} e^{Z_j/\lambda_s}\right)}} = \frac{e^{W_t+\lambda_t I_t}}{\sum_{s=1}^{S} e^{W_s+\lambda_s I_s}}$$

with

$$I_s = \ln\left(\sum_{j\in B_s} e^{Z_j/\lambda_s}\right)$$

I_s is variously called the *logsum*, *expected maximum utility* of a nest, or the *inclusive value*.

It can be appreciated that the probability of choosing alternative i in a nested model is the product of two multinomial logit models in the form of the marginal probability of choosing nest t (sometimes called the upper model), and the probability of choosing i *conditional* on choosing nest t

$$P_i = P_{i|t} \cdot P_t = \frac{e^{(Z_i+W_t)/\lambda_t}}{\sum_{j\in B_t} e^{(Z_j+W_t)/\lambda_t}} \cdot \frac{e^{W_t+\lambda_t I_t}}{\sum_{s=1}^{S} e^{W_s+\lambda_s I_s}}$$

The role of the parameters λ_s is more intuitive in this formulation of the model. As noted above, λ_s is bounded between zero and one. We can examine the limiting cases, as follows:

1. When $\lambda_s < 0$ for any s, increases in the value of I_s would actually decrease the probability of choosing that nest; a result that is inconsistent with utility maximization.
2. When $\lambda \to 0$, changes in the value of I_s have an increasingly small impact on the probability of selecting that nest.
3. When $\lambda_s = 1$ for all s, the choice probability becomes

$$P_i = \frac{e^{V_j}\left(\sum_{j\in B_s} e^{V_j}\right)^0}{\sum_{s=1}^{S}\left(\sum_{j\in B_s} e^{V_j}\right)^1} = \frac{e^{V_j}}{\sum_{k=1}^{J} e^{V_k}}$$

As seen in the latter case, the model collapses to the multinomial logit model, since the denominator simply becomes the sum of e^{V_j} for all alternatives across every nest.

In fact, $1 - \lambda_s$ is a measure of the correlation in the nest, so when $\lambda = 1$ this indicates a correlation of zero.

It is possible to test the hypothesis that a nesting parameter is identical to one, by means of a t-test, as follows:

$$t = \frac{\lambda_s - 1}{\sigma_{\lambda_s}}$$

where σ_{λ_s} is the standard deviation of the parameter.

7.8 Estimation of the Nested Logit Model

The nested logit model can be estimated with package {mlogit} by using additional arguments in the `mlogit()` function to define the nests.

We will illustrate estimation using the same example as above. The chunk of code below defines a formula with installation costs (ic) and operation costs (oc) as alternative-specific variables with a generic coefficient. In addition, the nests are defined by means of a list. In this model, we define two nests that correspond to "room" systems (er and gr) and "central" systems (ec, gc, and hp). Call this model `nl1` for "nested logit 1"

```
nl1 <- mlogit(depvar ~ oc + ic, H,
              nests = list(room = c( 'er', 'gr'),
                           central = c('ec','gc','hp')),
              steptol = 1e-12)
summary(nl1)
```

```
Call:
mlogit(formula = depvar ~ oc + ic, data = H, nests = list(room = c("er",
    "gr"), central = c("ec", "gc", "hp")), steptol = 1e-12)

Frequencies of alternatives:choice
      ec       er       gc       gr       hp
0.071111 0.093333 0.636667 0.143333 0.055556

bfgs method
22 iterations, 0h:0m:0s
g'(-H)^-1g =  -268
last step couldn't find higher value

Coefficients :
                  Estimate  Std. Error  z-value  Pr(>|z|)
(Intercept):er -1.1306e+00  8.1194e-02 -13.9248 < 2.2e-16 ***
(Intercept):gc -5.9927e-03  1.4482e-02  -0.4138  0.679023
(Intercept):gr -1.1754e+00  7.9973e-02 -14.6976 < 2.2e-16 ***
(Intercept):hp -3.7729e-02  1.5564e-02  -2.4240  0.015349 *
oc             -1.8999e-04  6.1450e-05  -3.0919  0.001989 **
ic             -7.1706e-05  2.5598e-05  -2.8012  0.005091 **
iv:room         2.0968e-02  2.3418e-03   8.9538 < 2.2e-16 ***
```

```
iv:central      2.4194e-02  9.0264e-03   2.6804  0.007353 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood: -1003.4
McFadden R^2:  0.01842
Likelihood ratio test : chisq = 37.66 (p.value = 1.3171e-07)
```

It can be seen that the coefficients for installation costs and operation costs are significant and have the expected signs. In addition, the coefficients of the inclusive values are estimated for each nest. Both coefficients appear as significant; however, in this case, the z-value is calculated in reference to zero. As explained in the preceding section, in reality, we should think of those coefficients as correlations $(1 - \lambda_s)$. Accordingly, the correlations for the two nests are

```
1 - nl1$coefficients["iv:room"]
```

```
  iv:room
0.9790318
```

```
1 - nl1$coefficients["iv:central"]
```

```
iv:central
 0.9758055
```

Clearly, the correlations within the nests are very high. We can test whether the correlation is significant by means of the t-test discussed before

```
(nl1$coefficients["iv:room"] - 1) / sqrt(vcov(nl1)["iv:room","iv:room"])
```

```
  iv:room
-418.0641
```

```
(nl1$coefficients["iv:central"] - 1) / sqrt(vcov(nl1)["iv:central","iv:central"])
```

```
iv:central
 -108.1057
```

The cutoff value of the t-test at the 5% level of confidence is ±1.96. The high (negative) values of the t-test indicate that the null hypothesis, i.e., $\lambda_s = 1$ can be rejected at a high level of confidence.

In addition, since the multinomial logit model is a special case of the nested logit model (when $\lambda_s = 1$ for all s), it is also possible to conduct a likelihood ratio test, as follows:

```
lrtest(model3, nl1)
```

```
Likelihood ratio test

Model 1: depvar ~ ic + oc
Model 2: depvar ~ oc + ic
  #Df  LogLik Df  Chisq Pr(>Chisq)
1   6 -1008.2
2   8 -1003.4  2 9.6697   0.007948 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The likelihood ratio test also indicates that the null hypothesis, i.e., that the nested logit collapses to the multinomial logit, can be rejected at a high level of confidence $(p - val < 0.01)$.

It is possible to impose some restrictions to the estimation of the parameters for the nests. The `mlogit()` function allows an additional argument that forces all parameters $lambda_s$ to take identical values. Given the similarity of the two parameters, it is sensible to use this argument (`un.nest.el`, for "unique nest elasticity"), and set it to `TRUE` to estimate only one parameter (call this Nested Logit Model 2, `nl2`)

```
nl2 <- mlogit(depvar ~ ic + oc, H,
              nests = list(room = c( 'er', 'gr'), central = c('ec', 'gc', 'hp')),
              un.nest.el = TRUE,
              steptol = 1e-12)
summary(nl2)
```

```
Call:
mlogit(formula = depvar ~ ic + oc, data = H, nests = list(room = c("er",
    "gr"), central = c("ec", "gc", "hp")), un.nest.el = TRUE,
    steptol = 1e-12)

Frequencies of alternatives:choice
      ec       er       gc       gr       hp
0.071111 0.093333 0.636667 0.143333 0.055556

bfgs method
19 iterations, 0h:0m:0s
g'(-H)^-1g = 0.0443
successive function values within tolerance limits

Coefficients :
                  Estimate  Std. Error  z-value  Pr(>|z|)
(Intercept):er -1.1290e+00  7.8672e-02 -14.3508 < 2.2e-16 ***
(Intercept):gc -1.0474e-02  1.5358e-02  -0.6820 0.4952521
(Intercept):gr -1.1817e+00  8.0048e-02 -14.7628 < 2.2e-16 ***
(Intercept):hp -4.5459e-02  1.5516e-02  -2.9298 0.0033921 **
ic             -8.3899e-05  2.5826e-05  -3.2486 0.0011598 **
oc             -2.2841e-04  5.9509e-05  -3.8383 0.0001239 ***
iv              2.7374e-02  3.2338e-03   8.4649 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood: -1003.5
McFadden R^2:  0.018329
Likelihood ratio test : chisq = 37.473 (p.value = 3.6549e-08)
```

Notice how the log likelihood of the Nested Logit Model 2 is almost identical to the log likelihood of the Nested Logit Model 1. Since `nl2` is more parsimonious, it makes sense to prefer this model. Also, the likelihood ratio test fails to reject the null hypothesis of a statistically significant difference between the two models (p-value = 0.66).

```
lrtest(nl2, nl1)
```

```
Likelihood ratio test

Model 1: depvar ~ ic + oc
Model 2: depvar ~ oc + ic
  #Df  LogLik Df Chisq Pr(>Chisq)
1   7 -1003.5
2   8 -1003.4  1 0.187     0.6654
```

7.9 Substitution Patterns with the Nested Logit Model

In this section, we will revisit the question of substitution patterns, but now using the nested logit model. To simulate the substitution patterns, we will remove each alternative in turn as we did before. This allows us to examine how the adoption rates change in response.

We begin by copying the model matrix

```
X <- model.matrix(nl2)
head(X,12)
```

```
   (Intercept):er (Intercept):gc (Intercept):gr (Intercept):hp      ic     oc
1               0              0              0              0  859.90 553.34
2               1              0              0              0  995.76 505.60
3               0              1              0              0  866.00 199.69
4               0              0              1              0  962.64 151.72
5               0              0              0              1 1135.50 237.88
6               0              0              0              0  796.82 520.24
7               1              0              0              0  894.69 486.49
8               0              1              0              0  727.93 168.66
9               0              0              1              0  758.89 168.66
10              0              0              0              1  968.90 199.19
11              0              0              0              0  719.86 439.06
12              1              0              0              0  900.11 404.74
```

Using this model matrix, we can calculate the utilities of each alternative. Notice that each utility is divided by the coefficient of the inclusive value. Notice that in this example there are no variables that are constants inside the nests, so $W_s = 0$ for all nests (i.e., there are no nest-specific variables).

```
# Electric central
exp_V_ec <-
  exp((X[alt == c("ec"), "oc"] * coef(nl2)["oc"] +
         X[alt == c("ec"), "ic"] * coef(nl2)["ic"])
                / coef(nl2)["iv"])

# Gas central
exp_V_gc <-
  exp((coef(nl2)["(Intercept):gc"] +
         X[alt == c("gc"), "oc"] * coef(nl2)["oc"] +
         X[alt == c("gc"), "ic"] * coef(nl2)["ic"])
                / coef(nl2)["iv"])

# Heat pump
exp_V_hp <-
  exp((coef(nl2)["(Intercept):hp"] +
         X[alt == c("hp"), "oc"] * coef(nl2)["oc"] +
         X[alt == c("hp"), "ic"] * coef(nl2)["ic"])
                / coef(nl2)["iv"])

# Electric room
exp_V_er <-
  exp((coef(nl2)["(Intercept):er"] +
         X[alt == c("er"), "oc"] * coef(nl2)["oc"] +
         X[alt == c("er"), "ic"] * coef(nl2)["ic"])
                / coef(nl2)["iv"])

# Gas room
exp_V_gr <-
  exp((coef(nl2)["(Intercept):gr"] +
         X[alt == c("gr"), "oc"] * coef(nl2)["oc"] +
         X[alt == c("gr"), "ic"] * coef(nl2)["ic"])
                / coef(nl2)["iv"])
```

The conditional probabilities are the logit models within each nest

```
# Conditional probabilities of systems within the central nest
# after removing ec
cp_mec_c <- data.frame(gc = exp_V_gc / (exp_V_gc + exp_V_hp),
                       hp = exp_V_hp / (exp_V_gc + exp_V_hp))
# Conditional probabilities of systems within the central nest
# after removing gc
cp_mgc_c <- data.frame(ec = exp_V_ec / (exp_V_ec + exp_V_hp),
                       hp = exp_V_hp / (exp_V_ec + exp_V_hp))
# Conditional probabilities of systems within the central nest
# after removing hp
cp_mhp_c <- data.frame(ec = exp_V_ec / (exp_V_ec + exp_V_gc),
                       gc = exp_V_gc / (exp_V_ec + exp_V_gc))

# Conditional probabilities of systems within the room nest
# after removing a system in the central nest
```

```
cp_mc_r <- data.frame(er = exp_V_er / (exp_V_er + exp_V_gr),
                       gr = exp_V_gr / (exp_V_er + exp_V_gr))

# Conditional probabilities of systems within the room nest
# after removing er
cp_mer_r <- data.frame(gr = exp_V_gr / (exp_V_gr))
# Conditional probabilities of systems within the room nest
# after removing gr
cp_mgr_r <- data.frame(er = exp_V_er / (exp_V_er))

# Conditional probabilities of systems within the central nest
# after removing a system in the room nest
cp_mr_c <- data.frame(ec = exp_V_ec / (exp_V_ec + exp_V_gc + exp_V_hp),
                      gc = exp_V_gc / (exp_V_ec + exp_V_gc + exp_V_hp),
                      hp = exp_V_hp / (exp_V_ec + exp_V_gc + exp_V_hp))
```

The marginal probabilities are the logit probabilities of choosing a nest, given the expected maximum utility of each nest

```
#After removing ec
mp_mec <- data.frame(central = exp(coef(nl2)["iv"] * log(exp_V_gc + exp_V_hp))
                       / (exp(coef(nl2)["iv"] * log(exp_V_gc + exp_V_hp)) +
                            exp((coef(nl2)["iv"] * log(exp_V_er + exp_V_gr)))),
                       room = exp(coef(nl2)["iv"] * log(exp_V_er + exp_V_gr)) /
                         (exp(coef(nl2)["iv"] * log(exp_V_gc + exp_V_hp)) +
                            exp((coef(nl2)["iv"] * log(exp_V_er + exp_V_gr))))
                       )
#After removing gc
mp_mgc <- data.frame(central = exp(coef(nl2)["iv"] * log(exp_V_ec + exp_V_hp))
                       / (exp(coef(nl2)["iv"] * log(exp_V_ec + exp_V_hp)) +
                            exp((coef(nl2)["iv"] * log(exp_V_er + exp_V_gr)))),
                       room = exp(coef(nl2)["iv"] * log(exp_V_er + exp_V_gr)) /
                         (exp(coef(nl2)["iv"] * log(exp_V_ec + exp_V_hp)) +
                            exp((coef(nl2)["iv"] * log(exp_V_er + exp_V_gr))))
                       )

#After removing hp
mp_mhp <- data.frame(central = exp(coef(nl2)["iv"] * log(exp_V_ec + exp_V_gc))
                       / (exp(coef(nl2)["iv"] * log(exp_V_ec + exp_V_gc)) +
                            exp((coef(nl2)["iv"] * log(exp_V_er + exp_V_gr)))),
                       room = exp(coef(nl2)["iv"] * log(exp_V_er + exp_V_gr)) /
                         (exp(coef(nl2)["iv"] * log(exp_V_ec + exp_V_gc)) +
                            exp((coef(nl2)["iv"] * log(exp_V_er + exp_V_gr))))
                       )

#After removing er
mp_mer <- data.frame(central = exp(coef(nl2)["iv"] * log(exp_V_ec + exp_V_gc + exp_V_hp))
                       / (exp(coef(nl2)["iv"] * log(exp_V_ec + exp_V_gc + exp_V_hp)) +
                            exp((coef(nl2)["iv"] * log(exp_V_gr)))),
                       room = exp(coef(nl2)["iv"] * log(exp_V_gr)) /
                         (exp(coef(nl2)["iv"] * log(exp_V_ec + exp_V_gc + exp_V_hp)) +
                            exp((coef(nl2)["iv"] * log(exp_V_gr))))
                       )

#After removing gr
mp_mgr <- data.frame(central = exp(coef(nl2)["iv"] * log(exp_V_ec + exp_V_gc + exp_V_hp))
                       / (exp(coef(nl2)["iv"] * log(exp_V_ec + exp_V_gc + exp_V_hp)) +
                            exp((coef(nl2)["iv"] * log(exp_V_er)))),
                       room = exp(coef(nl2)["iv"] * log(exp_V_er)) /
                         (exp(coef(nl2)["iv"] * log(exp_V_ec + exp_V_gc + exp_V_hp)) +
                            exp((coef(nl2)["iv"] * log(exp_V_er))))
                       )
```

Once that the conditional and marginal choice probabilities for each case have been calculated, the choice probabilities are the product of the conditional and marginal probabilities

```
#After removing ec
nlp_mec <- data.frame(cp_mec_c, cp_mc_r, mp_mec) %>%
  transmute(p_ec = NA,
            p_gc = gc * central,
            p_hp = hp * central,
            p_er = er * room,
            p_gr = gr * room)

#After removing gc
nlp_mgc <- data.frame(cp_mgc_c, cp_mc_r, mp_mgc) %>%
  transmute(p_ec = ec * central,
            p_gc = NA,
            p_hp = hp * central,
            p_er = er * room,
            p_gr = gr * room)

#After removing hp
nlp_mhp <- data.frame(cp_mhp_c, cp_mc_r, mp_mhp) %>%
  transmute(p_ec = ec * central,
            p_gc = gc * central,
            p_hp = NA,
            p_er = er * room,
            p_gr = gr * room)

#After removing er
nlp_mer <- data.frame(cp_mr_c, cp_mer_r, mp_mer) %>%
  transmute(p_ec = ec * central,
            p_gc = gc * central,
            p_hp = hp * central,
            p_er = NA,
            p_gr = gr * room)

#After removing gr
nlp_mgr <- data.frame(cp_mr_c, cp_mgr_r, mp_mgr) %>%
  transmute(p_ec = ec * central,
            p_gc = gc * central,
            p_hp = hp * central,
            p_er = er * room,
            p_gr = NA)
```

A quick sanity check can be conducted to ensure that the sum of probabilities for each decision-maker is one

```
summary(rowSums(nlp_mec, na.rm = TRUE))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      1       1       1       1       1       1 
```

```
summary(rowSums(nlp_mgc, na.rm = TRUE))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      1       1       1       1       1       1 
```

```
summary(rowSums(nlp_mhp, na.rm = TRUE))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      1       1       1       1       1       1 
```

```
summary(rowSums(nlp_mer, na.rm = TRUE))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      1       1       1       1       1       1 
```

```
summary(rowSums(nlp_mgr, na.rm = TRUE))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      1       1       1       1       1       1 
```

Given the above, we can summarize the choice probabilities in the form of adoption rates. The table below shows the original adoption rates (when no alternative is removed) and for the different situations of interest, after removing each alternative

```
# Original adoption rates
# Using the fitted function to calculate the probabilities for each household
p_o <- apply(fitted(nl2, outcome = FALSE), 2, mean)

df <- data.frame(Alternative = c("None", "ec", "gc", "hp", "er", "gr" ),
  rbind(c(p_o["ec"], p_o["gc"], p_o["hp"], p_o["er"], p_o["gr"]),
        apply(nlp_mec, 2, mean),
        apply(nlp_mgc, 2, mean),
        apply(nlp_mhp, 2, mean),
        apply(nlp_mer, 2, mean),
        apply(nlp_mgr, 2, mean))
)

df %>%
  kable(col.names = c("Alternative Removed",
                      "ec",
                      "gc",
                      "hp",
                      "er",
                      "gr"),
        digits = 3) %>%
  kable_styling()
```

Alternative Removed	ec	gc	hp	er	gr
None	0.071	0.637	0.055	0.088	0.148
ec	NA	0.701	0.061	0.088	0.149
gc	0.411	NA	0.343	0.092	0.154
hp	0.077	0.686	NA	0.088	0.149
er	0.072	0.639	0.056	NA	0.234
gr	0.072	0.641	0.056	0.232	NA

Notice that the patterns of substitution are no longer proportional between nests. Are these substitution patterns more sensible?

7.10 Paired Combinatorial Logit

Before a nested logit model can be estimated, an analyst must posit a plausible choice structure (see the two nested structures in Fig. 7.1). It is not always evident what the choice structure should be, and the analyst may need to experiment with several structures. Another model of the GEV family, first developed by Chu (1989) and reintroduced into the literature by Koppelman and Wen (2000), takes a different approach to pre-specified choice structures: instead of defining the nests a priori, the model creates nests paired nests of each alternative with every other, as shown in Fig. 7.2. Appropriately, this model is called the paired combinatorial logit.

The paired combinatorial logit results from using the following function in the GEV recipe:

$$G = \sum_{i=1}^{J-1}\left(\sum_{j=i+1}^{J} y_i^{1/\lambda_{ij}} + y_j^{1/\lambda_{ij}}\right)^{\lambda_{ij}}$$

The resulting probabilities can be expressed in a similar fashion to the probabilities of the nested logit model, in terms of marginal and conditional probabilities.

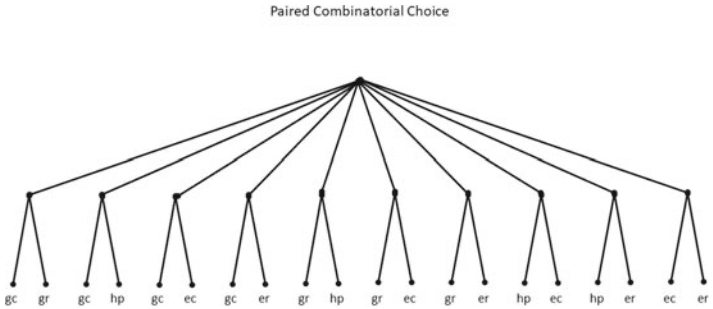


Fig. 7.2 Paired combinatorial logit choice structure

Accordingly, P_i (the probability of choosing alternative i) is the sum of the joint probabilities of choosing i in the various paired combinatorial nests

$$P_i = \sum_{j \neq i} P_{i|ij} P_{ij}$$

$P_{i|ij}$ in the equation above is the conditional probability of choosing i given that it is in the nest that contains j, and P_{ij} is the marginal probability of choosing the nest that contains alternatives i and j. These probabilities are given by

$$\begin{aligned} P_{i|ij} &= \frac{e^{V_i/\lambda_{ij}}}{e^{V_i/\lambda_{ij}} + e^{V_j/\lambda_{ij}}} \\ P_{ij} &= \frac{(e^{V_i/\lambda_{ij}} + e^{V_j/\lambda_{ij}})^{\lambda_{ij}}}{\sum_{k=1}^{J-1} \sum_{m=k+1}^{J} (e^{V_k/\lambda_{km}} + e^{V_m/\lambda_{km}})^{\lambda_{km}}} \end{aligned}$$

A potential issue with the paired combinatorial logit is the proliferation of nests, with their corresponding parameters. The number of nests equals

$$\frac{J!}{r!(J-r)!}$$

Since the model posits pairs of alternatives in each nest, this is

$$\frac{J!}{2(J-2)!}$$

As seen in Fig. 7.2, when $J = 5$ this means ten nests. With $J = 6$ this grows to 15, with $J = 7$ this grows to 21, and so on. Effectively, the number of nests grows by $J - 1$ for each additional alternative in the choice set.

The paired combinatorial logit model is implemented in package {mlogit} and can be estimated by selecting "pcl" for the nests: this will automatically create all paired combinations. The next chunk of code presents the results of estimating this model

```
pcl <- mlogit(depvar ~ ic + oc, H,
              nests = "pcl",
              steptol = 1e-12)
summary(pcl)
```

```
Call:
mlogit(formula = depvar ~ ic + oc, data = H, nests = "pcl", steptol = 1e-12)

Frequencies of alternatives:choice
      ec       er       gc       gr       hp
0.071111 0.093333 0.636667 0.143333 0.055556

bfgs method
9 iterations, 0h:0m:1s
g'(-H)^-1g = 2.42E+04
last step couldn't find higher value
```

```
Coefficients :
                  Estimate Std. Error z-value Pr(>|z|)
(Intercept):er   0.9117511  3.5428729  0.2573 0.796910
(Intercept):gc   0.0598752  2.8296714  0.0212 0.983118
(Intercept):gr  -2.1026308  2.3623182 -0.8901 0.373428
(Intercept):hp  -1.8065353  6.2500251 -0.2890 0.772547
ic              -0.0021546  0.0023784 -0.9059 0.364982
oc              -0.0084739  0.0090062 -0.9409 0.346760
iv:ec.er         1.7900964  3.7110198  0.4824 0.629541
iv:ec.gc         0.8796763  8.6470812  0.1017 0.918970
iv:ec.gr         1.5433716  5.6604277  0.2727 0.785115
iv:ec.hp         2.9988296  3.8633513  0.7762 0.437616
iv:er.gc         0.8441107  4.6616241  0.1811 0.856308
iv:er.gr         0.0309148  0.0094296  3.2785 0.001044 **
iv:er.hp         1.1092344  3.6451052  0.3043 0.760893
iv:gc.gr         2.9247211  2.8344957  1.0318 0.302151
iv:gc.hp         0.4692067 16.0563954  0.0292 0.976687
iv:gr.hp         2.1680975  5.4665445  0.3966 0.691654
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Log Likelihood: -1004.9
McFadden R^2:  0.01692
Likelihood ratio test : chisq = 34.591 (p.value = 0.00054385)
```

We see that the paired combinatorial logit model is not very appealing for this particular example: none of the cost variables appears to be significant; furthermore, several of the nest parameters λ give values that are not consistent with utility maximization (recall that this parameter should be between zero and one).

The paired combinatorial logit model cannot be compared to nested logit models directly (in general, the nests of the paired combinatorial logit are incompatible with those of the nested logit model). That said, we can point at McFadden's ρ^2 index, which is lower for the paired combinatorial logit: this model uses more parameters to give a worse log likelihood value.

It is possible to impose restrictions to the model to reduce the number of nests, as shown next, by means of argument `constPar`, which can be a vector of names of the parameters to restrict. In the above example, $\lambda_{ec.gc}$ may not be significantly different from one. The z-score to compare this coefficient to one is as follows:

```
as.numeric((1 - pcl$coefficients["iv:ec.gc"])/8.6470812)
```

```
[1] 0.01391495
```

With a p-value of 0.5055511, we fail to reject the null hypothesis that $\lambda_{ec.gc}$ is different from one, so we can restrict this parameter as shown next

```r
pcl2 <- mlogit(depvar ~ ic + oc, H,
               nests = "pcl",
               constPar=c("iv:ec.gc"),
               steptol = 1e-12)
summary(pcl2)
```

```
Call:
mlogit(formula = depvar ~ ic + oc, data = H, nests = "pcl", constPar = c("iv:ec.gc"),
    steptol = 1e-12)

Frequencies of alternatives:choice
      ec       er       gc       gr       hp
0.071111 0.093333 0.636667 0.143333 0.055556

bfgs method
7 iterations, 0h:0m:1s
g'(-H)^-1g =  5.52
last step couldn't find higher value

Coefficients :
                 Estimate Std. Error z-value  Pr(>|z|)
(Intercept):er  1.2367842  1.3852165  0.8928 0.3719400
(Intercept):gc  0.3082732  1.2823258  0.2404 0.8100189
(Intercept):gr -1.7825253  1.5976572 -1.1157 0.2645454
(Intercept):hp -1.4831195  3.9207360 -0.3783 0.7052257
ic             -0.0020715  0.0010490 -1.9748 0.0482950 *
oc             -0.0076288  0.0030696 -2.4853 0.0129449 *
iv:ec.er        1.2609877  2.4210587  0.5208 0.6024772
iv:ec.gr        1.9345644  3.2497080  0.5953 0.5516402
iv:ec.hp        2.6672495  3.4672908  0.7693 0.4417387
iv:er.gc        0.2429807  1.1655145  0.2085 0.8348581
iv:er.gr        0.0529682  0.0139888  3.7865 0.0001528 ***
iv:er.hp        1.0999662  2.9507150  0.3728 0.7093125
iv:gc.gr        2.6319952  1.4206127  1.8527 0.0639227 .
iv:gc.hp        0.4802512  9.5416271  0.0503 0.9598577
iv:gr.hp        1.9378174  5.2934473  0.3661 0.7143065
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log Likelihood: -1005.6
McFadden R^2:  0.016265
Likelihood ratio test : chisq = 33.254 (p.value = 0.00047857)
```

Now the cost variables are significantly different from zero, but some nest parameters are still outside of the (0, 1] range and/or may not be different from 1.

7.11 Elasticities of the Nested and Paired Combinatorial Logit Models

Recall that the elasticity is the change in probability that results from a one percent change in some attribute.

In the case of the nested logit, it is also possible to obtain expressions for the direct- and cross-point elasticity, however, these need to take into account the fact that substitution patterns are not proportional. In particular, the coefficient of the inclusive value affects the probability of choosing a nest and therefore any alternatives contained there.

Accordingly, the direct-point elasticity in the case of a nested logit model is (see Louviere et al. 2000, pp. 148–49, but notice the difference with the nest parameters in that reference)

$$E^{P_{in}}_{x_{ink}} = \left[(1 - P_t) + \left(\frac{1 - \lambda_t}{\lambda_t}\right)(1 - P_{i|t})\right]\beta_{ik}x_{ink}$$

where P_t is the marginal probability of choosing nest t and $P_{i|t}$ is the probability of choosing i conditional on choosing nest t. Note that when an alternative is not nested, the conditional probability is one, and therefore, the elasticity is identical to the elasticity of the multinomial logit model.

The cross-elasticity for alternatives in a partition of the nest is

$$E^{P_{jn}}_{x_{ink}} = -\left[P_t + \left(\frac{1 - \lambda_t}{\lambda_t}\right)P_{i|t}\right]\beta_{ik}x_{ink}$$

The elasticities for the nested logit model 2 are calculated next. First, copy the first five elements of the model matrix (one row per alternative), and copy the indices to `alt`

```
X_mean <- model.matrix(nl2)[1:5,]
alt <- index(H)$alt[1:5]
```

Next, calculate the mean installation cost and operation cost for each system

```
mean_ic <- H %>%
  group_by(alt) %>%
  summarize(ic = mean(ic)) %>%
  arrange(alt)

mean_oc <- H %>%
  group_by(alt) %>%
  summarize(oc = mean(oc)) %>%
  arrange(alt)
```

Replace the mean installation costs in the model matrix for the effects:

```
X_mean[,5] <- mean_ic$ic
X_mean[,6] <- mean_oc$oc
```

Calculate the exponential of the utility functions with operation costs and installation costs set to the mean values of each alternative; notice that we use the coefficient of the inclusive value

```
# Electric central
exp_V_ec <-
  exp((X_mean[alt == c("ec"), "oc"] * coef(nl2)["oc"] +
         X_mean[alt == c("ec"), "ic"] * coef(nl2)["ic"])
                / coef(nl2)["iv"])
# Electric room
exp_V_er <-
  exp((coef(nl2)["(Intercept):er"] +
         X_mean[alt == c("er"), "oc"] * coef(nl2)["oc"] +
         X_mean[alt == c("er"), "ic"] * coef(nl2)["ic"])
                / coef(nl2)["iv"])
# Gas central
exp_V_gc <-
  exp((coef(nl2)["(Intercept):gc"] +
         X_mean[alt == c("gc"), "oc"] * coef(nl2)["oc"] +
         X_mean[alt == c("gc"), "ic"] * coef(nl2)["ic"])
                / coef(nl2)["iv"])

# Gas room
exp_V_gr <-
  exp((coef(nl2)["(Intercept):gr"] +
         X_mean[alt == c("gr"), "oc"] * coef(nl2)["oc"] +
         X_mean[alt == c("gr"), "ic"] * coef(nl2)["ic"])
                / coef(nl2)["iv"])

# Heat pump
exp_V_hp <-
  exp((coef(nl2)["(Intercept):hp"] +
         X_mean[alt == c("hp"), "oc"] * coef(nl2)["oc"] +
         X_mean[alt == c("hp"), "ic"] * coef(nl2)["ic"])
                / coef(nl2)["iv"])
```

The conditional probabilities are the logit models within each nest

```
# Conditional probabilities of systems within the central nest
cp_c <- data.frame(ec = exp_V_ec / (exp_V_ec + exp_V_gc + exp_V_hp),
                       gc = exp_V_gc / (exp_V_ec + exp_V_gc + exp_V_hp),
                       hp = exp_V_hp / (exp_V_ec + exp_V_gc + exp_V_hp))

# Conditional probabilities of systems within the room nest
cp_r <- data.frame(er = exp_V_er / (exp_V_er + exp_V_gr),
                       gr = exp_V_gr / (exp_V_er + exp_V_gr))
```

The marginal probabilities are the logit probabilities of choosing a nest, given the expected maximum utility of each nest

```
#After removing ec
mp <- data.frame(central = exp(coef(nl2)["iv"] * log(exp_V_ec + exp_V_gc + exp_V_hp))
                       / (exp(coef(nl2)["iv"] * log(exp_V_ec + exp_V_gc + exp_V_hp)) +
                            exp((coef(nl2)["iv"] * log(exp_V_er + exp_V_gr)))),
                       room = exp(coef(nl2)["iv"] * log(exp_V_er + exp_V_gr)) /
                         (exp(coef(nl2)["iv"] * log(exp_V_gc + exp_V_hp)) +
                            exp((coef(nl2)["iv"] * log(exp_V_er + exp_V_gr)))))
```

We will collect all terms needed to calculate the joint probabilities and the elasticities in data frame `nlp`

```
nlp <- data.frame(system = c("ec", "er", "gc", "gr", "hp"),
                  # Conditional probability
                  cp = c(cp_c$ec, cp_r$er, cp_c$gc, cp_r$gr, cp_c$hp),
                  # Marginal probability
                  mp = c(mp$central, mp$room, mp$central, mp$room, mp$central),
                  beta_ic = c(as.numeric(nl2$coefficients["ic"])),
                  beta_oc = c(as.numeric(nl2$coefficients["oc"])),
                  lambda = c(as.numeric(nl2$coefficients["iv"]))) %>%
  # Joint probability
  mutate(p = cp * mp)
```

To calculate the elasticities, increase the cost of installation by 1%

```
nlp <- cbind(nlp, X_mean[,5:6]) %>%
  # Increase installation cost 1%
  mutate(ic_1pct = 1.01 * ic)
```

Calculate the direct elasticities of a 1% change in the mean installation cost of each alternative

```
direct_elasticities <- nlp %>%
  transmute(DEM = ((1 - mp)    +    (1 - cp) * (1 - lambda)/lambda)    *    beta_ic * ic)

direct_elasticities
```

```
         DEM
1 -2.2670814
2 -1.9302931
3 -0.3718028
4 -1.0574735
5 -2.9232971
```

Calculate the cross-elasticities given a 1% change in the mean cost of each of the alternatives

```
elasticities <- nlp %>%
  transmute(CEM_ec = -(mp    +  (1 - lambda)/lambda * cp)    *    beta_ic * mean_ic$ic[mean_ic == "ec"],
            CEM_er = -(mp    +  (1 - lambda)/lambda * cp)    *    beta_ic * mean_ic$ic[mean_ic == "er"],
            CEM_gc = -(mp    +  (1 - lambda)/lambda * cp)    *    beta_ic * mean_ic$ic[mean_ic == "gc"],
            CEM_gr = -(mp    +  (1 - lambda)/lambda * cp)    *    beta_ic * mean_ic$ic[mean_ic == "gr"],
            CEM_hp = -(mp    +  (1 - lambda)/lambda * cp)    *    beta_ic * mean_ic$ic[mean_ic == "hp"]) %>%
  # Transmute so that each row is the elasticity due to changes to a system
  t()
```

Replace the diagonal with the direct elasticities

```
diag(elasticities) <- direct_elasticities$DEM
```

The matrix of elasticities is

```
elasticities
```

```
                 1          2          3         4          5
CEM_ec -2.2670814  0.9095451  2.1325130  1.581221  0.2238297
CEM_er  0.3103446 -1.9302931  2.5447285  1.886871  0.2670961
CEM_gc  0.2450219  0.8569092 -0.3718028  1.489715  0.2108765
CEM_gr  0.2907392  1.0167950  2.3839701 -1.057473  0.2502228
CEM_hp  0.3300748  1.1543623  2.7065096  2.006829 -2.9232971
```

Compare the elasticies of the nested logit model to the elasticies of multinomial logit model Model 3, we first calculate the mean of the cost variables

```
# Copy a single row of the input data frame
ic_oc_mean <- Heating[1,]

# Calculate the mean of the cost variables for each system
mean_ic_oc <- Heating %>%
  select(starts_with("ic") | starts_with("oc"))%>%
  summarise(across(.cols = everything(),
                   mean))

# Replace the cost of installation and operation with the mean values:
ic_oc_mean[3:12] <- mean_ic_oc
```

We can now use the data with the costs set to the mean to compute the elasticities according to Model 3:

```
effects(model3,
        covariate = "oc",
        type = "rr",
        data = mlogit.data(ic_oc_mean,
                           shape = "wide",
                           choice = "depvar",
                           varying = 3:12))
```

```
            ec          er          gc          gr         hp
ec -3.11089165  0.22499750  0.22499750  0.22499750  0.2249975
er  0.26819485 -2.73835328  0.26819486  0.26819486  0.2681949
gc  0.77599637  0.77599638 -0.42818936  0.77599637  0.7759964
gr  0.15516657  0.15516657  0.15516657 -0.92557230  0.1551666
hp  0.08493646  0.08493646  0.08493646  0.08493646 -1.4493621
```

7.12 Exercises

The blue bus-red bus paradox is a classical illustration of the limitations of the multinomial logit model. This paradox is stated next.

The blue bus-red bus paradox

There are two initial modes, car and blue buses, with systematic utility functions as follows:

$$V_{blue} = V_{car}$$

According to the multinomial logit model, the probability of choosing either mode is 0.5, since

$$P_{car} = \frac{e^V_{car}}{e^V_{car} + e^V_{blue}} = \frac{e^V_{car}}{e^V_{car} + e^V_{car}} = \frac{1}{2}$$

and

$$P_{blue} = 1 - P_{car} = \frac{1}{2}$$

A new alternative is introduced. In fact, the new alternative is just some old blue buses painted red. Since consumers do not care about the color of buses, the utility of this new alternative is

$$V_{yellow} = V_{bus} = V_{car}$$

The new choice probabilities are now

$$\begin{aligned} P_{car} &= \tfrac{e^V car}{e^V car + e^V blue + e^V yellow} = \tfrac{1}{3} \\ P_{blue} &= \tfrac{e^V car}{e^V car + e^V blue + e^V yellow} = \tfrac{1}{3} \\ P_{yellow} &= 1 - P_{car} - P_{blue} = \tfrac{1}{3} \end{aligned}$$

Proportional substitution patterns imply that the new mode (red bus) draws equally from the alternatives, i.e., car and blue buses. Clearly, this does not make sense. An entrepreneur could paint buses in many different colors and reduce the probability of choosing car to zero as a consequence.

1. Restate the blue bus-red bus situation as a nested logit model. What are the marginal and conditional probabilities of this model?
2. Use model `nl2` in this chapter and calculate the direct-point elasticity at the mean values of the variables, for an increase in the installation costs of Gas Central systems.
3. Use model `nl2` in this chapter and calculate the cross-point elasticity at the mean values of the variables, for a 1% increase in the operation costs of Gas Central systems.

4. Re-estimate the nested logit model in this chapter, but change the nests to types of energy as follows:

- Gas: gas central, gas room.
- Electricity: electric central, electric room, heat pump.

Use a single coefficient for the inclusive variables (i.e., set `un.nest.el = TRUE`). Are the results reasonable? Discuss.

References

Ben-Akiva, M., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and applications to travel demand*. Cambridge: The MIT Press.

Chu, C. (1989). A paired combinatorial logit model for travel demand analysis. In *Proceedings of the Fifth World Conference on Transportation Research* (Vol. 1989(4), pp. 295–309).

Daly, Andrew, & Bierlaire, Michel. (2003). *A general and operational representation of GEV models*. EPFL, Lausanne, May: Department of Mathematics.

Koppelman, F. S., & Wen, C. H. (2000). The paired combinatorial logit model: Properties, estimation and application. *Transportation Research Part B-Methodological,34*(2), 75–89. https://doi.org/10.1016/S0191-2615(99)00012-0.

Luce, R Duncan. (1959). *Individual choice behavior: A theoretical analysis*. New York: Wiley.

McFadden, D. (1978). Modelling the choice of residential location. In A. Karlqvist, L. Lundqvist, F. Snickars, & J. W. Weibull (Eds.), *Spatial interaction theory and planning models* (pp. 75–96). Amsterdam: North-Holland.

Train, K. (2009). *Discrete choice methods with simulation* (2nd ed.). Cambridge: Cambridge University Press.

Chapter 8
Non-proportional Substitution Patterns II: The Probit Model

Perfectionism means that you try not to leave so much mess to clean up.

— Anne Lamott, Bird by Bird: Some Instructions on Writing and Life

Perfection's unattainable but it isn't unapproachable.
— Peter Watts, Blindsight

Perfection exacts a price, but it's the imperfect who pay it.

— Margaret Atwood, MadAddam

8.1 More on Flexible Substitution Patterns

In Chap. 7, the topic of non-proportional substitution was discussed, and a method for deriving logit models using the Generalized Extreme Value (GEV) system was presented. In particular, the nested (or hierarchical) logit model was introduced as an alternative modelling approach to alleviate the issues that emerge when a multinomial logit model is not fully specified. If there are hidden correlations, the proportional substitution patterns that result from the Independence from Irrelevant Alternatives property may be inappropriate.

The nested logit model approximates a hierarchical decision process where decision-makers first choose a class of alternatives (a nest) and then they choose an alternative within the nest. Within nests, the model behaves like the multinomial logit model and leads to proportional substitution. But between nests, the patterns of substitution are no longer proportional. In practice, this means that alternatives within the same nest tend to get a more than proportional share when there are substitutions, and alternatives in other nests tend to receive a less than proportional share. There is no correlation structure per se in the model, but rather a more sophisticated representation of the decision-making process.

A. Páez and G. Boisjoly, *Discrete Choice Analysis with R*, Use R!,
https://doi.org/10.1007/978-3-031-20719-8_8

In this chapter, an alternative to GEV-based models is introduced for even more flexible substitution patterns. This model is based on the use of the normal distribution for the random utility instead of the extreme value distribution. The joint mulivariate normal distribution naturally accommodates correlations between the alternatives. The model derived in this way is called the *multinomial probit* and can be used to represent arbitrary substitution patterns. This makes the probit model extremely flexible, but as we will see, this flexibility comes at a price, with considerable technical and computational demands.

8.2 Preliminaries

To introduce the probit model, We will return to the example of heating systems that we previously used to illustrate the nested logit model. Recall that this data set concerns a choice situation with five distinct heating systems:

- Gas Central (gc)
- Gas Room (gr)
- Electric Central (ec)
- Electric Room (er)
- Heat Pump (hp)

The results obtained with the nested logit model suggested associations between central systems and between room-based systems, and therefore, a nested logit model was an improvement over the multinomial logit model.

We begin by loading the packages used in this chapter

```
library(dplyr) # A Grammar of Data Manipulation
library(ggplot2) # Create Elegant Data Visualisations Using the Grammar of Graphics
library(kableExtra) # Construct Complex Table with 'kable' and Pipe Syntax
library(mlogit) # Multinomial Logit Models
```

In addition, we load the data set used in this section (from the `mlogit` package)

```
data("Heating")
```

The data set is in "wide" format, which means that there is one record per decision-making unit (i.e. per household), so we need to change the data to "long" format:

```
H <- mlogit.data(Heating,
                 shape = "wide",
                 choice = "depvar",
                 varying = c(3:12))
```

8.3 How to Use This Note

Every chunk of code, including the following, is available for practice purposes:

```
print("If you are always trying to be normal, you will never know how amazing you can be.")
```

```
[1] "If you are always trying to be normal, you will never know how amazing you can be."
```

If you are working with RStudio or other graphical interface you can type the chunks of code to experiment with them. As an alternative, you may copy and paste the source code into your `R` or RStudio console, or create a script/notebook to save the code and any experiments you conduct.

8.4 Learning Objectives

In this chapter, you will learn about:

1. The fundamental concepts of the multinomial probit
2. The estimation of a multinomial probit
3. Substitution patterns.

8.5 Suggested Readings

- Ben-Akiva, M., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and applications to travel demand* (Chap. 5, pp. 128–129). MIT Press.
- Ortuzar, J. D., & L. G. (2011). *Modelling transport*, 4th edn., (Chap. 7, pp. 248–250). Wiley.
- Train, K. (2009). *Discrete choice methods with simulation*, 2nd edn., (Chap. 5). Cambridge University Press.

8.6 Fundamental Concepts

Let us return to the fundamental concepts of discrete choice modelling. Recall that, to derive a discrete choice model, we state a choice problem for decision-maker n when they are faced with J discrete alternatives. We assume that the utility of each alternative is decomposed in two parts, a systematic utility V and a random utility ϵ. As an example, think of the alternatives in the example of the heating systems as follows:

$$
\begin{aligned}
U_{ec} &= V_{ec} + \epsilon_{ec}\\
U_{er} &= V_{er} + \epsilon_{er}\\
U_{gc} &= V_{gc} + \epsilon_{gc}\\
U_{gr} &= V_{gr} + \epsilon_{gr}\\
U_{hp} &= V_{hp} + \epsilon_{hp}
\end{aligned}
$$

Imagine that the systematic utilities V_j (where j represents the alternatives) are perfectly specified in the empirical context of interest, meaning that every attribute that matters from the perspective of the utility has been included in V and the form of the function is correct. In this case, the terms ϵ_j capture only idiosyncratic variations in utility that are by definition random. The multinomial logit model is in that case an appropriate modelling approach, besides being convenient due to its closed form and ease of implementation.

Imagine now that the specification of the systematic utilities *misses* something relevant that influences the utility of *some* of the alternatives. In the example, this could be something that relates to the type of installation of the heating systems:

$$
\begin{aligned}
U_{ec} &= V_{ec} + \epsilon_{ec} + \mu_c\\
U_{er} &= V_{er} + \epsilon_{er} + \mu_r\\
U_{gc} &= V_{gc} + \epsilon_{gc} + \mu_c\\
U_{gr} &= V_{gr} + \epsilon_{gr} + \mu_r\\
U_{hp} &= V_{hp} + \epsilon_{hp} + \mu_c
\end{aligned}
$$

The terms μ_c are the part of the systematic utility that is *missing*, and that relate to the preference for central heating systems, whereas μ_r is the part that is missing from the systematic utility of room-based systems. The fact that these terms are missing from the specification of the systematic utility does not mean that they do not matter; by definition, if they affect the utility, they must be somewhere, and that somewhere ends up being the rest of the utility, so that:

$$
\begin{aligned}
\tilde{\epsilon}_{ec} &= \epsilon_{ec} + \mu_c\\
\tilde{\epsilon}_{er} &= \epsilon_{er} + \mu_r\\
\tilde{\epsilon}_{gc} &= \epsilon_{gc} + \mu_c\\
\tilde{\epsilon}_{gr} &= \epsilon_{gr} + \mu_r\\
\tilde{\epsilon}_{hp} &= \epsilon_{hp} + \mu_c
\end{aligned}
$$

As seen above, the utility terms that are supposed to be random are no longer independent due to the common omitted but relevant terms. The utilities will tend to vary following the peculiar patterns of μ_c and μ_r: there is an affinity for central systems, which are perceived as being more similar between them, and the same thing happens with room systems. In other words, the omitted but relevant terms lead to a situation where the random utilities tend to *covary*, that is, to display a pattern of correlation. This is the reason why substitution patterns are not sensible when unaccounted-for correlations lurk in the background: when the attributes of one alternative change (or the alternative disappears altogether), there might be a stronger preference for alternatives that are perceived to be more similar to it due to a common missing term.

The nested logit model solves this situation by grouping similar alternatives within nests. If the nesting structure matches the pattern of the missing elements, then the model can provide a better approximation of the underlying preferences than the logit model, which assumes that the utilities of the alternatives are uncorrelated.

We can point at least two issues with the implementation of the nested logit model: (1) the nesting structure must be decided a priori by the analyst, and it is possible that the chosen structure does not match the underlying pattern of correlations; and (2) as we saw in Chap. 7, the nested logit model replicates a multinomial structure within nests, together with IIA and proportional substitution patterns. This implies that alternatives within the same nest are all considered equally similar. Conversely, all alternatives that are not within the same nest are considered to be fully independent.

The probit model is a flexible model that can address these issues. To derive the model, we begin by collecting the "random" utility terms as follows for decision-maker n (i.e., there is one random utility term for each alternative)

$$\tilde{\epsilon}_n = [\tilde{\epsilon}_1, \ldots, \tilde{\epsilon}_j, \ldots, \tilde{\epsilon}_J]$$

We keep in mind that these random utility terms may include some omitted but relevant aspects of the utility.

Previously, in Chap. 4, the logit model was derived based on the assumption that the random utility terms ϵ_j followed the Extreme Value Type I distribution, and that they were independent from each other (i.e., the random utility terms are uncorrelated). Based on this, the difference of random utilities followed the logistic distribution, hence the name "logit" for the model. In the case of the probit model, we turn to the assumption that the vector of random utilities $\tilde{\epsilon}_n$ follows the normal distribution with a mean of zero and a covariance matrix Σ_n

$$\tilde{\epsilon}_n \sim N[0, \Sigma_{\tilde{\epsilon}}]$$

The covariance matrix here is the key to the flexibility of the probit model. It defines, in a fairly general way, the patterns of covariation, and therefore, the correlations between random utility terms. Put simply, the covariance indicates whether jointly distributed random variables (here the random utility terms) relate in a systematic fashion. For two alternatives that are similar (say, the two room-based heating systems), we would expect that if a common part of the utility is missing, the random utility terms of these alternatives will tend to vary in the same direction, and therefore, have a high covariance. In contrast, if the missing term does not influence the utility of, say, heat pump system, we would expect any room-based alternative to have a low covariance with the heat pump system.

The covariance matrix of a model with J alternatives has $J(J+1)/2$ unique covariance terms, including J variance terms in its main diagonal. Let us take as an example a situation with four alternatives, in which case we would have four variance terms and six covariance terms. The covariance matrix is as follows:

$$\Sigma_{\tilde{\epsilon}} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ \cdot & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ \cdot & \cdot & \sigma_{33} & \sigma_{34} \\ \cdot & \cdot & \cdot & \sigma_{44} \end{bmatrix}$$

While covariance matrices can be specific to individuals, here we omit the subscript n for simplicity.

Based on the above, the probability distribution function of the random terms is the following:

$$\phi(\tilde{\epsilon}_n) = \frac{1}{(2\pi)^{J/2}|\Sigma_{\tilde{\epsilon}}|^{1/2}} e^{-\frac{1}{2}\tilde{\epsilon}_n^T \Sigma_{\tilde{\epsilon}}^{-1} \tilde{\epsilon}_n}$$

Recall that in the case of a logit model, the random terms were distributed following the EV Type I probability distribution function ($f(x; \mu, \sigma) = e^{-(x+e^{-(x-\mu)/\sigma})}$), where the distribution only depends on two parameters, μ and σ. In contrast, as we can see from the above joint distribution, the terms $\epsilon_n = [\epsilon_1, \dots, \epsilon_j, \dots, \epsilon_J]$ are all related through the common covariance matrix, and are independent *only if* the covariance matrix is diagonal, in which case the distribution simplifies to

$$\phi(\tilde{\epsilon}_n) = \prod_{j}^{J} \frac{1}{\sigma_j \sqrt{2\pi}} e^{-\frac{1}{2\sigma_j}\tilde{\epsilon}_n^2}$$

As noted before, the covariance matrix can be specific to individual n, if it includes attributes that vary across individuals. In this way, the probit model can also account for taste variation between individuals. Taste variation will be discussed in more detail in later chapters; here, we will focus on the correlations between error terms and how this provides a way to model flexible substitution patterns.

At this point, it is important to remember that a discrete choice model works on the *differences* between utilities, so the covariance matrix above cannot be used directly. Fortunately, we know that the difference between two normal distributions also follows a normal distribution, so

$$\bar{\epsilon} = \tilde{\epsilon}_j - \tilde{\epsilon}_i \sim N[0, \Sigma_{\bar{\epsilon}}]$$

where $\bar{\epsilon}$ is the vector of differences between random utilities, i.e. $\tilde{\epsilon}_j - \tilde{\epsilon}_j, \forall j \neq i$. The joint probability distribution of the *differences* between the error terms follows a similar structure as the one for the error terms themselves, but with $J - 1$ dimensions

$$\Sigma_{\bar{\epsilon}} = \begin{bmatrix} \theta_{22} & \theta_{23} & \theta_{24} \\ \cdot & \theta_{33} & \theta_{34} \\ \cdot & \cdot & \theta_{44} \end{bmatrix}$$

The terms in this matrix relate to the terms of the original covariance matrix as follows (see Train 2009, p. 102):

$$\begin{aligned}
\theta_{22} &= \sigma_{22} + \sigma_{11} - 2\sigma_{12} \\
\theta_{23} &= \sigma_{23} + \sigma_{11} - \sigma_{12} - \sigma_{13} \\
\theta_{24} &= \sigma_{24} + \sigma_{11} - \sigma_{12} - \sigma_{14} \\
\theta_{33} &= \sigma_{33} + \sigma_{11} - 2\sigma_{13} \\
\theta_{34} &= \sigma_{34} + \sigma_{11} - \sigma_{13} - \sigma_{14} \\
\theta_{44} &= \sigma_{44} + \sigma_{11} - 2\sigma_{14}
\end{aligned}$$

As was the case of the logit model, the scale of the utility (the variance) needs to be normalized (the scale cannot be uniquely identified, and it is irrelevant to the choice mechanism). To do so, one of the variance terms (in the diagonal) is set to 1 and the other terms are normalized after it. We then obtain the following normalized matrix:

$$\bar{\Sigma}_{\bar{\epsilon}} = \begin{bmatrix} \lambda_{22} & \lambda_{23} & \lambda_{24} \\ \cdot & \lambda_{33} & \lambda_{34} \\ \cdot & \cdot & 1 \end{bmatrix}$$

We now have a covariance matrix with five distinct parameters, instead of the 10 elements in the original covariance matrix. After obtaining the covariance matrix of the differences of utilities and normalizing the scale of the utility, the terms of the normalized covariance matrix $\bar{\Sigma}_{\bar{\epsilon}}$ relate to the original covariance matrix of the random utility terms in the following way:

$$\begin{aligned}
\lambda_{22} &= \tfrac{\sigma_{22}+\sigma_{11}-2\sigma_{12}}{\sigma_{44}+\sigma_{11}-2\sigma_{14}} \\
\lambda_{23} &= \tfrac{\sigma_{23}+\sigma_{11}-\sigma_{12}-\sigma_{13}}{\sigma_{44}+\sigma_{11}-2\sigma_{14}} \\
\lambda_{24} &= \tfrac{\sigma_{24}+\sigma_{11}-\sigma_{12}-\sigma_{14}}{\sigma_{44}+\sigma_{11}-2\sigma_{14}} \\
\lambda_{33} &= \tfrac{\sigma_{33}+\sigma_{11}-2\sigma_{13}}{\sigma_{44}+\sigma_{11}-2\sigma_{14}} \\
\lambda_{34} &= \tfrac{\sigma_{34}+\sigma_{11}-\sigma_{13}-\sigma_{14}}{\sigma_{44}+\sigma_{11}-2\sigma_{14}}
\end{aligned}$$

The original parameters σ_{ij} of the covariance matrix of the random utility terms had a straightforward interpretation as the variance/covariance between the random utilities of the alternatives. An unfortunate aspect of the process of working with the differences between utilities and normalizing the scale of the utility, is that the elements of $\bar{\Sigma}_{\bar{\epsilon}}$ lack a straightforward interpretation. Worse, there are five estimable parameters in $\bar{\Sigma}_{\bar{\epsilon}}$, but there are ten parameters in $\Sigma_{\bar{\epsilon}}$, so there is no way to uniquely identify those ten parameters, and they must remain confounded by the way they enter the various λ expressions.

Once we have a workable covariance matrix, we are ready to express the joint distribution of the differences between random utilities:

$$\phi(\bar{\epsilon}_n) = \frac{1}{(2\pi)^{(J-1)/2}|\bar{\Sigma}_{\bar{\epsilon}}|^{1/2}} e^{-\frac{1}{2}\bar{\epsilon}_n^T \bar{\Sigma}_{\bar{\epsilon}}^{-1} \bar{\epsilon}_n}$$

As we have seen in previous chapters, the probability of choosing alternative j is given in terms of the difference of the random utilities

$$P_j = P(V_j - V_k \leq \bar{\epsilon}_n)$$

This involves integration of the probability distribution. As seen in Chap. 3, one appealing aspect of logit model is that the relevant integral has an analytical solution, or so-called closed form as follows:

$$F(x; \mu, \sigma) = \frac{1}{1 + e^{-(\epsilon_n - \mu)/\sigma}}$$

Alas, the normal distribution has no such analytical solution. The probability of choosing alternative j is equal to the integral of the multivariate normal distribution function. The number of integrals corresponds to $J - 1$. In the case of four alternatives, the probability of choosing alternative 1 is given by the following integral:

$$P_1 = \int_{V_2 - V_1}^{\infty} \int_{V_3 - V_1}^{\infty} \int_{V_4 - V_1}^{\infty} \phi(\bar{\epsilon}_2, \bar{\epsilon}_3, \bar{\epsilon}_4) | \bar{\Sigma}_{\bar{\epsilon}}) \mathrm{d}\bar{\epsilon}_4 \mathrm{d}\bar{\epsilon}_3 \mathrm{d}\bar{\epsilon}_2$$

where each $\bar{\epsilon}_i$ represents the difference between the random utility of alternative j with respect to alternative 1.

8.7 Estimation of Probit Models

The lack of a closed form for the probit probability integral means that it needs to be approximated using numerical techniques, which can make the model computationally expensive. Scanning Train (2009, pp. 114–15) and Ortúzar and Willumsen (2011, Fourth Edition: pp. 292–93) two non-simulation approaches appear to be of use to numerically calculate the integral: quadrature methods and Clark's algorithm (see Clark 1961; Daganzo et al. 1977).

Quadrature methods are a kind of "brute force" approach, and they depend on evaluating the integral at very small intervals and then aggregating the areas. For one dimensional integrals, the word "quadrature" refers to forming slim rectangles that approximate the function at a specific value of the variable. In higher dimension, this is sometimes referenced as "cubature", essentially the same idea but making thin elongated cubes for two-dimensional integrals, and hypercubes for higher dimensional integrals. According to Ortúzar and Willumsen (2011, Fourth Edition pp. 292–93) this is the most accurate way of calculating the integral, but it is not practical for $J > 4$, and may lead to numerical errors if some probabilities are close to zero.

The other non-simulation method is based on Clark's finding that the maximum of several normally distributed variables approximates a normal distribution

(Clark 1961). The operative word here is “approximates”, and Train (2009, p. 115) notes that the approximation can at times be very inaccurate.

Instead of these approaches, estimation of probit models usually relies on simulation techniques. We will not go into much detail about simulation approaches, and Train (2009) covers them in great detail in Chaps. 5, 9, and 10 of his wonderful book. But, it is worthwhile to at least present the intuition behind simulation approaches.

As noted in Chap. 3, a geometric interpretation of an integral is as the area (or volume) under a function. When the function involves only straight lines, calculating the areas is simple; the challenge is when the functions are curves. One approach to approximate the area under the curve is as follows.

First, we create a data frame with values from the normal distribution, and compute the density using a fine grid

```
norm_pdf <- data.frame(x = seq(0.2,
                               1.2,
                               0.01)) %>%
  mutate(f = dnorm(x,
                   mean = 0,
                   sd = 0.5))
```

We can frame the segment of the curve that we are interested by means of a rectangle: all we need is the interval in x that we are interested in, and the maximum value of the function in that interval (the minimum value of course is zero). This is shown in Fig. 8.1.

```
ggplot(data = norm_pdf,
       aes(x = x, y = f)) +
  geom_rect(aes(xmin = 0.2, xmax = 1.2,
                ymin = 0.0, ymax = max(f)),
            fill = NA,
            color = "black") +
  geom_line()
```

Calculating the area of the rectangle is easy: base times height (which in this example gives 0.7365). But how much of that area is *under* the curve? A form of accept–reject simulation is as follows:

1. Throw a random point in the space of the rectangle (or cube/hypercube) that contains the segment of curve of interest.
2. Check whether the value on the dimension of f is less than the value of the function for the corresponding x.
3. If it is, label the point as “accept”, otherwise as “reject”.
4. Repeat this procedure r times.
5. The approximation of the area under the curve is the proportion of times a point was accepted times the area of the frame.

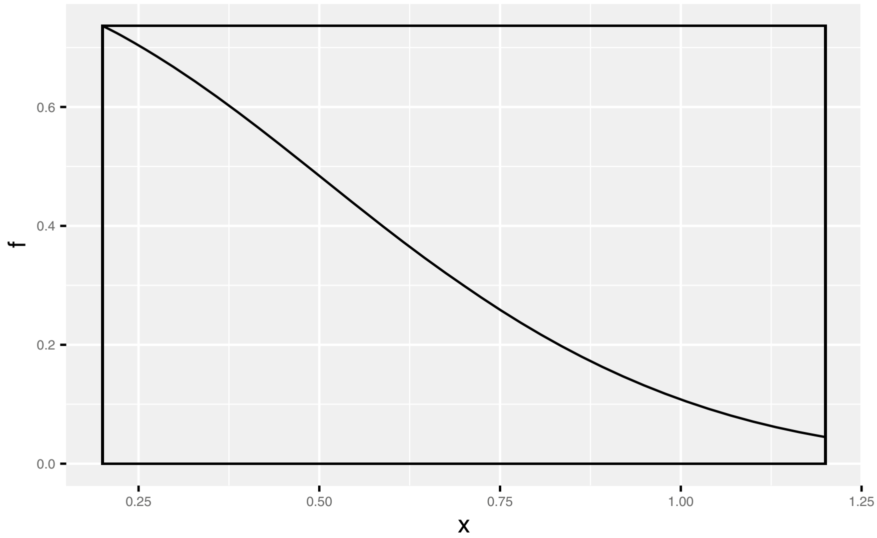


Fig. 8.1 A segment of the normal curve with mean of zero and standard deviation 0.5

Here we approximate the area under the curve by means of $r = 100$ random draws

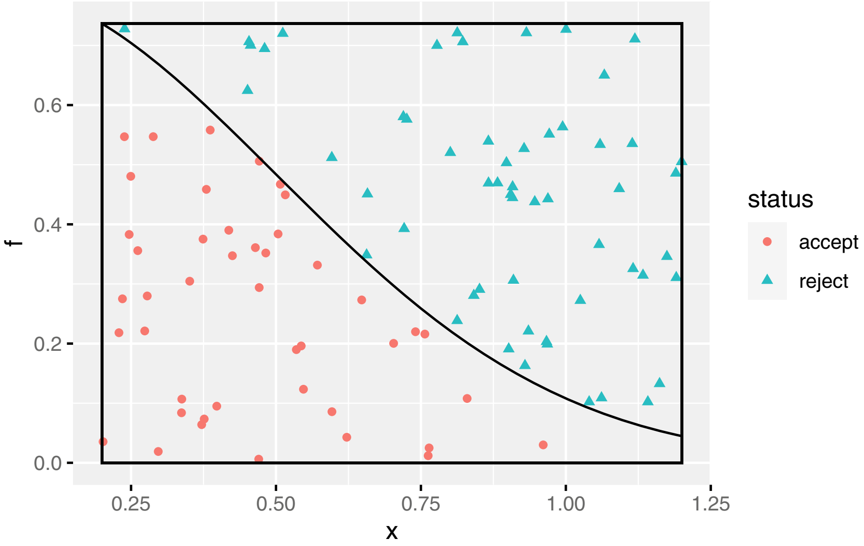
```
# Set seed for replicability (this defines a starting number from which the sequence of random numbers is generated)
set.seed(35739)

# Set number of random draws
r <- 100
sim_pdf <- data.frame(x = runif(r,
                                # Draw values with uniform probability in the
                                # interval of xmin and xmax
                                min(norm_pdf$x),
                                max(norm_pdf$x)),
                      fsim = runif(r,
                                   # Draw values with uniform probability in the
                                   # interval between zero and fmax
                                   0,
                                   max(norm_pdf$f)))

# Clear the seed
set.seed(NULL)
```

The accept–reject part is as follows: we take the simulated draws and compare them to the curve.

```
sim_pdf <- sim_pdf %>%
  mutate(f = dnorm(x,
                   mean = 0,
                   sd = 0.5),
         status = ifelse(fsim <= f,
                         "accept",
                         "reject"))
```

We then plot the simulated points

```
ggplot(data = norm_pdf,
       aes(x = x, y = f)) +
  geom_point(data = sim_pdf,
             aes(x = x, y = fsim,
                 color = status,
                 shape = status))  +
  geom_line() +
  geom_rect(aes(xmin = 0.2, xmax = 1.2,
                ymin = 0.0, ymax = max(f)),
            fill = NA,
            color = "black")
```



The proportion of “accepted” draws out of all draws is 0.44. According to our simulation, *approximately* 44% of the area of the rectangle lies under the curve. Knowing that the area of the rectangle is 0.7365, the area under the curve can be approximated to 0.3241). As a reference, the “exact” area under the curve (calculated using function `pnorm()`) is 0.3364. Again, the operative word when talking about our simulation is *approximately*, and the quality of the approximation will depend on the number of draws (repeat the experiment but now with $r = 1000$ and $r = 10000$).

You can use this chunk of code to assess the quality of the approximation

```
# Proportion of "accepted" draws
prop_accept <- sim_pdf %>% filter(status == "accept") %>% nrow()/r

# Approximation of area under the curve
(max(norm_pdf$x) - min(norm_pdf$x)) * max(norm_pdf$f) * (prop_accept)
```

```
[1] 0.3240777
```

```
# Exact area
pnorm(max(norm_pdf$x),
      mean = 0,
      sd = 0.5) - pnorm(min(norm_pdf$x),
                        mean = 0,
                        sd = 0.5)
```

```
## [1] 0.3363807
```

The procedure described above is not used in practice, since the randomness might mean poor coverage of some regions, and depending on the number of draws, a bad simulation can result in very misleading results. Other approaches that increase the coverage and reduce the variance are discussed in Train (2009). The algorithm used by the package {mlogit} is GHK, which operates on the differences of utilities (see Train 2009, Sect. 5.6.3).

8.8 Example

Armed with the theory and at least some conceptual understanding of simulation for estimation, we are now ready to illustrate how to estimate a multinomial probit model. For comparison purposes, we will re-estimate the multinomial logit model of Chap. 6 and the nested logit model of Chap. 7

```
# Time the process
start_time_mnl <- Sys.time()

model3 <- mlogit(depvar ~ ic + oc,
                 Heating,
                 reflevel = "gc",
                 shape = "wide",
                 choice = "depvar",
                 varying = c(3:12))

# Time the process
end_time_mnl <- Sys.time()
estimation_time_mnl <- end_time_mnl - start_time_mnl
```

```
# Time the process
start_time_nl <- Sys.time()

nl2 <- mlogit(depvar ~ ic + oc,
              H,
              reflevel = "gc",
              nests = list(room = c( 'er', 'gr'),
                           central = c('ec', 'gc', 'hp')),
              un.nest.el = TRUE,
              steptol = 1e-12)

# Time the process
end_time_nl <- Sys.time()
estimation_time_nl <- end_time_nl - start_time_nl
```

Notice that we timed the estimation of these two models. Estimating the multinomial logit model took approximately 0.1264 s. Estimation of the nested logit took 0.531 s. Estimation of the nested logit model is usually more time intensive, but still within reasonable bounds. Note that the estimation times may vary from one system to another. If you are working with the code, you can go ahead and compare the estimation times of the two models in your own system.

The multinomial probit model is estimated with the `mlogit()` function of the {mlogit} package, by specifying `probit = TRUE`. This option allows the estimation of a fully flexible correlation structure among alternatives. By default, all parameters of the normalized matrix (as described in the previous section) are estimated.

In the example below, we set gas central as the reference. The number of random draws for the numerical simulation is set to $r = 50$ and, for replicability, we also specify a random seed for the simulation. This ensures that the results can be replicated. You will see that estimating a probit model can be time intensive (Table 8.1).

```
# Time the process
start_time_prob1 <- Sys.time()

prob1 <- mlogit(depvar ~ ic + oc,
                H,
                reflevel = "gc",
                probit = TRUE,
                seed = 3245,
                R = 50)

# Time the process
end_time_prob1 <- Sys.time()
estimation_time_prob1 <- end_time_prob1 - start_time_prob1

stargazer::stargazer(prob1,
                     header = FALSE,
                     single.row = TRUE,
                     title = "Estimation results: Multinomial Probit Model 1")
```

The time needed to estimate multinomial probit model 1 is 4.6967 min, that is, at least two orders of magnitude greater than the time needed to estimate either the multinomial or hierarchical logit models. Remember, if you are working with the Rnotebook and estimate the model, you might get different estimation times than those mentioned here, as they will vary from one system to another.

Table 8.1 Estimation results: multinomial probit model 1

	Dependent variable
	Depvar
(Intercept):ec	0.005 (0.416)
(Intercept):er	0.093 (0.402)
(Intercept):gr	−15.980 (15.577)
(Intercept):hp	−1.345* (0.762)
ic	−0.002*** (0.001)
oc	−0.004*** (0.001)
ec.er	−0.050 (0.938)
ec.gr	0.014 (4.446)
ec.hp	−0.085 (0.131)
er.er	1.318** (0.540)
er.gr	9.327 (8.765)
er.hp	−0.886** (0.412)
gr.gr	12.667 (12.122)
gr.hp	1.011** (0.447)
hp.hp	0.002 (0.080)
Observations	900
R^2	0.021
Log Likelihood	−1,001.260
LR Test	41.928*** (df = 16)

Note—*p<0.1; **p<0.05; ***p<0.01

The results of the model are consistent with our earlier findings with the multinomial and hierarchical logit models: both the installation and operation costs are negative and significantly different from zero. It is important to mention here that we cannot compare the magnitude of coefficients between different types of models (e.g., probit vs logit), because different scales are used.

We will now examine the covariance matrix associated with this model. As mentioned earlier, interpretation of the coefficients of the model is not straightforward. The covariance matrix of the utility differences can be built using the parameters estimated above.

```r
# Initialize a 4-by-4 matrix for the covariance terms
L1 <- matrix(0, 4, 4)

# Assign the coefficients to the matrix (lower part)
L1[!upper.tri(L1)] <- c(1, coef(prob1)[7:15])

# Multiply the lower part matrix by its transpose to fill the upper diagonal
L1 %*% t(L1)
```

```
             [,1]        [,2]         [,3]        [,4]
[1,]  1.00000000 -0.04966945   0.01389838 -0.08527967
[2,] -0.04966945  1.73937351  12.29176610 -1.16348024
[3,]  0.01389838 12.29176610 247.44170872  4.54087388
[4,] -0.08527967 -1.16348024   4.54087388  1.81448045
```

Notice that the normalization was applied to the first element in the main diagonal of the matrix. As seen in the results of the estimation, some (but not all!) elements of the covariance matrix are significantly different from zero at conventional levels of confidence. How many alternatives are *not* correlated through their random utilities? The covariance matrix of the random utility terms for this model is as follows:

$$\Sigma_{\tilde{\epsilon}} = \begin{bmatrix} \sigma_{gc.gc} & \sigma_{gc.ec} & \sigma_{gc.er} & \sigma_{gc.gr} & \sigma_{gc.hp} \\ \cdot & \sigma_{ec.ec} & \sigma_{ec.er} & \sigma_{ec.gc} & \sigma_{ec.hp} \\ \cdot & \cdot & \sigma_{er.er} & \sigma_{er.gc} & \sigma_{er.hp} \\ \cdot & \cdot & \cdot & \sigma_{gr.gr} & \sigma_{gr.hp} \\ \cdot & \cdot & \cdot & \cdot & \sigma_{hp.hp} \end{bmatrix}$$

However, this is not the covariance matrix that is estimated; instead, it is the covariance matrix of the differences of utilities *after normalizing the scale*. We can see from the results that the term that was normalized is the variance of $\lambda_{ec.ec}$

$$\bar{\Sigma}_{\tilde{\epsilon}} = \begin{bmatrix} 1 & \lambda_{ec.er} & \lambda_{ec.gr} & \lambda_{ec.hp} \\ \cdot & \lambda_{er.er} & \lambda_{er.gr} & \lambda_{er.hp} \\ \cdot & \cdot & \lambda_{gr.gr} & \lambda_{gr.hp} \\ \cdot & \cdot & \cdot & \lambda_{hp.hp} \end{bmatrix}$$

Let us consider only the terms of the covariance matrix that are significantly different from zero, namely $\lambda_{er.er}$, $\lambda_{er.hp}$, and $\lambda_{gr.hp}$. These terms relate to the covariance matrix of the random utilities as follows (see Train 2009, pp. 99–100, for an explanation of the method to transform the covariance matrix):

$$\begin{aligned} \lambda_{er.er} &= \frac{\sigma_{gc.gc}+\sigma_{er.er}-2\sigma_{er.gc}}{\sigma_{gc.gc}+\sigma_{ec.ec}-2\sigma_{gc.ec}} \\ \lambda_{er.hp} &= \frac{\sigma_{gc.gc}+\sigma_{er.hp}-\sigma_{er.gc}-\sigma_{gc.hp}}{\sigma_{gc.gc}+\sigma_{ec.ec}-2\sigma_{gc.ec}} \\ \lambda_{gr.hp} &= \frac{\sigma_{gc.gc}+\sigma_{gr.hp}-\sigma_{gr.gc}-\sigma_{gc.hp}}{\sigma_{gc.gc}+\sigma_{ec.ec}-2\sigma_{gc.ec}} \end{aligned}$$

The above makes clear that even when we consider only the significant elements of the estimated covariance matrix, these involve nine out of fifteen distinct terms of the original covariance matrix of the random utilities, indicating a complex pattern of correlations. Unfortunately, we have only three significant values of λ and nine σ terms, which means that the original correlations cannot be identified. Had all covariance terms been significant in the estimated matrix, we would still have only nine numerical values but fifteen terms in the original matrix. Given the inability to retrieve those values, and the complex way they interact in the estimable matrix, the covariance parameters of a probit model are fundamentally uninterpretable.

It is possible to impose restrictions on the covariance matrix by setting some parameters to fixed values or by specifying relations between the elements. This allows an analyst to specify how the errors terms relate to each others. The normalization procedure presented above ensures that all parameters are identified, that is, that they can be estimated. The same cannot be said when restrictions are imposed on the covariance matrix. However, this process is fraught and it is possible (as shown by Train 2009, pp. 102–6) to impose restrictions that fail to normalize the matrix, thus leading to unidentifiable parameters. The identifiability of the model depends on what is being restricted, and a challenge is that the presence of unidentified parameters is often not intuitive, given the complexity associated with the covariance matrix.

8.9 What About the Substitution Patterns?

Since it is difficult to meaningfully interpret the coefficients of the covariance matrix, scenario analyses are particularly relevant to understand the output of a probit model. To understand the substitution patterns associated with a probit model, scenario analyses are especially helpful.

We will simulate the following scenario: the installation costs of the gas central system increases by 15%. We will then compare the probabilities of the current situation with those of the new scenario.

To simulate this scenario, we begin by copying the input data frame:

```
H_increase <- H
```

In the new data frame that will simulate the increase in cost, we replace the cost of installation as follows:

```
H_increase[H_increase$alt == "gc", "ic"] <- 1.15 * H_increase[H_increase$alt == "gc", "ic"]
```

We can calculate the market shares of the “do nothing” and “increase” scenario and compare their shares (which are the mean values of the predictions), based on different estimated models.

Let us start with the multinomial logit model

```
scenario_mnl <- data.frame(Policy = c("Do nothing", "15% increase"),
           rbind(apply(predict(model3,
                               newdata = H),
                       2,
                       mean),
                 apply(predict(model3,
                               newdata = H_increase),
                       2,
                       mean)))

scenario_mnl
```

```
         Policy        gc         ec         er        gr         hp
1   Do nothing 0.6366667 0.07111111 0.09333333 0.1433333 0.05555556
2 15% increase 0.5949957 0.07904654 0.10375571 0.1601958 0.06200620
```

We can calculate the percentage change in shares that results from the increase in installation costs of gas central systems

```
Ratios_mnlogit <- data.frame((scenario_mnl[2,2:6] - scenario_mnl[1,2:6])/scenario_mnl[1,2:6] * 100)
row.names(Ratios_mnlogit) <- c("Percentage change in shares")
Ratios_mnlogit
```

```
                                   gc      ec       er       gr       hp
Percentage change in shares -6.545176 11.1592 11.16683 11.76454 11.61116
```

The results suggest a decrease of about 6.5% in the share of gas central when installation costs of these systems increase by 15%. The changes in the shares of other systems reflect the proportional substitution patterns, and each alternative grows by approximately 11% of their initial share.

Next, we repeat the exercise with the nested logit

```
scenario_nl <- data.frame(Policy = c("Do nothing", "15% increase"),
                       rbind(apply(predict(nl2, newdata = H), 2, mean),
                             apply(predict(nl2, newdata = H_increase), 2, mean)))

scenario_nl
```

```
         Policy        gc         ec         er        gr         hp
1   Do nothing 0.6365696 0.07142767 0.08818753 0.1484775 0.05533768
2 15% increase 0.5947078 0.09373354 0.08870532 0.1493938 0.07345960
```

The substitution patterns are no longer proportional between nests, even if they remain constant within nests

```
Ratios_nlogit <- data.frame((scenario_nl[2,2:6] -
                               scenario_nl[1,2:6])/scenario_nl[1,2:6] * 100)
row.names(Ratios_nlogit) <- c("Percentage change in shares")
Ratios_nlogit
```

```
                                   gc       ec        er        gr       hp
Percentage change in shares -6.576165 31.22861 0.5871416 0.6171279 32.74788
```

We see that the model predicts a drop in the share of gas central of approximately 6.6%, which is very close to that predicted by the multinomial logit model. But central systems are the biggest winners with percentual increases of approximately 32% over their initial shares, while the share of room systems barely budges with tiny increases of about 0.6%.

Finally, let us assess the substitution patterns associated with the multinomial probit model

```
scenario_prob1 = data.frame(Policy = c("Do nothing", "15% increase"),
                         rbind(apply(predict(prob1,
                                             newdata = H),
                                     2,
                                     mean),
                               apply(predict(prob1,
                                             newdata = H_increase),
                                     2,
                                     mean)))
scenario_prob1
```

```
        Policy        gc         ec         er        gr         hp
1   Do nothing 0.6356646 0.07268863 0.09167621 0.1477090 0.05316574
2 15% increase 0.5729279 0.09528045 0.11458619 0.1490408 0.06764994
```

We then calculate the percentage change in the predicted shares

```
Ratios_probit <- data.frame((scenario_prob1[2,2:6] -
                               scenario_prob1[1,2:6])/scenario_prob1[1,2:6] * 100)
row.names(Ratios_probit) <- c("Ratio of probabilities")
Ratios_probit
```

```
                              gc       ec       er        gr       hp
Ratio of probabilities -9.869474 31.08027 24.99009 0.9016318 27.24347
```

The multinomial probit model predicts a bigger loss in the share of gas central systems, at about 9.9% (compared to less than 7% according to the two logit models). The more flexible correlation patterns, in addition, mean that there is no proportional substitution at all: the largest gains in shares are for central systems, but unlike the nested logit model, electric room systems also see large gains but gas room systems again barely budge, which suggests that gas room systems are not perceived as approximate substitutes for gas central systems.

8.10 More About Simulation-Based Estimation

It is worthwhile highlighting here that the results of the multinomial and nested logit model are *exact* solutions of the underlying integrals. Simulation-based estimation, in contrast, is an approximation that depends on two exogenous parameters, as seen in the example: a random seed (which we recorded for replicability) and the number of random draws. As the accept–reject procedure above implied, the quality of the approximation depends on the number of random draws. Is it possible that increasing the number of random draws can improve the quality of the model (Table 8.2)? In the next chunk of code, we use the same random seed, but increase the number of random draws to $r = 150$ from $r = 50$

```
# Time the process
start_time_prob2 <- Sys.time()

prob2 <- mlogit(depvar ~ ic + oc,
                H,
                reflevel = "gc",
                probit = TRUE,
                seed = 3245,
                R = 150)

# Time the process
end_time_prob2 <- Sys.time()
estimation_time_prob2 <- end_time_prob2 - start_time_prob2

stargazer::stargazer(prob2,
                     header = FALSE,
                     single.row = TRUE,
                     title = "Estimation results: Multinomial Probit Model 2")
```

Is this a better or worse model? The answer is, *we don't know*. The models are identical but for the two *hyperparameters* (random seed and number of draws). Their log likelihoods are virtually identical. But the first multinomial probit model is superficially more appealing because more parameters *seem to be* significant.

Table 8.2 Estimation results: multinomial probit model 2

	Dependent variable
	Depvar
(Intercept):ec	0.065 (0.479)
(Intercept):er	0.186 (0.530)
(Intercept):gr	−36.917 (292.839)
(Intercept):hp	−1.186 (1.084)
ic	−0.002** (0.001)
oc	−0.004*** (0.001)
ec.er	0.232 (1.001)
ec.gr	9.599 (85.787)
ec.hp	−0.805 (1.542)
er.er	1.138* (0.646)
er.gr	14.643 (124.278)
er.hp	0.340 (1.668)
gr.gr	31.196 (245.251)
gr.hp	1.158 (1.519)
hp.hp	0.399 (1.501)
Observations	900
R^2	0.019
Log Likelihood	−1,002.568
LR Test	39.311*** (df = 16)

Note—*p<0.1; **p<0.05; ***p<0.01

Table 8.3 Estimation results: multinomial probit model 3

	Dependent variable
	Depvar
(Intercept):ec	0.112 (0.392)
(Intercept):er	0.007 (0.439)
(Intercept):gr	−18.329 (61.951)
(Intercept):hp	−2.459 (2.402)
ic	−0.002*** (0.001)
oc	−0.004*** (0.001)
ec.er	−0.503 (0.897)
ec.gr	5.646 (20.265)
ec.hp	0.734 (0.717)
er.er	1.323 (0.828)
er.gr	13.712 (45.280)
er.hp	−1.499 (1.262)
gr.gr	10.204 (34.292)
gr.hp	1.256 (1.152)
hp.hp	0.120 (0.784)
Observations	900
R^2	0.021
Log Likelihood	−1,001.194
LR Test	42.059*** (df = 16)

Note—*p<0.1; **p<0.05; ***p<0.01

However, consider the next model, again using $r = 50$ but with a different random seed (Table 8.3)

```
# Time the process
start_time_prob3 <- Sys.time()

prob3 <- mlogit(depvar ~ ic + oc,
                H,
                reflevel = "gc",
                probit = TRUE,
                seed = 3246,
                R = 50)

# Time the process
end_time_prob3 <- Sys.time()
estimation_time_prob3 <- end_time_prob3 - start_time_prob3

stargazer::stargazer(prob3,
                     header = FALSE,
                     single.row = TRUE,
                     title = "Estimation results: Multinomial Probit Model 3")
```

There is a temptation to think of the parameters of a model estimated using simulation as point estimates, when in fact they are random variables that depend on the chosen values for the simulation and the number of draws. For this reason it is more appropriate to think of the parameter estimates as a distribution. At the moment, with only three models, we are seeing only three realizations of that distribution so we cannot possibly know which of them is closer to the "mean" of the distribution: hence, our inability to declare a model better than the rest. Unfortunately, to be in a better position to do that, we would need to estimate a large number of models, and experimenting with various hyperparameters (for instance, doing a grid search) quickly becomes inconvenient: the time needed to estimate the multinomial probit model `Model3` was already 6.2549 min. To further compound things, good guidance in the literature in terms of the number of draws is lacking, and this hyperparameter will likely vary by empirical context.

This discussion should serve as a warning against the temptation to try to "hack" the random seed to produce results that are more "convincing". This should be avoided. On the flip side, a consumer of a multinomial probit model should avoid treating the reported parameters as point estimates, and always ask information about the simulation, including, for example, the simulation variance (which, as we noted, can be expensive to compute).

8.11 Final Remarks

This chapter introduced the multinomial probit model. Through the covariance matrix, the probit model allows for flexible correlation structures and non-proportional substitution patterns. In addition, it can accommodate varying preferences across individuals. This flexibility comes at a cost: probit models are computationally more expensive than other models. As well, specifying and interpreting a probit model is not necessarily intuitive. Fortunately, other models—building on some of the concepts introduced here—have been developed to account for taste heterogeneity and correlations among alternatives, while providing results that can be more easily interpreted. These models are presented in the next chapters.

8.12 Exercises

1. Repeat the illustrative accept-reject simulation multiple times (e.g., $s = 1000$) using $r = 100$ and summarize the results in the form of a histogram or frequency polygon. Next, do the same experiment but with $r = 1000$ and $r = 5000$. Compare to the exact area and discuss your results.

Hint: You can simultaneously draw $s \times r$ random numbers and then collect them in s groups (see package {dplyr} functions `group_by()` and `summarize()`).

1. Re-estimate the multinomial probit model in this chapter, still with $r = 50$ but using a different random seed. Do this several times. The estimated values of the parameters will change, which is to be expected…but which ones change more? Why do you think this happens?
2. Use the models you estimated in Question 2 and repeat the scenario where the installation cost of gas central systems increases by 15%. What can you say about the patterns of substitution when you calculate them with different models? What would you do to be more confident about the estimated patterns of substitution?
3. Re-estimate the multinomial probit model in this chapter with $r = 50$ and the seed set to 3245, but change the reference level to "ec". What happens? What do you think are the implications for experimenting with multinomial probit model specifications?

References

Ben-Akiva, M., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and applications to travel demand*. Cambridge: The MIT Press.

Clark, Charles E. (1961). The greatest of a finite set of random variables. *Operations Research, 9*(2), 145–62.

Daganzo, Carlos F., Bouthelier, Fernando, & Sheffi, Yosef. (1977). Multinomial probit and qualitative choice: A computationally efficient algorithm. *Transportation Science, 11*(4), 338–58.

Ortúzar, J. D., & Willumsen, L. G. (2011). *Modelling transport* (4th ed.). New York: Wiley.

Train, K. (2009). *Discrete choice methods with simulation* (2nd ed.). Cambridge: Cambridge University Press.

Chapter 9
Dealing with Heterogeneity I: The Latent Class Logit Model

Byrdes of on kynde and color flok and flye allwayes together.

— William Turner, The Rescuing of Romish Fox

9.1 Taste Variations in the Population

Chapters 7 and 8 were concerned with substitution patterns, particularly those resulting from the Independence of Irrelevant Alternatives (IIA) property of the logit model. As discussed there, when the specification of the model is incomplete, the presence of residual correlation can often lead to inappropriate—even nonsensical—proportional substitution patterns. Accordingly, the Generalized Extreme Value (GEV) family of models (of which the nested logit is a member) and the multinomial probit model aim at introducing more flexible substitution patterns. The nested logit model achieves this by inducing a hierarchical decision-making structure where alternatives within nests are correlated, whereas the multinomial probit model introduces a flexible, but technically demanding and computationally expensive, covariance structure for multinomial choices.

In this chapter, we discuss another important issue in the analysis of discrete choices, namely taste variation or heterogeneity. Central to decision-making, the theme of taste variation has been present from the beginning of our discussion of discrete choice methods. In simple terms, taste variation refers to the possibility that there are systematic variations in the utility that different segments of a population derive from various alternatives. In other words, individuals with different characteristics might display different preferences. In Chap. 3, we introduced the simplest, most straightforward way to account for this, by stipulating that the utility function could include a vector of attributes Z used to describe the decision-maker

$$Z = [z_1, z_2, \ldots, z_k]$$

A. Páez and G. Boisjoly, *Discrete Choice Analysis with R*, Use R!,
https://doi.org/10.1007/978-3-031-20719-8_9

When the attributes of the decision-maker enter the utility function, we account for potential variations in preferences across different segments of the population, as described by Z. Let us illustrate this with a simple example. Consider a situation with two discrete active travel alternatives, say w for walk and c for cycle

$$\begin{aligned} U_w &= V_w + \epsilon_w \\ U_c &= V_c + \epsilon_c \end{aligned}$$

Now, suppose that the attributes of the alternative are t for travel time and r for physical effort

$$\begin{aligned} X_w &= [t_w, r_w] \\ X_c &= [t_c, r_c] \end{aligned}$$

In addition, two attributes of decision-makers are considered: a for age and I for income

$$Z_i = [a_i, I_i]$$

One way to specify the utility functions in this example is as follows:

$$\begin{aligned} V_{iw} &= \beta_w \ \ +\beta_t t_w + \beta_r r_w \ \ +\beta_a a_i + \beta_I I_i \\ V_{ic} &= (0) \ \ +\beta_t t_c + \beta_r r_c \quad +(0) + (0) \end{aligned}$$

Since the attributes of the decision-maker are constant across alternatives, and the model works on the differences between utilities, they can enter at most in $J-1$ functions (where J is the number of alternatives) to ensure that the parameters can be identified. This is similar to the alternative-specific constant(s).

For the sake of the example, suppose that the coefficients β_a and β_I are positive and negative, respectively: this would indicate that older people have a preference for walking, whereas higher income people have a preference for cycling: these coefficients help to capture taste variations in the population.

A second way to model taste variation within this simple framework is as follows. Let us say that now we introduce the attributes of decision-makers as interactions with the attributes of the alternatives

$$\begin{aligned} V_{iw} &= \beta_w \ \ +\beta_{at} t_w a_i + \beta_{It} t_w I_i + \beta_{ar} r_w a_i + \beta_{Ir} r_w I_i \\ V_{ic} &= (0) \ \ +\beta_{at} t_c a_i + \beta_{Ir} r_c I_i + \beta_{ar} r_c a_i + \beta_{Ir} r_c I_i \end{aligned}$$

If we collect the common terms, we see that the differences in utilities are now

$$V_{iw} - V_{ic} = (\beta_w - 0) + (\beta_{at} a_i + \beta_{It} I_i)(t_w - t_c) + (\beta_{ar} a_i + \beta_{Ir})(r_w - r_c)$$

We can rename the parameters as follows

$$\begin{aligned}\beta_t &= \beta_{at} a_i + \beta_{It} I_i\\ \beta_r &= \beta_{ar} a_i + \beta_{Ir} I_i\end{aligned}$$

and thus show that the responses to the attributes of the alternatives are no longer constant but can vary instead as a function of the attributes of decision-makers. This alternate strategy to model taste variation allows the attributes of the decision-maker to influence the utility, but instead of doing so directly, they model differential responses to *specific alternative attributes*. In other words, the effect of alternative attributes (say physical effort) can vary according to individual characteristics. For example, a negative β_{ar} would indicate that the disutility of physical effort tends to increase with age.

When we entered the decision-maker attributes as standalone variables in the utility, we coded in the model our belief that older people have a preference for walking, considering constant travel time and effort. When using variable interactions, the specification of the model implies instead that the physical effort affects the utility of walking differently as age changes.

In this chapter, we introduce another approach to modeling taste variation that operates on the principle that population segments are unobserved. These unobserved segments are called *latent classes*; for this reason, the model is called the *latent class logit*. Besides being a valuable addition to the discrete choice modeling toolbox, in our view, the latent class logit model provides an excellent introduction to concepts that are central to the development of *mixed logit models*, to be discussed in Chap. 10.

9.2 How to Use This Chapter

Remember that the source code used in this chapter is available. Throughout the notes, you will find examples of code in segments of text called *chunks*. This is an example of a chunk

```r
print("Ignorance is the foundation of absolute power")
```

```
[1] "Ignorance is the foundation of absolute power"
```

You can copy and paste the source code into your `R` or RStudio console, or create a script/notebook to save the code and any experiments you conduct.

9.3 Learning Objectives

In this chapter, you will learn about

1. A framework for modeling taste variations.
2. The latent class logit model.
3. Behavioral insights from the latent class logit model.

9.4 Suggested Readings

- Louviere, J. J., Hensher, & D. A., Swait, J. D. (2000). Stated choice methods: Analysis and application (Chap. 6, pp. 205–206). Cambridge University Press.
- Train, K. (2009). Discrete choice methods with simulation, 2nd edn. (Chap. 6 pp. 135–136). Cambridge University Press.

9.5 Preliminaries

Make sure that you have installed version gmnl_1.1-3.3 (or later) of package {gmnl}

```
remotes::install_github("mauricio1986/gmnl")
```

Load the packages used in this chapter:

```
library(dplyr) # A Grammar of Data Manipulation
library(ggplot2) # Create Elegant Data Visualisations Using the Grammar of Graphics
library(gmnl) # Multinomial Logit Models with Random Parameters
library(gridExtra) # Miscellaneous Functions for "Grid" Graphics
library(kableExtra) # Construct Complex Table with 'kable' and Pipe Syntax
library(mlogit) # Multinomial Logit Models
library(stargazer) # Well-Formatted Regression and Summary Statistics Tables
library(tibble) # Simple Data Frames
library(tidyr) # Tidy Messy Data
```

9.6 An Appetite for Risk?

To motivate the discussion on taste variation, we will use a data set reported in research by Leon and Miguel (2017) on the transportation choices of individuals traveling to and from the airport in Freetown, Sierra Leone. This data set is available in the {mlogit} package

```
data("RiskyTransport",
     package = "mlogit")
```

The airport in Freetown is not on the mainland, and the alternatives for travel to and from are ferry, hovercraft, helicopter, and water taxi. Information about the number of fatalities due to accidents by each of these modes was available and allowed the researchers to calculate the risk of death by mode.

The choice set is not balanced, meaning that not all alternatives were available to all individuals surveyed for the study. This is due to the seasonality or occasional unavailability of some modes. For the example in this chapter, we will extract a subset of observations corresponding to those individuals who had all four alternatives available as part of their choice set. To do this, we identify all respondents with four alternatives:

```
all_available <- RiskyTransport %>%
  group_by(chid) %>%
  summarise(no_rows = length(chid), .groups = 'drop') %>%
  filter(no_rows == 4) %>% select(chid)
```

Next, we do an inner join of those respondents with the full data set. This join will only preserve the rows in the table corresponding to the respondents in `all_available` after we drop all missing values

```
RT <- inner_join(RiskyTransport,
                 all_available,
                 by = "chid") %>%
  drop_na()
```

The two key variables (in addition to the mode chosen) are the generalized cost of the transport mode (`cost`) and the fatality rate in deaths per 100,000 trips (`risk`). The following chunk of code calculates the summary statistics by mode of transportation:

```
df <- RT %>%
  group_by(mode) %>%
  summarize(proportion = sum(choice),
            `min (cost)` = min(cost),
            `mean (cost)` = mean(cost),
            `max(cost)` = max(cost),
            `min (risk)` = min(risk),
            `mean (risk)` = mean(risk),
            `max (risk)` = max(risk),
            .groups = 'drop') %>%
  mutate(proportion = proportion/sum(proportion)) %>%
  column_to_rownames(var = "mode")
```

Table 9.1 Summary statistics: cost and risk in risky transport data set

	Proportion	Min (cost)	Mean (cost)	Max(cost)	Min (risk)	Mean (risk)	Max (risk)
Helicopter	0.003	54.28	155.53	312.98	18.41	18.41	18.41
WaterTaxi	0.40	31.22	95.53	270.95	2.55	2.55	2.55
Ferry	0.46	1.61	63.96	510.84	4.43	4.43	4.43
Hovercraft	0.13	30.78	108.97	282.42	3.88	3.88	3.88

These are tabulated next using {stargazer}:

```
stargazer::stargazer(df,
                     type ="latex",
                     rownames = TRUE,
                     summary = FALSE,
                     digits =2,
                     header = FALSE,
                     label = "tab:descriptive-statistics",
                     title = "Summary Statistics: Cost and Risk in Risky Transport Data Set")
```

We can see from Table 9.1 that the two most popular modes of travel to and from Freetown airport are water taxi and ferry, with hovercraft a distant third. Helicopter is the most expensive mode and the least popular, in addition to being the riskiest mode of travel. Notice that while the cost by mode varies (due to seasonal variations in fare), `risk` is in fact a constant for each mode. For this reason, and to avoid perfect multicollinearity, any models that use this variable cannot include a constant term.

The data set includes information on number of seats, noise, crowdedness, convenience of location, and clientele by mode. Finally, it includes information about the decision-makers, including whether they are African, their declared life expectancy, their declared hourly wage, their imputed hourly wage, their level of education, a self-ranked response on their degree of fatalism, gender, age, whether they have children, and if they know how to swim.

Although the survey includes sample weights, we will ignore them after subsetting the data (note that this limits the generalizability of the analysis).

We add some interactions and non-linear terms to the data set

```
RT <- RT %>%
  mutate(`cost:dwage` = cost * dwage,
         `risk:dwage` = risk * dwage,
         dwage2 = dwage^2)
```

The data set is in “long” form, which means that there is one record per decision-making situation per decision-maker. Therefore, we must format the data frame in preparation for use with {mlogit} and {gmnl}. Package {gmnl} accepts data objects of class `mlogit.data` so we process the data frame to obtain the appropriate data object class

Table 9.2 Estimation results: base multinomial logit model

	Dependent variable
	Choice
Cost	−0.014*** (0.002)
Risk	−0.253*** (0.049)
Observations	320
Log Likelihood	−335.213

Note—*p<0.1; **p<0.05; ***p<0.01

```
RT <- mlogit.data(RT,
                   choice = "choice",
                   shape = "long",
                   alt.var = "mode",
                   id.var = "id",
                   chid.var = "chid")
```

9.7 Latent Class Logit

We begin the discussion by estimating a base model with two alternative-specific attributes, namely cost and risk (see Table 9.2).

The utility function for individual n and mode i is as follows:

$$V_{ni} = \beta_{cost} cost_i + \beta_{risk} risk_i$$

where `risk` depends on the chances of not completing a trip safely.

This model is estimated as follows, suppressing the constants in the second term of the formula:

```
mnl.rt0 <- mlogit(choice ~ cost + risk | 0,
                data = RT)

stargazer::stargazer(mnl.rt0,
                     header = FALSE,
                     single.row = TRUE,
                     title = "Estimation results: Base Multinomial Logit Model")
```

The coefficients for `cost` and `risk` provide useful information. In this case, the typical willingness to pay is as follows:

$$-\frac{\partial cost}{\partial risk} = \frac{\beta_{risk}}{\beta_{cost}}$$

Since risk is related to the probability of not surviving a trip, the ratio of the coefficients is the willingness to pay to reduce the risk of accidental death:

$$-\frac{\beta_{risk}}{\beta_{cost}} = \frac{-0.25309}{-0.01401} \simeq 18.06$$

A question we may ask is whether there are unobserved variations in the behavior. In other words, is it possible that some individuals have a greater tolerance for risk? If so, their willingness to pay might be lower than what is estimated here.

To develop the latent class logit model, we need to redefine the utility functions of the model in the following manner:

$$V_{niq} = \beta_{q,cost} cost_i + \beta_{q,safe} risk_i$$

where q is a subindex to indicate that the coefficients are not fixed for all individuals, but may instead vary by class q. For instance, let us suppose that there are two classes of behaviors in the population, and one class consists of cost-conscious individuals ($q = 1$), whereas the second class is composed of risk-conscious individuals ($q = 2$).

Now the utility functions are

$$\begin{aligned} V_{ni1} &= \beta_{1,cost} cost_i + \beta_{1,risk} risk_i \\ V_{ni2} &= \beta_{2,cost} cost_i + \beta_{2,risk} risk_i \end{aligned}$$

with

$$\begin{gathered} \beta_{1,cost} \neq \beta_{2,cost} \\ \text{and} \\ \beta_{1,risk} \neq \beta_{2,risk} \end{gathered}$$

Making the usual assumption about the random utility (i.e., Extreme Value Type I), the probability that an individual who is cost-conscious will choose alternative i is

$$\frac{e^{V_{ni1}}}{\sum_k e^{V_{nk1}}}$$

and the probability that an individual who is risk-conscious will choose alternative i is

$$\frac{e^{V_{ni2}}}{\sum_k e^{V_{nk2}}}$$

We see that each of these probabilities is a multinomial logit model. If we knew which decision-makers were risk-conscious and which cost-conscious, we could simply split the sample and estimate separate models for each class. The issue, however, is that typically we do not know which decision-makers belong to which of the two classes, and therefore, the classes are only *latent*. A way out of this conundrum is to assume that decision-makers belong with a certain probability to each decision-making class. For example, the probability of any one decision-maker to be cost-conscious is p_1, and the probability of being risk-conscious is p_2. Since the decision-makers must be in one or the other class, $p_1 + p_2 = 1$.

Given these probabilities, it is possible to condition the probability of choosing i on the probability of a decision-maker being in class q

$$\begin{array}{l} P_{ni|1} = p_1 \frac{e^{V_{ni1}}}{\sum_k e^{V_{nk1}}} \\ \text{and} \\ P_{ni|2} = p_2 \frac{e^{V_{ni2}}}{\sum_k e^{V_{nk2}}} \end{array}$$

The class-membership probability can be defined by means of a logit-like expression

$$\begin{array}{c} p_1 = \frac{e^{\gamma_1}}{e^{\gamma_1}+e^{\gamma_2}} \\ \text{and} \\ p_2 = \frac{e^{\gamma_2}}{e^{\gamma_1}+e^{\gamma_2}} \end{array}$$

Notice that

$$p_1 + p_2 = \frac{e^{\gamma_1}}{e^{\gamma_1} + e^{\gamma_2}} + \frac{e^{\gamma_2}}{e^{\gamma_1} + e^{\gamma_2}} = 1$$

We can set $\gamma_1 = 0$ to obtain the following expressions (this is similar to setting one alternative-specific constant in the utility functions to zero)

$$\begin{array}{c} p_1 = \frac{1}{1+e^{\gamma_2}} \\ \text{and} \\ p_2 = \frac{e^{\gamma_2}}{1+e^{\gamma_2}} \end{array}$$

Accordingly, the unconditional probability of the decision-maker choosing alternative i is

$$P_{ni} = p_1 \frac{e^{V_{ni1}}}{\sum_k e^{V_{nk1}}} + p_2 \frac{e^{V_{ni2}}}{\sum_k e^{V_{nk2}}}$$

And since the sum of probabilities is $p_1 + p_2 = 1$, the unconditional probability is essentially the weighted average of the probabilities for each latent class.

For the discussion above, we assumed that $q = 2$. More generally, it is possible to allow for an arbitrary number of groups $q = 1, 2, \cdots, Q$ each with their own distinctive set of coefficients, for Q latent classes. In the general case, the unconditional probability of decision-maker n choosing alternative i is

$$P_{ni} = \sum_q^Q P_{ni|q} = \sum_q^Q p_q \frac{e^{V_{niq}}}{\sum_k e^{V_{nkq}}}$$

and the class-membership probability for latent class q is

$$p_q = \frac{e^{\gamma_q}}{\sum_{z=1}^{Q} e^{\gamma_z}}$$

with $\gamma_1 = 0$.

9.8 Estimation

The latent class logit model is made of logit-type probabilities: the probability of belonging to class q and the probability of choosing alternative j. As we know, these probabilities are the closed-form solution of integrals. We can write the log likelihood function of the latent class logit model in this way

$$l = \sum_n ln\Big[\sum_q p_q(\prod_i P_{ni}^{y_{ni}})\Big]$$

where y_{ni} is an indicator variable that takes the value of one if decision-maker n chose alternative i, and zero otherwise.

The log likelihood function can be maximized using conventional optimization techniques, contingent on the number of latent classes Q, which is an exogenous hyperparameter: in other words, Q is not estimated endogenously but must be provided by the analyst prior to estimating every other parameter in the model.

9.9 Properties of the Latent Class Logit Model

The latent class logit model allows an analyst to investigate variations in taste that can be explained by the existence of unobserved (hence latent) segmentation of the population in Q classes. In addition, it is interesting to highlight two other properties of this model.

The first one can be illustrated by means of the ratio of probabilities of two alternatives, say i and j

$$\frac{P_{ni}}{P_{nj}} = \frac{\sum_q^Q p_q \frac{e^{V_{niq}}}{\sum_k e^{V_{nkq}}}}{\sum_q^Q p_q \frac{e^{V_{njq}}}{\sum_k e^{V_{nkq}}}}$$

It should be clear that when $Q = 1$ (and therefore $p_q = 1$) the model collapses to the multinomial logit model and proportional substitution patterns. On the other hand, when $Q \geq 2$ the denominator of the logit probabilities is inside the summation for the classes, and cannot be canceled out. This can be seen below in the case of $Q = 2$, where it is clear that there are no common factors to allow the denominator to vanish

$$\frac{P_{ni}}{P_{nj}} = \frac{p_1 \frac{e^{V_{ni1}}}{\sum_k e^{V_{nk1}}} + p_2 \frac{e^{V_{ni2}}}{\sum_k e^{V_{nk1}}}}{p_1 \frac{e^{V_{nj1}}}{\sum_k e^{V_{nk1}}} + p_2 \frac{e^{V_{nj2}}}{\sum_k e^{V_{nk1}}}}$$

As a consequence of this, the latent class logit model does not display Independence from Irrelevant Alternatives.

Secondly, latent class logit models with higher number of classes do *not* nest into each other. The reason for this is that the parameters that define the probability of belonging to a class are contingent on the number of classes. Therefore, reducing the number of classes, say from $Q = 3$ to $Q = 2$, is not equivalent to restricting some parameters to zero. Consequently, latent class models cannot be compared by means of the likelihood ratio test. Further, since the log likelihood always improves with the addition of classes, it is not possible to compare the likelihood directly.

Since the tools that we used before for comparing models do not work in this case, different information criteria that account for the size of the models must be used. Roeder et al. (1999) suggest using the Bayesian Information Criterion

$$BIC = k \ln(n) - 2\hat{l}$$

where k is the number of parameters in the model, n is the sample size, and $\hat{l}$ is the maximized value of the likelihood of the model.

Alternatively, Shen (2009) suggests using Akaike's Information Criterion

$$AIC = 2k - 2\hat{l}$$

or a variation in the form of the Consistent Akaike's Information Criterion

$$CAIC = k[ln(n) + 1] - 2\hat{l}$$

The three criteria above use the likelihood of the model and apply a penalty based on the size of the model and possibly the size of the sample. Since the negative of the likelihood is used in the calculations, minimizing an information criteria is an indicator of goodness-of-fit.

In particular, when using AIC, the following decision rule can be applied. Suppose that there are candidate models with $q = 1, 2, \cdots, Q$ latent classes. Calculate the AIC for each model, such that there are $AIC_{q=1}, AIC_{q=2}, \cdots, AIC_{q=Q}$. Denote the minimum AIC as AIC_{min}. The *relative likelihood* is defined as

$$RL = e^{\frac{AIC_{min} - AIC_q}{2}}$$

The relative likelihood is proportional to the probability that the model with q classes minimizes the estimated information loss.

9.10 Empirical Example

Here we revisit the empirical example of appetite for risk. To estimate latent class models, we use the package {gmnl}. This package uses a similar syntax for specifying multi-part formulas as {mlogit}. In particular, the parts of a formula are

$$\begin{aligned} choice \sim \text{ alternative attributes with generic coefficients } | \\ \text{individual attributes } | \\ \text{alternative attributes with specific coefficients } | \\ \text{variables for random coefficients } | \\ \text{variables for latent class model} \end{aligned}$$

We proceed to estimate our first latent class model as follows, with $Q = 2$ (i.e., two latent classes), and for this, we use just a constant for the specification of the class-membership probability

```
lc2 <- gmnl(choice ~ cost + risk | 0 | 0 | 0 | 1,
            data = RT,
            model = 'lc',
            Q = 2,
            panel = TRUE,
            method = "bhhh")
```

```
  Estimating LC model
```

The parts of the model are as follows. The utility functions (with $q = 1, 2$) are

$$V_{niq} = \beta_{q,cost} cost_i + \beta_{q,risk} risk_i$$

and the class-membership functions are

$$\begin{aligned} p_1 &= \frac{1}{1+e^{\gamma_2}} \\ p_2 &= \frac{e^{\gamma_2}}{1+e^{\gamma_2}} \end{aligned}$$

For the next model, we use $Q = 3$ (i.e., three latent classes), with again just constant terms for the class-membership functions

```
lc3 <- gmnl(choice ~ cost + risk | 0 | 0 | 0 | 1,
            data = RT,
            model = 'lc',
            Q = 3,
            panel = TRUE,
            method = "bhhh")
```

```
  Estimating LC model
```

Table 9.3 Base models: multinomial logit (MNL), latent class Q = 2 (LC2), latent class Q = 3 (LC3)

Variable	MNL		LC2		LC3	
	Estimate	p-value	Estimate	p-value	Estimate	p-value
class.1.cost	−0.014	<0.0001	−0.126	0.017	−0.115	0.02
class.1.risk	−0.253	<0.0001	−0.635	0.154	−0.037	0.926
class.2.cost	–	–	0.003	0.336	−0.068	0.004
class.2.risk	–	–	−0.372	<0.0001	−2.415	0.003
(class)2	–	–	0.421	0.011	−0.089	0.705
class.3.cost	–	–	–	–	0.01	0.005
class.3.risk	–	–	–	–	−0.277	0.063
(class)3	–	–	–	–	0.079	0.713

Note
Log Likelihood: MNL = −335.213; Latent Class (Q = 2) = −310.677; Latent Class (Q = 3) = −299.196

The utility functions are as before but with $q = 1, 2, 3$, and the class-membership functions are

$$p_1 = \frac{1}{1+e^{\gamma_2}+e^{\gamma_3}}$$
$$p_2 = \frac{e^{\gamma_2}}{1+e^{\gamma_2}+e^{\gamma_3}}$$
$$p_3 = \frac{e^{\gamma_3}}{1+e^{\gamma_2}+e^{\gamma_3}}$$

The results of the models we have estimated so far are summarized in the Table 9.3. Notice how model LC3, improves the likelihood, but returns non-significant parameters for the latent class model (what does this mean?).

Next, we proceed to compare the multinomial logit model and LC2. The AIC of the base multinomial logit model is

```
2 * length(coef(mnl.rt0)) - 2 * mnl.rt0$logLik
```

```
'log Lik.' 674.426 (df=2)
```

The AIC of model LC2 is

```
2 * length(coef(lc2)) - 2 * lc2$logLik$maximum
```

```
[1] 631.3531
attr(,"df")
[1] 5
```

The AIC of model LC3 is

```
2 * length(coef(lc3)) - 2 * lc3$logLik$maximum
```

```
[1] 614.3928
attr(,"df")
[1] 8
```

The minimum AIC is for latent class model 3. If we calculate the relative likelihood with respect to the multinomial logit moel

```
as.numeric(exp(((2 * length(coef(lc3)) - 2 * lc3$logLik$maximum) -
                  (2 * length(coef(mnl.rt0)) - 2 * mnl.rt0$logLik))/2))
```

```
[1] 9.203835e-14
```

Therefore, the standard multinomial logit model is less than 0.0001 times as likely as the latent class model with $Q = 3$ to minimize the information loss. In other words, it is very unlikely to be as informative as the latent class model.

Let us examine our latent class model with $Q = 3$. We can see that in class 1, the coefficient for cost is negative and significant but that for risk is not significant. In class 2, both coefficients (cost and risk) are negative and significant. Finally, class 3 has a positive and significant coefficient for cost, and a negative and significant coefficient for risk.

Based on these results, we can try to "profile" or give labels to the three latent classes: the first latent class seems to be more cost-averse than risk-averse (call it the "cheap and reckless"). The second latent class is both cost- and risk-averse (call it "expensive is bad but so is death"). The third class is risk-averse but not cost-averse (call this class "status seeking but safe"). These labels are obviously tongue-in-cheek, but offer a useful way to think about the different classes. A fun aspect of estimating latent class logit models is to create descriptive labels for the different latent classes.

9.10.1 Behavioral Insights

We can now return to our original question regarding the willingness to pay to reduce the risk of death. If we use model LC3, we have three expressions as follows (with $q = 1, 2, 3$):

$$-\frac{\partial cost}{\partial risk} = \frac{\beta_{q,risk}}{\beta_{q,cost}}$$

The willingness to pay of a person in the cost-averse latent class is willing to pay relatively little to reduce the risk of death

```
as.numeric(lc3$coefficients[2]) / as.numeric(lc3$coefficients[1])
```

```
[1] 0.317219
```

In contrast, the willingness to pay of a person in the risk-averse class is much higher

```
as.numeric(lc3$coefficients[4]) / as.numeric(lc3$coefficients[3])
```

```
[1] 35.60782
```

Finally, the willingness to pay of a person who is a member of class 3 is (what does the negative sign mean?)

```
as.numeric(lc3$coefficients[6]) / as.numeric(lc3$coefficients[5])
```

```
[1] -28.37432
```

We must remember here that class-membership is probabilistic, and for this reason, we do not know who belongs to what class with full certainty. The estimated willingness to pay is the weighted average of the willingness to pay of all classes. Examining the class-membership parameters of LC3, we see that they are not significantly different from γ_1 which we normalized to zero, then:

$$\begin{aligned} p_1 &= \frac{1}{1+1+1} = \frac{1}{3} \\ p_2 &= \frac{1}{1+1+1} = \frac{1}{3} \\ p_3 &= \frac{1}{1+1+1} = \frac{1}{3} \end{aligned}$$

This means that membership in any of the three classes is equally likely. The estimated willingness to pay then is

$$p_1 \cdot \frac{\beta_{1,risk}}{\beta_{1,cost}} + p_2 \cdot \frac{\beta_{2,risk}}{\beta_{2,cost}} + p_3 \cdot \frac{\beta_{3,risk}}{\beta_{3,cost}}$$

or:

```
1/3 * as.numeric(lc3$coefficients[2]) / as.numeric(lc3$coefficients[1]) +
  1/3 * as.numeric(lc3$coefficients[4]) / as.numeric(lc3$coefficients[3]) +
  1/3 * as.numeric(lc3$coefficients[6]) / as.numeric(lc3$coefficients[5])
```

```
[1] 2.516907
```

9.11 Adding Individual-Level attributes

The models discussed in the preceding section did not include any attributes of decision-makers. A natural question that emerges at this point is whether the inclusion of individual-level attributes can capture the variations in taste that were modeled by means of the latent classes. As discussed at the beginning of the chapter, it is possible to capture at least *some* variations in taste by introducing decision-maker-level attributes in the utility functions.

Here, we revisit our models after adding some relevant attributes. To do this, we estimate two multinomial logit models using different specification strategies. For the first model, we introduce the decision-maker attributes in all but one utility function (MNL-COV). As per our previous discussion, this implies that the attributes of the decision-maker directly influence the utility. The second strategy is to interact the attributes of decision-makers with the alternative-level attributes. This is equivalent to expanding the parameters of the alternative attributes, which means that the characteristics of the decision-makers influence their responses to specific aspects of the alternatives. The relevant utility functions then are as follows:

$$V_{ni} = (b_{cost} + b_{cost:dwage} dwage_n) cost_i + (b_{risk} + b_{risk:dwage} dwage_n) risk_i$$

These models are estimated next

```
mnl.cov <- mlogit(choice ~ cost + risk | dwage + 0,
                  data = RT)

mnl.exp <- mlogit(choice ~ cost + cost:dwage +
                     risk + risk:dwage | 0,
                  data = RT)
```

The models are summarized in Table 9.4.

Table 9.4 Models: multinomial logit with covariates (MNL-COV) and multinomial logit with expanded coefficients (MNL-EXP)

Variable	MNL-COV		MNL-EXP	
	Estimate	p-value	Estimate	p-value
Cost	−0.012	<0.0001	−0.016	<0.0001
Risk	−0.166	0.019	−0.143	0.025
dwage:WaterTaxi	0.059	0.361	–	–
dwage:Ferry	0.06	0.347	–	–
dwage:Hovercraft	0.045	0.485	–	–
cost:dwage	–	–	0	0.001
risk:dwage	–	–	−0.004	0.025

Note
Log Likelihood: MNL-COV = −323.807; MNL-EXP = −329.728

The AIC of the MNL-COV is

```
as.numeric(2 * length(coef(mnl.cov)) - 2 * mnl.cov$logLik)
```

```
[1] 657.6149
```

and the AIC of MNL-EXP is

```
as.numeric(2 * length(coef(mnl.exp)) - 2 * mnl.exp$logLik)
```

```
[1] 667.4561
```

The minimum AIC is still for latent class model LC3 (with $Q = 3$). We can calculate the relative likelihood with respect to the best performing multinomial logit model

```
as.numeric(exp(((2 * length(coef(lc3)) - 2 * lc3$logLik$maximum) -
                  (2 * length(coef(mnl.cov)) - 2 * mnl.cov$logLik))/2))
```

```
[1] 4.115802e-10
```

Clearly, the latent class model is still the best candidate for fit.

9.12 Adding Variables to the Latent Class-Membership Model

The latent class model can accommodate individual-level attributes in the class-membership model. The model thus becomes

$$p_q = \frac{e^{\gamma_q' x_i}}{\sum_{z=1}^{Q} e^{\gamma_z' x_i}}$$

where now γ_q is a vector of size $1 \times h$ and x_1 is a vector of h individual-level attributes. As before, $\gamma_1 = 0$.

To estimate a model with variables in the selection model, the fifth part of the formula is used, as shown next (call this model LC2-COV)

```
lc2.cov <- gmnl(choice ~ cost + risk | 0 | 0 | 0 | dwage,
           data = RT,
           model = 'lc',
           Q = 2,
           panel = TRUE,
           method = "nm",
           iterlim = 1200)
```

```
Estimating LC model
```

```
summary(lc2.cov)
```

```

Model estimated on: Mon Jul 25 6:38:03 PM 2022

Call:
gmnl(formula = choice ~ cost + risk | 0 | 0 | 0 | dwage, data = RT,
    model = "lc", Q = 2, panel = TRUE, method = "nm", iterlim = 1200)

Frequencies of categories:

Helicopter  WaterTaxi      Ferry Hovercraft
  0.003125   0.403125   0.459375   0.134375

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

Coefficients:
               Estimate Std. Error z-value  Pr(>|z|)
class.1.cost -0.1159165  0.0333901 -3.4716 0.0005174 ***
class.1.risk  0.0861966  0.1056785  0.8156 0.4147008
class.2.cost  0.0018230  0.0031749  0.5742 0.5658431
class.2.risk -0.6166901  0.0994760 -6.1994 5.668e-10 ***
(class)2     -0.3362670  0.2237047 -1.5032 0.1327943
dwage:class2  0.0317098  0.0080091  3.9592 7.519e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Optimization of log likelihood by Nelder--Mead maximization
Log Likelihood: -302.22
Number of observations: 320
Number of iterations: 1149
Exit of MLE: successful convergence
```

The AIC of this model is

```
as.numeric(2 * length(coef(lc2.cov)) - 2 * lc2.cov$logLik$maximum)
```

```
[1] 616.4306
```

Let us compare this against our best candidate so far, namely model LC3, which had an AIC of

```
as.numeric(2 * length(coef(lc3)) - 2 * lc3$logLik$maximum)
```

```
[1] 614.3928
```

These values are very similar. Turning to the relative likelihood to compare the two models

```
as.numeric(exp(((2 * length(coef(lc3)) - 2 * lc3$logLik$maximum) -
                      (2 * length(coef(lc2.cov)) - 2 * lc2.cov$logLik$maximum))/2))
```

```
[1] 0.3609982
```

we can see that model LC2-COV has about a 37% probability of capturing at least as much information as LC3. This model is interesting in that it suggests that the variable dwage significantly helps to explain the latent classes.

This model points at two classes of decision-makers, cost-averse and risk-averse. In addition, the probability of membership in those classes changes as a function of declared wage. The class-membership model is

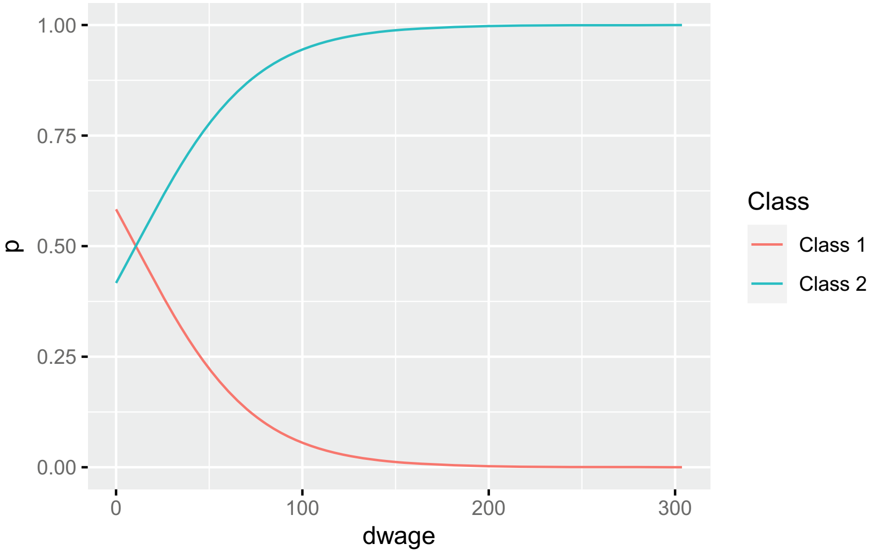
$$\begin{aligned} p_1 &= \frac{1}{1+e^{-0.3363+0.0317dwage}} \\ p_2 = 1 - p_1 &= \frac{e^{-0.3363+0.0317dwage}}{1+e^{-0.3363+0.0317dwage}} \end{aligned}$$

Clearly, as the declared wage increases, the probability of belonging to class 1 (cost-averse) tends to decline. To explore the effect of declared wage on the membership probability, we create a data frame for plotting

```
#Create a data frame for plotting:
df <- data.frame(dwage = seq(min(RT$dwage),
                             to = max(RT$dwage),
                             by = (max(RT$dwage) - min(RT$dwage))/100))

# Use the class-membership model to calculate the membership probabilities
df <- df %>%
  mutate(p_1 = 1 -
           exp(coef(lc2.cov)["(class)2"] + coef(lc2.cov)["dwage:class2"] * dwage)/
           (1 + exp(coef(lc2.cov)["(class)2"] + coef(lc2.cov)["dwage:class2"] * dwage)),
         p_2 = exp(coef(lc2.cov)["(class)2"] + coef(lc2.cov)["dwage:class2"] * dwage)/
           (1 + exp(coef(lc2.cov)["(class)2"] + coef(lc2.cov)["dwage:class2"] * dwage))) %>%
  # Pivot longer to put all probabilities in a single column, and label by class
  pivot_longer(cols = -dwage,
               names_to = "Class",
               values_to = "p") %>%
  mutate(Class = ifelse(Class == "p_1",
                        "Class 1",
                        "Class 2"))
```

The data frame is used to plot the class-membership probability by level of wage

```
# Plot
ggplot(df, aes(x = dwage)) +
  geom_line(aes(x = dwage,
                y = p,
                color = Class))
```



The plot illustrates how individuals with lower declared wages have a higher probability of being in class 1, which is cost-averse. This is sensible. We need to keep in mind, though, that people with lower incomes may not necessarily have an appetite for risk, but rather are forced into more risky behaviors due to limited means. The probability of being in class 1 tends to decline quite rapidly as declared wage increases, and overall most members of the sample tend to be risk-averse.

The examples above show how the latent class logit modeling framework can provide interesting insights about taste variations in the population—and do so while potentially giving a better statistical fit compared to other modeling alternatives.

9.13 Exercises

1. What does it mean if all parameters in the latent class probabilities are non-significant?
2. Use the model LC3 in this chapter to calculate the shares of the sample in each of the three latent classes.
3. Use model LC2-COV in this chapter and calculate the willingness to pay for reducing the risk of accidental death for each latent class. How do these values compare to the WTP we estimated using the base multinomial logit model at the beginning of the chapter (i.e., `mnl.rt0`)? Discuss.
4. Load the following data set from the `AER` package:

```
data("TravelMode", package = "AER")
```

Estimate a latent class logit model using this data set. Justify your modeling decisions, including the number of classes to use and the use of covariates.

References

León, G., & Miguel, E. (2017). Risky transportation choices and the value of a statistical life. *American Economic Journal: Applied Economics, 9*(1), 202–28.

Louviere, J. J., Hensher, D. A., & Swait, J. D. (2000). *Stated choice methods: Analysis and applications*. Cambridge: Cambridge University Press.

Roeder, Kathryn, Lynch, Kevin G., & Nagin, Daniel S. (1999). Modeling uncertainty in latent class membership: A case study in criminology. *Journal of the American Statistical Association, 94*(447), 766–76.

Shen, Junyi. (2009). Latent class model or mixed logit model? A comparison by transport mode choice data. *Applied Economics, 41*(22), 2915–24.

Train, K. (2009). *Discrete choice methods with simulation* (2nd ed.). Cambridge: Cambridge University Press.

Chapter 10
Dealing with Heterogeneity II: The Mixed Logit Model

Life produces a different taste each time you take it.
— Frank Herbert
Develop flexibility and you will be firm; cultivate yielding and you will be strong.

— Liezi, The Book of Master Lie

10.1 More on Taste Variation

Chapter 9 introduced the latent class logit model, a technique useful to model taste variations in a sample. In this chapter, a variation on the theme will be introduced, namely the mixed logit model. We will see how the mixed logit model is related to the latent class logit model: the key difference is how the latent segments are conceptualized.

To illustrate the mixed logit model, we will use a data set from the package {AER} (Applied Econometrics with `R`) that deals with the mode of transportation used by travelers traveling between Sydney and Melbourne. We begin by loading the packages used in this chapter

```
library(AER) # Applied Econometrics with R
library(dplyr) # A Grammar of Data Manipulation
library(ggplot2) # Create Elegant Data Visualisations Using the Grammar of Graphics
library(gmnl) # Multinomial Logit Models with Random Parameters
library(gridExtra) # Miscellaneous Functions for "Grid" Graphics
library(kableExtra) # Construct Complex Table with 'kable' and Pipe Syntax
library(mlogit) # Multinomial Logit Models
library(stargazer) # Well-Formatted Regression and Summary Statistics Tables
library(tibble) # Simple Data Frames
library(tidyr) # Tidy Messy Data
```

A. Páez and G. Boisjoly, *Discrete Choice Analysis with R*, Use R!,
https://doi.org/10.1007/978-3-031-20719-8_10

The data set is called `TravelMode`

```
data("TravelMode",
     package = "AER")
```

This data set is a companion to Greene's popular econometrics textbook (Greene 2003). The data include information about mode choices for individual travelers, namely travel by air, train, bus, or car. The following attributes about the alternatives are available:

1. `vcost`: direct cost associated with the mode-vehicle cost (in $).
2. `travel`: travel time in vehicle (in min).
3. `wait`: waiting time (in min).
4. `gcost`: a generalized cost measure.

The median of the vehicle cost, in-vehicle travel time and waiting time, as well as the proportion of choices by mode, are summarized in the following table:

```
Proportion <- TravelMode %>%
  filter(choice == "yes") %>%
  select(mode) %>%
  group_by(mode) %>%
  summarise(no_rows = length(mode))

# Calculate the median of the variables
df <- TravelMode %>%
  group_by(mode) %>%
  summarize(vcost = median(vcost),
            wait = median(wait),
            travel = median(travel))

# Calculate proportions
df$Proportion <- Proportion$no_rows/(nrow(TravelMode)/4)

df %>%
  kable(digits = 3) %>%
  kable_styling()
```

mode	vcost	wait	travel	Proportion
air	81	64	124.5	0.276
train	42	34	607.5	0.300
bus	32	35	601.5	0.143
car	16	0	577.0	0.281

In addition to the attributes of the modes of transportation, there are two individual-level attributes, the income of the traveler (in 10,000s) and the size of the traveling party (number of people in party).

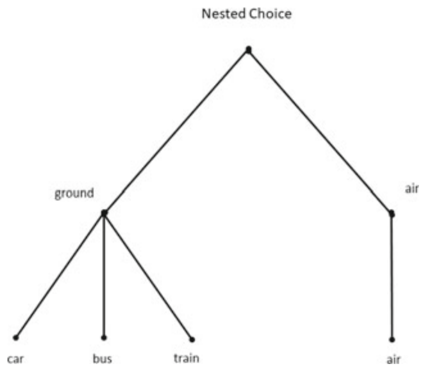


Fig. 10.1 Nests in the nested logit model

The data set is in "long" form, which means that there is one record per decision-making situation per decision-maker. Here, we format the data frame in preparation for use with the packages `mlogit` and `gmnl`

```
TM <- mlogit.data(TravelMode,
                  choice = "choice",
                  shape = "long",
                  alt.levels = c("air",
                                 "train",
                                 "bus",
                                 "car"))
```

For the sake of comparison, we will estimate multinomial logit, nested logit, and multinomial probit models for the example. The code for estimating each of these models is as follows:

```
#Multinomial logit
mnl0 <- mlogit(choice ~ vcost + travel + wait | 1,
             data = TM)

#Nested logit:
nl <- mlogit(choice ~ vcost + travel + wait | 1,
             data = TM,
             nests = list(land = c( "car",
                                    "bus",
                                    "train"),
                          air = c("air")),
             un.nest.el = TRUE)

#Multinomial probit:
prbt <- mlogit(choice ~ vcost + travel + wait | 1,
             data = TM,
             probit = TRUE)
```

Notice that the nesting structure places all land-based modes in a nest and air in a separate nest (see Fig. 10.1; you can try different nesting structures if you wish).

The results of these models are summarized in Table 10.1.

```
# Estimate a constants only model to calculate McFadden's _adjusted_ rho2
mnl_null_lo = -283.83

mnl0.summary <- rownames_to_column(data.frame(summary(mnl0)$CoefTable),
                                   "Variable") %>%
  transmute(Variable,
            Estimate,
            pval = `Pr...z..`)

nl.summary <- rownames_to_column(data.frame(summary(nl)$CoefTable),
                                 "Variable") %>%
  transmute(Variable,
            Estimate,
            pval = `Pr...z..`)

prbt.summary <- rownames_to_column(data.frame(summary(prbt)$CoefTable),
                                   "Variable") %>%
  transmute(Variable,
            Estimate,
            pval = `Pr...z..`)

# Join summary tables
df_logit <- mnl0.summary %>%
  full_join(nl.summary,
            by = "Variable") %>%
  full_join(prbt.summary,
            by = "Variable")

kable(df_logit,
      "latex",
      digits = 4,
      col.names = c("Variable",
                    "Estimate",
                    "p-value",
                    "Estimate",
                    "p-value",
                    "Estimate",
                    "p-value"),
      caption = "\\label{tab:base-models}Base models:
      multinomial logit (MNL), nested logit (NL), multinomial probit (MNP)") %>%
  kable_styling() %>%
  add_header_above(c(" " = 1, "MNL" = 2, "NL" = 2, "MNP" = 2)) %>%
  footnote(general = c(paste0("Log-Likelihood: MNL = ",
                              round(mnl0$logLik[1], digits = 3),
                              "; NL = ",
                              round(nl$logLik[1], digits = 3),
                              "; MNP = ",
                              round(prbt$logLik[1], digits = 3)),
                       paste0("McFadden Adjusted R^2: MNL = ",
                              round(1 - (mnl0$logLik[1] - nrow(mnl0.summary)) /
                                      mnl_null_lo,
                                    digits = 3),
                              "; NL = ",
                              round(1 - (nl$logLik[1] - nrow(nl.summary)) /
                                      mnl_null_lo,
                                    digits = 3),
                              "; MNP = ",
                              round(1 - (prbt$logLik[1] - nrow(prbt.summary)) /
                                      mnl_null_lo,
                                    digits = 3))))
```

Table 10.1 Base models: multinomial logit (MNL), nested logit (NL), multinomial probit (MNP)

	MNL		NL		MNP	
Variable	Estimate	p-value	Estimate	p-value	Estimate	p-value
(Intercept):train	−0.7867	0.1917	0.5679	0.1106	0.3759	0.1614
(Intercept):bus	−1.4336	0.0352	0.1988	0.6538	0.2616	0.2837
(Intercept):car	−4.7399	0.0000	−1.8571	0.0084	−0.7027	0.0454
vcost	−0.0139	0.0365	−0.0106	0.0227	−0.0070	0.0090
travel	−0.0040	0.0000	−0.0036	0.0000	−0.0018	0.0015
wait	−0.0969	0.0000	−0.0554	0.0000	−0.0222	0.0000
iv	NA	NA	0.4655	0.0000	NA	NA
train.bus	NA	NA	NA	NA	0.8199	0.0001
train.car	NA	NA	NA	NA	0.8734	0.0000
bus.bus	NA	NA	NA	NA	0.3296	0.1037
bus.car	NA	NA	NA	NA	0.2652	0.3918
car.car	NA	NA	NA	NA	0.3895	0.0168

Note
Log Likelihood: MNL = −192.889; NL = −187.029; MNP = −192.562
McFadden Adjusted R^2: MNL = 0.299; NL = 0.316; MNP = 0.283

The nested logit model and the multinomial probit model, as discussed in the preceding chapters, accommodate correlations among alternatives. You will recall that the coefficient of the inclusive value of the nested logit is between 0 and 1, and it is significantly different from 1 (when the coefficient of the inclusive value is close to one, the model collapses to the multinomial logit model). The z-score to test whether it is different from one is

```
(nl$coefficients["iv"] - 1) / sqrt(vcov(nl)["iv","iv"])
```

```
       iv
-5.370013
```

With $p < 0.0001$, the $z-$score indicates that we can comfortably reject the null hypothesis that $\lambda = 1$. The multinomial probit model also has covariance components that are significant, although, as discussed in Chap. 8, these covariances lack a substantive interpretation.

Based on the results of the models, we can obtain some behavioral insights. Recall that the ratio of two coefficients represents the *willingness to pay*. If we use the coefficients of the nested logit model (which seems to be the best model in the example), we can see that the willingness to pay is 0.27 dollars per minute of travel time saved and 3.99 dollars per minute of waiting time saved

$$-\frac{\partial \text{vcost}}{\partial \text{travel}} = \frac{\beta_{travel}}{\beta_{vcost}} = \frac{-0.0037}{-0.0139} \simeq 0.27$$

$$-\frac{\partial \text{vcost}}{\partial \text{wait}} = \frac{\beta_{wait}}{\beta_{vcost}} = \frac{-0.0554}{-0.0139} \simeq 3.99$$

From this, we can see that travelers dislike time spent waiting more than time spent traveling, and would be willing to pay more on a per-minute basis to reduce waiting time than travel time.

A question that remains outstanding with these models is whether there are taste variations in the population. For instance, we might wonder whether all decision-makers dislike travel time (or waiting time) identically, or whether there is heterogeneity (i.e., variability) in the way they respond to these attributes of the alternatives. Are people with higher incomes more impatient and value their time more highly? Are people traveling with larger parties having more fun, and therefore, more tolerant of longer waits?

A straightforward way to answer these questions would be to incorporate covariates for the individual attributes that we think are useful/relevant to explain variations in taste. But, as with the unexplained correlations that contradict proportional substitution patterns, in empirical applications, it is also possible that unexplained heterogeneity remains even after introducing available covariates.

The objective of this chapter is to introduce a method, based on the logit model, to account for unobserved heterogeneity. The model is variously called *mixed logit*, *kernel logit*, and *logit with error components or random coefficients*. These labels result from the way in which a mix of distributions are used in the model to produce either error components or random coefficients. These distributions wrap around the kernel of a multinomial logit model. Technically, the multinomial probit model can deal with the issue of variations in taste, but as we have seen, this is technically demanding and the interpretation of the results is complicated too. The mixed logit presents similar computational challenges as the multinomial probit, since the distributions for the mixture require numerical integration. On the other hand, the results are often easier to interpret from a behavioral standpoint.

10.2 How to Use This Note

Remember that the source code used in this chapter is available. Throughout the notes, you will find examples of code in segments of text called *chunks*. This is an example of a chunk

```
print("A piece of advice: never punch a shark in the mouth")
```

```
[1] "A piece of advice: never punch a shark in the mouth"
```

You can copy and paste the source code into your R or RStudio console, or create a script/notebook to save the code and any experiments you conduct.

10.3 Learning Objectives

In this chapter, you will learn about:

1. Taste heterogeneity.
2. Random coefficients.
3. The mixed logit model.
4. Behavioral insights from random coefficients.

10.4 Suggested Readings

- Ben-Akiva, M., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and applications to travel demand* (Chap. 5, pp. 124–125). MIT Press.
- Hensher, D. A., Rose, J. M., & Greene, W. H (2005). *Applied choice analysis: A primer* (Chaps. 15 and 16). Cambridge University Press.
- Louviere, J. J., Hensher, D. A., & Swait, J. D. (2000). *Stated choice Methods: analysis and application* (Chap. 6, pp. 199–205). Cambridge University Press.
- Ortuzar, J. D., & Willumsen, L. G. (2011). *Modelling transport*, 4th edn (Chap. 7, pp. 250–256). Wiley.
- Train, K. (2009). *Discrete choice methods with simulation*, 2nd edn. (Chapter 6 and Chap. 11). Cambridge University Press.

10.5 Mixed Logit

To introduce the mixed logit model, we will begin with a simple situation with two alternatives—say, air travel (A) and ground travel (G). One way to investigate taste variations in the population is to allow each decision-maker to have their own coefficient for at least some variables. This is illustrated by means of the following utility functions (notice that the utilities have alternative-related attributes only):

$$
\begin{aligned}
U_{n,A} &= \theta_A + \alpha_{nc}\text{cost}_A + \alpha_{ns}\text{speed}_A + \mu_{nd}\text{duration}_A + \mu_{nt}\text{time}_A + \epsilon_{n,A}\\
U_{n,G} &= + \alpha_{nc}\text{cost}_G + \alpha_{ns}\text{speed}_G + \mu_{nd}\text{duration}_G + \mu_{nt}\text{time}_G + \epsilon_{n,G}
\end{aligned}
$$

As seen above, the coefficients are *specific to decision-maker n*. Clearly, estimating a coefficient for *each* decision-maker is not possible due to the incidental parameter problem: the number of parameters approaches the size of the sample. What we can do, instead, is to define the parameters in some parsimonious way that requires fewer than N coefficients (i.e., the number of decision-makers in the sample).

To do this, suppose that we select some coefficients for expansion in the following manner:

$$\alpha_{nc} = a_c + \eta_{nc}$$
$$\alpha_{ns} = a_s + \eta_{ns}$$

where the terms a_c and a_s are fixed coefficients, and the terms η_{nc} and η_{ns} are random terms drawn from, say, a bivariate random normal distribution:

$$\eta \sim N(0, \Sigma_\eta)$$

The parameters are now specific to the decision-maker, and depend on only a small number of parameters, i.e., a_c, a_s, and the parameters used to specify covariance matrix Σ, which in the case of two alternatives involves at most three parameters, since

$$\Sigma_\eta = \begin{bmatrix} \sigma_{\eta,cc} & \sigma_{\eta,sc} \\ \sigma_{\eta,cs} & \sigma_{\eta,ss} \end{bmatrix}$$

Next, suppose that the terms μ_{nd} and μ_{nt} are random terms drawn from, say again, a bivariate random normal distribution

$$\mu \sim N(0, \Sigma_\mu)$$

with

$$\Sigma_\mu = \begin{bmatrix} \sigma_{\mu,dd} & \sigma_{\mu,td} \\ \sigma_{\mu,dt} & \sigma_{\mu,tt} \end{bmatrix}$$

Finally, assume that the terms ϵ follow the Extreme Value Type I distribution.

There are a few situations that can be derived from this setup. We discuss them next.

First, imposing the largest number of restrictions to the parameters, we have $\eta_{nc} = \eta_{ns} = \mu_{nd} = \mu_{nt} = 0$. In this case, the utility functions become

$$\begin{aligned} U_{n,A} &= \theta_A + a_c\text{cost}_A + a_s\text{speed}_A + \epsilon_{n,A} \\ U_{n,G} &= \quad\;\; + a_c\text{cost}_G + a_s\text{speed}_G + \epsilon_{n,G} \end{aligned}$$

and clearly the model collapses into the logit model.

Next, we could restrict only some of the parameters, instead of all. If we stipulate that $\mu_{nd} = \mu_{nt} = 0$, the utility functions become

$$\begin{aligned} U_{n,A} &= \theta_A + \alpha_{nc}\text{cost}_A + \alpha_{ns}\text{speed}_A + \epsilon_{n,A} \\ U_{n,G} &= \quad\;\; + \alpha_{nc}\text{cost}_G + \alpha_{ns}\text{speed}_G + \epsilon_{n,G} \end{aligned}$$

This is called a model with *random coefficients*.

Alternatively, we could restrict parameters so that $\eta_{nc} = \eta_{ns} = 0$, in which case the utility functions become

$$\begin{aligned} U_{n,A} &= \theta_A \ +a_c\text{cost}_A + a_s\text{speed}_A \ +\mu_{nd}\text{duration}_A + \mu_{nt}\text{time}_A \ +\epsilon_{n,A} \\ U_{n,G} &= \quad\ \ +a_c\text{cost}_G + a_s\text{speed}_G \ +\mu_{nd}\text{duration}_G + \mu_{nt}\text{time}_G \ +\epsilon_{n,G} \end{aligned}$$

The result is the *error components model.* In fact, the error components and the random coefficients are equivalent depending on whether $a_c = a_s = 0$ or conversely, whether constants are added to the error components.

More generally, we can express the utility of alternative i for decision-maker n as follows:

$$U_{ni} = \theta' w_{ni} + \alpha_n' x_{ni} + \mu_n' z_{ni} + \epsilon_{ni}$$

where θ is a vector of fixed parameters associated with attributes w_{ni}, α_n is a vector of random coefficients associated with attributes x_{ni}, and μ_n is a vector of error components associated with attributes z_{ni}. As we saw above, there are several ways of specifying the coefficients to give error components or random coefficients. These terms, in turn, depend on the parsimonious use of random distributions.

For simplicity, then, let us say that the utility is written as follows:

$$U_{ni} = \beta_n' x_{ni} + \epsilon_{ni} = \beta(\sigma)' x_{ni} + \epsilon_{ni} = V_{ni} + \epsilon_{ni}$$

where σ is the collection of parameters that define the distributions used in the specification of the utility function. This means that, following the usual approach, we can obtain the following model for the probability of choosing alternative j:

$$P_{ni} = \frac{e^{V_{ni}}}{\sum_k e^{V_{nk}}} = \frac{e^{\beta_n' x_{ni}}}{\sum_k e^{\beta_n' x_{nk}}} = \frac{e^{\beta(\sigma)' x_{ni}}}{\sum_k e^{\beta(\sigma)' x_{nk}}}$$

Clearly, when there are no random coefficients or error terms, $\beta(\sigma) = \beta$, which is a set of fixed coefficients, and the model is simply the multinomial logit, which can be estimated using the usual approach.

On the other hand, when we add a mixture or mixtures of distributions to obtain the mixed logit (with random coefficients and/or error components), we typically do not know a priori the parameters of the distributions, and therefore, the probability needs to be calculated as a weighted average of the possible values of β_n

$$P_{ni} = \int \frac{e^{\beta_n' x_{ni}}}{\sum_k e^{\beta_n' x_{nk}}} \phi(\beta|\sigma) d\beta = \int \frac{e^{\beta(\sigma)' x_{ni}}}{\sum_k e^{\beta(\sigma)' x_{nk}}} \phi(\beta|\sigma) d\beta$$

where the integral provides the weighted average given a mixing distribution ϕ.

Compare the mixed logit probability above to the latent class logit model probability of Chap. 9

$$P_{ni} = \sum_{q}^{Q} P_{ni|q} = \sum_{q}^{Q} p_q \frac{e^{V_{niq}}}{\sum_k e^{V_{nkq}}} = \sum_{q}^{Q} p_q \frac{e^{\beta_q x_{ni}}}{\sum_k e^{\beta_q x_{nk}}}$$

When discussing the latent class logit model, we said that the probability was a weighted average; this is the same for the mixed logit. In fact, we can think of the latent class logit model as a mixed logit where the mixing distribution is discrete: the integral becomes the summation of a discrete number of Q latent class probabilities p_q.

Just as was the case with the latent class logit model, a consequence of mixing distributions is that the mixed logit, despite having a logit probability as a kernel, does not display proportional substitution. First, recall that the ratio of probabilities of the multinomial logit model was

$$\frac{P_{ni}}{P_{nj}} = \frac{\frac{e^{V_{ni}}}{\sum_k e^{V_{nk}}}}{\frac{e^{V_{nj}}}{\sum_k e^{V_{nk}}}} = \frac{e^{V_{ni}}}{e^{V_{nj}}}$$

The denominator of the logit probabilities $\sum_k e^{V_{nk}}$ cancels out, which is a manifestation of the Independence of Irrelevant Alternatives property: the ratio of probabilities of a given pair of alternatives does not depend on the utilities/attributes of any other alternatives.

However, in the case of the mixed logit, the denominators of the probabilities are inside the integral and do not cancel each other

$$\frac{P_{ni}}{P_{nj}} = \frac{\int \frac{e^{V_{ni}}}{\sum_k e^{V_{nk}}} \phi(\beta|\sigma) d\beta}{\int \frac{e^{V_{nj}}}{\sum_k e^{V_{nk}}} \phi(\beta|\sigma) d\beta}$$

As a consequence, the ratio of probabilities depends on all alternatives, and not only alternatives i and j.

10.6 Estimation

Very much like the multinomial probit model, estimation of the mixed logit model is based on simulation techniques, more concretely Simulated Maximum Likelihood.

In general, the likelihood function is defined as (see Chap. 5)

$$L = \prod_{n=1}^{N} \prod_{j=1}^{J} P_{nj}^{y_{nj}}$$

As seen before, however, the probabilities, must be integrated

$$P_{ni} = \int \frac{e^{\beta(\sigma)' x_{ni}}}{\sum_k e^{\beta(\sigma)' x_{nk}}} \phi(\beta|\sigma) d\beta$$

The integrals are approximated using a simulation approach. In sketch form, the procedure is to obtain β conditional on a random draw of σ. This gives β^r. After $r = 1, ..., R$ random draws, the approximation of the integral (that is, the simulated probability) is:

$$\check{P}_{ni} = \frac{1}{R} \sum_{r=1}^{R} \frac{e^{(\beta^r)' x_{ni}}}{\sum_k e^{(\beta^r)' x_{nk}}}$$

The simulated *log likelihood* is then given by

$$l_{sim} = \sum_{n=1}^{N} \sum_{j=1}^{J} y_{nj} log(\check{P}_{nj})$$

The simulated log-likelihood is maximized as a function of σ.

10.7 Example

For a practical example, we estimate three mixed logit models using the variables `vcost` (vehicle cost), `travel` (travel time in min), and `wait` (waiting time in minutes). The difference between the models is which variables have random coefficients. The models are as follows:

Model 1 (MIXL T), a model with a random coefficient for *travel* time:

$$\begin{aligned}
V_{n,air} &= & &+\beta_{vcost} vcost_{air} + \beta_{wait} wait_{air} + \beta_{n,travel} travel_{air} \\
V_{n,train} &= \beta_{train} & &+\beta_{vcost} vcost_{train} + \beta_{wait} wait_{train} + \beta_{n,travel} travel_{train} \\
V_{n,bus} &= \beta_{bus} & &+\beta_{vcost} vcost_{bus} + \beta_{wait} wait_{bus} + \beta_{n,travel} travel_{bus} \\
V_{n,car} &= \beta_{car} & &+\beta_{vcost} vcost_{car} + \beta_{wait} wait_{car} + \beta_{n,travel} travel_{car}
\end{aligned}$$

Model 2 (MIXL W), a model with a random coefficient for *wait* time

$$\begin{aligned}
V_{n,air} &= & &+\beta_{vcost} vcost_{air} + \beta_{n,wait} wait_{air} + \beta_{travel} travel_{air} \\
V_{n,train} &= \beta_{train} & &+\beta_{vcost} vcost_{train} + \beta_{n,wait} wait_{train} + \beta_{travel} travel_{train} \\
V_{n,bus} &= \beta_{bus} & &+\beta_{vcost} vcost_{bus} + \beta_{n,wait} wait_{bus} + \beta_{travel} travel_{bus} \\
V_{n,car} &= \beta_{car} & &+\beta_{vcost} vcost_{car} + \beta_{n,wait} wait_{car} + \beta_{travel} travel_{car}
\end{aligned}$$

And Model 3 (MIXL W&T), a model with random coefficients for both *travel* and *wait* time

$$\begin{aligned}
V_{n,air} &= & &+\beta_{vcost}vcost_{air} + \beta_{n,wait}wait_{air} + \beta_{n,travel}travel_{air}\\
V_{n,train} &= \beta_{train} & &+\beta_{vcost}vcost_{train} + \beta_{n,wait}wait_{train} + \beta_{n,travel}travel_{train}\\
V_{n,bus} &= \beta_{bus} & &+\beta_{vcost}vcost_{bus} + \beta_{n,wait}wait_{bus} + \beta_{n,travel}travel_{bus}\\
V_{n,car} &= \beta_{car} & &+\beta_{vcost}vcost_{car} + \beta_{n,wait}wait_{car} + \beta_{n,travel}travel_{car}
\end{aligned}$$

The random coefficients are defined in the following manner, with the random components η based on the normal distributions:

$$\beta_{n,travel} = b_{travel} + \eta_{n,travel}$$

and

$$\beta_{n,wait} = b_{wait} + \eta_{n,wait}$$

The code to estimate the models is as follows:

```
# MIXL T
mixl_t <- gmnl(choice ~ vcost + travel + wait | 1,
               data = TM,
               model = "mixl",
               ranp = c(travel = "n"),
               R = 50)
```

```
  Estimating MIXL model
```

```
mixl_t$logLik$message
```

```
  [1] "successful convergence "
```

```
# MIXL W
mixl_w <- gmnl(choice ~ vcost + travel + wait | 1,
               data = TM,
               model = "mixl",
               ranp = c(wait = "n"),
               R = 50)
```

```
  Estimating MIXL model
```

```
mixl_w$logLik$message
```

```
  [1] "successful convergence "
```

```
# MIXL T&W
mixl <- gmnl(choice ~ vcost + travel + wait | 1,
             data = TM,
             model = "mixl",
             ranp = c(travel = "n", wait = "n"),
             R = 60)
```

```
Estimating MIXL model
```

```
mixl$logLik$message
```

```
[1] "successful convergence "
```

Note that in the absence of individual variables, a value of 1 indicates that constants are to be estimated.

The models are summarized in Table 10.2

```
# Estimate a constants only model to calculate McFadden's _adjusted_ rho2
mixl0 <- gmnl(choice ~ 1,
             data = TM,
             model = "mnl")

mixl_t.summary <- rownames_to_column(data.frame(summary(mixl_t)$CoefTable),
                                     "Variable") %>%
  transmute(Variable,
            Estimate,
            pval = `Pr...z..`)

mixl_w.summary <- rownames_to_column(data.frame(summary(mixl_w)$CoefTable),
                                     "Variable") %>%
  transmute(Variable,
            Estimate,
            pval = `Pr...z..`)

mixl.summary <- rownames_to_column(data.frame(summary(mixl)$CoefTable),
                                   "Variable") %>%
  transmute(Variable,
            Estimate,
            pval = `Pr...z..`)

mixl_table_1 <- full_join(mixl_t.summary,
                          mixl_w.summary,
                          by = "Variable") %>%
  full_join(mixl.summary,
            by = "Variable")

kable(mixl_table_1,
      "latex",
      digits = 4,
      col.names = c("Variable",
                    "Estimate",
                    "p-value",
                    "Estimate",
                    "p-value",
                    "Estimate",
                    "p-value"),
      caption = "\\label{tab:table-mixed-logit-models}Mixed logit models",
      ) %>%
  kable_styling() %>%
  add_header_above(c(" " = 1, "MIXL T" = 2, "MIXL W" = 2, "MIXL T&W" = 2)) %>%
  footnote(general = c(paste0("Log-Likelihood: MIXL T = ",
                              round(mixl_t$logLik$maximum,
                                    digits = 3),
                            "; MIXL W = ",
                            round(mixl_w$logLik$maximum,
                                  digits = 3),
                            "; MIXL T&W = ",
                            round(mixl$logLik$maximum,
                                  digits = 3)),
                       # Calculate McFadden's  rho-2
                       paste0("McFadden Adjusted R^2: MIXL T = ",
                              round(1 - (mixl_t$logLik$maximum - nrow(mixl_t.summary)) /
                                      mixl0$logLik$maximum,
                                    digits = 3),
                              "; MIXL W = ",
                              round(1 - (mixl_w$logLik$maximum - nrow(mixl_w.summary)) /
                                      mixl0$logLik$maximum,
                                    digits = 3),
                              "; MIXL T&W = ",
                              round(1 - (mixl$logLik$maximum - nrow(mixl.summary)) /
                                      mixl0$logLik$maximum,
                                    digits = 3))))
```

The standard deviation of the random coefficients in models MIXL T and MIXL W are estimated. In Model MIXL T&W, the levels of significance decline, with only a small increase in goodness-of-fit. Based on these results, the conclusion is that there is likely taste variations in waiting time, but probably not so in the case of travel time. Other coefficients have signs as expected. It can be seen that with respect to air travel, train is preferred, whereas bus and car are less preferred, other things being equal (see the alternative specific constants).

What do we learn from this model that we did not from the standard multinomial logit model or the nested logit model?

Table 10.2 Mixed logit models

Variable	MIXL T		MIXL W		MIXL T&W	
	Estimate	p-value	Estimate	p-value	Estimate	p-value
train:(intercept)	0.3613	0.7206	−0.0865	0.9230	5.2274	0.3990
bus:(intercept)	−0.4468	0.6673	−0.9021	0.3810	2.9903	0.5327
car:(intercept)	−4.9045	0.0000	−8.2200	0.0001	−13.9055	0.0344
vcost	−0.0261	0.0062	−0.0205	0.0453	−0.0776	0.1676
wait	−0.1164	0.0000	−0.1759	0.0000	−0.3897	0.0806
travel	−0.0081	0.0006	−0.0064	0.0001	−0.0300	0.1820
sd.travel	0.0053	0.0093	NA	NA	0.0194	0.2276
sd.wait	NA	NA	0.1014	0.0049	0.2199	0.0979

Note
Log-Likelihood: MIXL T = −189.105; MIXL W = −178.732; MIXL T&W = −174.898
McFadden Adjusted R^2: MIXL T = 0.309; MIXL W = 0.345; MIXL T&W = 0.355

10.8 Behavioral Insights from the Mixed Logit Model

The random coefficient of the MIXL W model was defined as follows:

$$\beta_{n,wait} = b_{wait} + \eta_{n,wait}$$

with η given by the normal distribution with mean b_{wait} and standard deviation σ. Therefore

$$\beta_{n,wait} \sim N(b_{wait}, \sigma^2)$$

Based on the estimated parameters, we can explore the variation in taste in the population with respect to wait time.

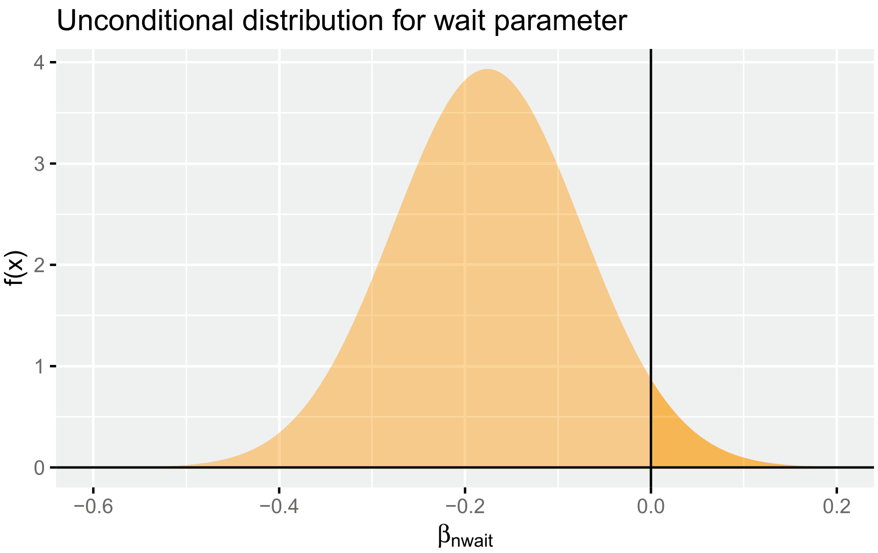
10.8.1 Unconditional Distribution of a Random Parameter

Given the parameters estimated above, we can plot the distribution of the random coefficient. This is done as follows. We first create a data frame with draws from the random distribution

```
# Retrieve the estimated parameters
mu <- coef(mixl_w)['wait']
sigma <- coef(mixl_w)['sd.wait']

# Create a data frame for plotting
df <- data.frame(x =seq(from = -0.6,
                        to = 0.2,
                        by = 0.005)) %>%
  # Draw from the normal distribution for x given the mean and sd
  mutate(normal = dnorm(x,
                        mean = mu,
                        sd = sigma))

# Same, but only positive values of x
df_p <- data.frame(x = seq(from = 0,
                           to = 0.2,
                           by = 0.005)) %>%
  mutate(normal = dnorm(x,
                        mean = mu,
                        sd = sigma))
```

The plot is as follows:

```
# Plot
ggplot() +
  # Plot the distribution
  geom_area(data = df,
            aes(x = x,
                y = normal),
            fill = "orange",
            alpha = 0.5) +
  # Plot the distribution for positive values of x only
  geom_area(data = df_p,
            aes(x = x,
                y = normal),
            fill = "orange",
            alpha = 0.5) +
  geom_hline(yintercept = 0) + # Add y axis
  geom_vline(xintercept = 0) + # Add x axis
  ylab("f(x)") + # Label the y axis
  xlab(expression(beta[n][wait])) + # Label the x axis
  ggtitle("Unconditional distribution for wait parameter")
```



As seen in the figure, the response to waiting time is heterogeneous in the population. While, in general, most people dislike waiting, some dislike waiting more than others. The plot also suggests that some people may in fact *enjoy* their wait time (notice how the distribution includes some positive values for the coefficient). We can calculate what proportion of the population has a positive response to waiting time

```
1 - pnorm(0,
          mean = coef(mixl_w)['wait'],
          sd = coef(mixl_w)['sd.wait'])
```

```
[1] 0.04135726
```

This suggests that about 4.1% of the population have a positive parameter for waiting time. Another way to formulate it is

```
1 - pnorm(-coef(mixl_w)['wait'] / coef(mixl_w)['sd.wait'])
```

```
      wait 
0.04135726 
```

As discussed above, the nested logit model suggested that the willingness to pay for reductions in waiting time was:

$$-\frac{\partial \text{vcost}}{\partial \text{wait}} = \frac{\beta_{wait}}{\beta_{vcost}} = \frac{-0.0554}{-0.0139} \simeq 3.99$$

or approximately 3.99 dollars per minute. This willingness to pay does not consider variations in the preference for waiting time, and therefore, assumes that this value is constant for the population. This is not the case for the preference inferred using the mixed logit model, which recognizes the possibility of variations in taste and is in fact a distribution.

Which begs the question, if the response to waiting time is a random variable, what is then the distribution of the willingness to pay? To answer this, we note that multiplying a normal random variable X by a constant c changes the mean *and* the standard deviation of the distribution as follows:

$$cE[X] = E[cX]$$
$$c^2 Var[X] = Var[cX]$$

Since the willingness to pay is the ratio of a random coefficient to a constant, its distribution depends on the distribution of the random coefficient

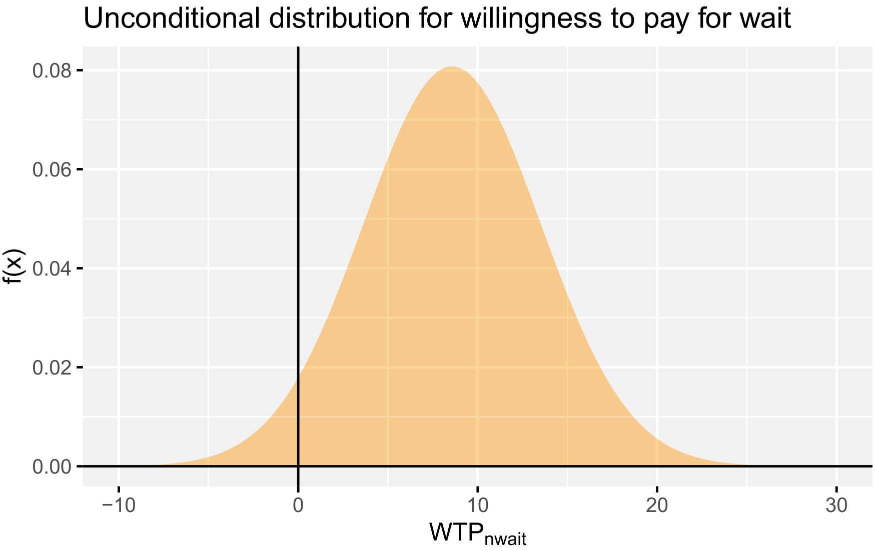
$$-\frac{\partial \text{vcost}}{\partial \text{wait}} \sim N\left(\frac{b_{wait}}{\beta_{vcost}}, \frac{\sigma^2_{wait}}{\beta^2_{vcost}}\right) = N\left(\frac{-0.1759}{-0.0205}, \frac{(0.1014)^2}{(-0.0205)^2}\right)$$

To visualize the distribution of willingness to pay for waiting time, we create a new data frame to draw values from the relevant random distribution

```r
# Define parameters for the distribution of willingness to pay
mu <- coef(mixl_w)['wait'] / coef(mixl_w)['vcost']
sigma <- sqrt(coef(mixl_w)['sd.wait']^2/ coef(mixl_w)['vcost']^2)

# Create a data frame for plotting
df <- data.frame(x =seq(from = -10, to = 30, by = 0.1)) %>%
  mutate(normal = dnorm(x, mean = mu, sd = sigma))
```

The distribution is as follows:

```r
# Plot
ggplot() +
  geom_area(data = df, aes(x, normal), fill = "orange", alpha = 0.5) +
#  geom_area(data = df_p, aes(x, normal), fill = "orange", alpha = 0.5) +
  #ylim(c(0, 1/(2 * L) + 0.2 * 1/(2 * L))) + # Set the limits of the y axis
  geom_hline(yintercept = 0) + # Add y axis
  geom_vline(xintercept = 0) + # Add x axis
  ylab("f(x)") + # Label the y axis
  xlab(expression(WTP[n][wait])) + # Label the x axis
  ggtitle("Unconditional distribution for willingness to pay for wait")
```



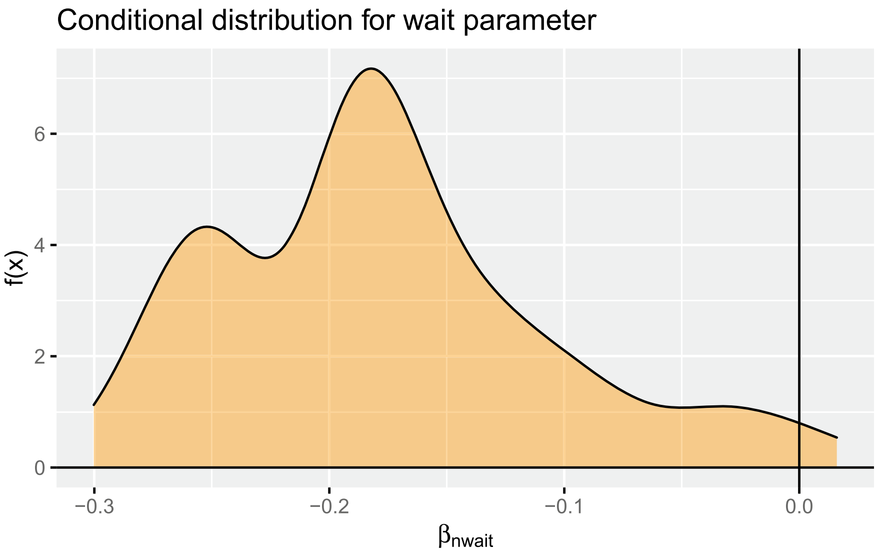
The willingness to pay to reduce waiting time was approximately four dollars per minute according to the nested logit model. The results of the latent class model, however, suggest that there is a segment of the population willing to pay more than that value, but there are also some people who would have to be *paid* to reduce their waiting time (the part of the curve with negative values).

10.8.2 Conditional Distribution of the Random Coefficients

The distributions shown above (for $\beta_{i,wait}$ and for the willingness to pay) are for the population used to estimate the model. The distribution is informative about preferences in the population; for instance, we saw that approximately 4.14 of the population have a *positive* preference for waiting time. A follow-up question is whether that preference varies by individual traveler? Could it be that air travelers have a stronger preference for waiting time than individuals who travel by bus? After all, airports are more glamorous than bus stations (and air travel is more popular than travel by bus, as we see from the modal shares). Train (2009, Chap. 11) describes a procedure to simulate the distribution of random coefficients *conditional* on the choices of individuals in the sample. This distribution is proportional to the density of the coefficient for the population, multiplied by the probability of choosing alternative i conditional on coefficient β_n.

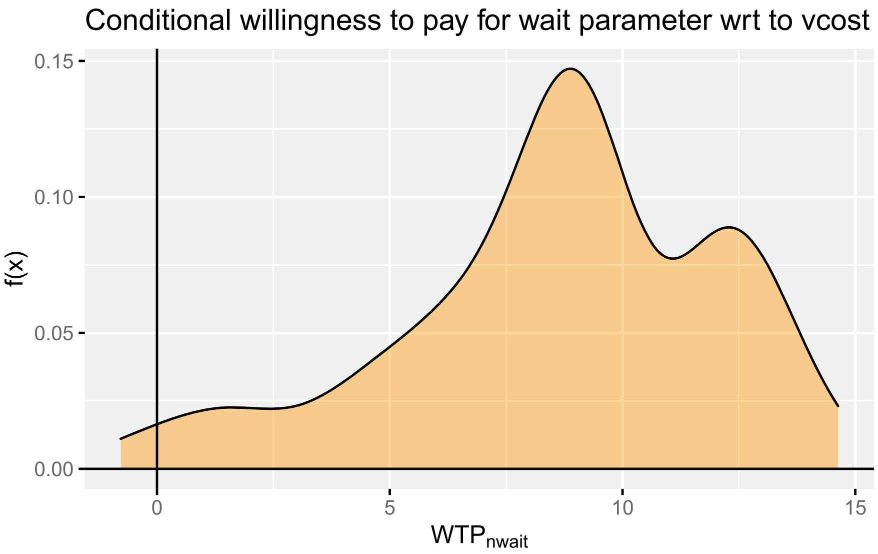
The conditional distribution for a random parameter and willingness to pay can be retrieved from a {gmnl} model by means of the function `effect.gmnl()`, as shown next

```
# Define parameters for the distribution
bn_wait <- effect.gmnl(mixl_w,
                       par = "wait",
                       # Choose conditional effect
                       effect = "ce")

# Create a data frame for plotting
df <- data.frame(bn_wait = bn_wait$mean)

# Plot
ggplot() +
  geom_density(data = df,
            aes(x = bn_wait),
            fill = "orange",
            alpha = 0.5) +
  geom_hline(yintercept = 0) + # Add y axis
  geom_vline(xintercept = 0) + # Add x axis
  ylab("f(x)") + # Label the y axis
  xlab(expression(beta[n][wait])) + # Label the x axis
  ggtitle("Conditional distribution for wait parameter")
```



Likewise, there is a conditional distribution for the willingness to pay

```
# Define parameters for the distribution
wtp_wait <- effect.gmnl(mixl_w,
                        par = "wait",
                        # Effects is willingness to pay
                        effect = "wtp",
                        # With respect to vcost
                        wrt = "vcost")

# Create a data frame for plotting
df <- data.frame(wtp_wait = wtp_wait$mean)

# Plot
ggplot() +
  geom_density(data = df,
            aes(x = wtp_wait),
            fill = "orange",
            alpha = 0.5) +
  geom_hline(yintercept = 0) + # Add y axis
  geom_vline(xintercept = 0) + # Add x axis
  ylab("f(x)") + # Label the y axis
  xlab(expression(WTP[n][wait])) + # Label the x axis
  ggtitle("Conditional willingness to pay for wait parameter wrt to vcost")
```



10.9 Using Covariates to Capture Variations In Taste

The models above strongly suggest that there are important variations in the way decision-makers respond to waiting time. However, no variables were used to characterize the decision-makers. As previously suggested, the variation might happen on a socio-economic dimension, for instance, income. In this section, we will explore whether including individual-level attributes can capture heterogeneity in decision-making, and if so, to what extent.

We will test two different models, with the specifications described below. The first is a multinomial logit with individual-level covariates (MNL-Covariates)

$$
\begin{array}{llll}
V_{n,air} & = & +\beta_{vcost} vcost_{air} + \beta_{wait} wait_{air} + \beta_{travel} travel_{air} & \\
V_{n,train} & = \beta_{train} & +\beta_{vcost} vcost_{train} + \beta_{wait} wait_{train} + \beta_{travel} travel_{train} & +\beta_{train:income} income_i + \beta_{train:size} size_i \\
V_{n,bus} & = \beta_{bus} & +\beta_{vcost} vcost_{bus} + \beta_{wait} wait_{bus} + \beta_{travel} travel_{bus} & +\beta_{bus:income} income_i + \beta_{bus:size} size_i \\
V_{n,car} & = \beta_{car} & +\beta_{vcost} vcost_{car} + \beta_{wait} wait_{car} + \beta_{travel} travel_{car} & +\beta_{car:income} income_i + \beta_{car:size} size_i
\end{array}
$$

This model is similar to the base multinomial logit model, with the addition of individual-level attributes.

The second is a multinomial logit with deterministic expansions of coefficients (MNL-Expansion)

$$
\begin{aligned}
V_{n,air} &= & +\beta_{vcost}vcost_{air} + \beta_{n,wait}wait_{air} + \beta_{n,travel}travel_{air}\\
V_{n,train} &= \beta_{train} & +\beta_{vcost}vcost_{train} + \beta_{n,wait}wait_{train} + \beta_{n,travel}travel_{train}\\
V_{n,bus} &= \beta_{bus} & +\beta_{vcost}vcost_{bus} + \beta_{n,wait}wait_{bus} + \beta_{n,travel}travel_{bus}\\
V_{n,car} &= \beta_{car} & +\beta_{vcost}vcost_{car} + \beta_{n,wait}wait_{car} + \beta_{n,travel}travel_{car}
\end{aligned}
$$

with expanded coefficients as follows:

$$\beta_{n,travel} = b_{travel} + b_{travel:income}income_n + b_{travel:size}size_n$$

and

$$\beta_{n,wait} = b_{wait} + b_{wait:income}income_n$$

To implement the expansion of the coefficients, we need to add variable interactions to the table, and reformat for use with functions of packages {mlogit} and {gmnl}

```
TM <- mutate(TravelMode,
             `wait:income` = wait * income,
             `travel:income` = travel * income,
             `wait:size` = wait * size,
             `travel:size` = travel * size)

TM <- mlogit.data(TM,
                  choice = "choice",
                  shape = "long",
                  alt.levels = c("air",
                                 "train",
                                 "bus",
                                 "car"))
```

The models are estimated using the following code:

```
mnl_cov <- mlogit(choice ~ vcost + travel + wait | income + size,
                  data = TM)

mnl_exp <- mlogit(choice ~ vcost + travel + travel:income + travel:size + wait + wait:income | 1,
                  data = TM)
```

The summary of the models is shown in the table below

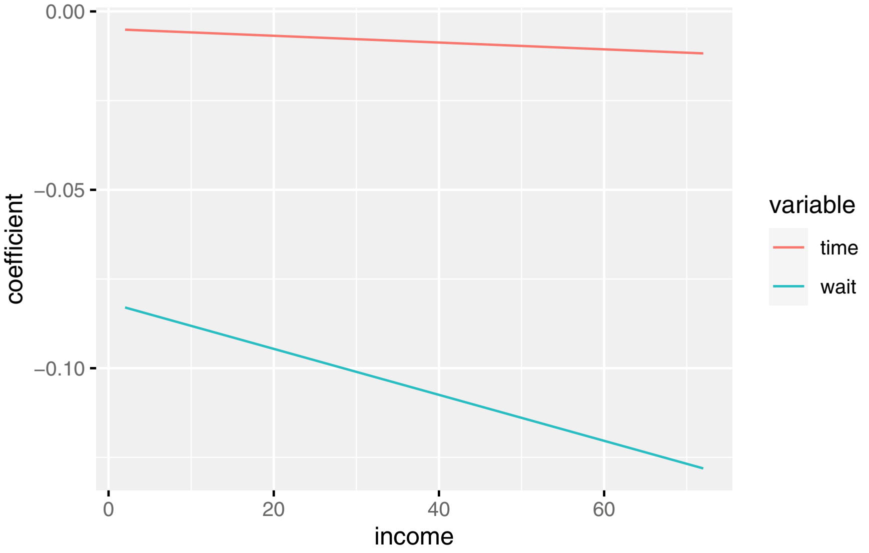
```
mnl_null_lo = -283.83

mnl1.summary <- rownames_to_column(data.frame(summary(mnl_cov)$CoefTable),
                                   "Variable") %>%
  transmute(Variable,
            Estimate,
            pval = `Pr...z..`)

mnl2.summary <- rownames_to_column(data.frame(summary(mnl_exp)$CoefTable),
                                   "Variable") %>%
  transmute(Variable,
            Estimate,
            pval = `Pr...z..`)

df_logit_2 <- mnl1.summary %>%
  full_join(mnl2.summary,
            by = "Variable")

kable(df_logit_2,
      "latex",
      digits = 4,
      col.names = c("Variable",
                    "Estimate",
                    "p-value",
                    "Estimate",
                    "p-value")) %>%
  kable_styling() %>%
  add_header_above(c(" " = 1, "MNL - Covariates" = 2, "MNL - Expansion" = 2)) %>%
  footnote(general = c(paste0("Log-Likelihood: MNL - Covariates = ",
                              round(mnl_cov$logLik[1],
                                    digits = 3),
                            "; MNL - Expansion = ",
                            round(mnl_exp$logLik[1],
                                  digits = 3)),
                       # Calculate McFadden's  rho-2
                       paste0("McFadden Adjusted R^2: MNL - Covariates = ",
                              round(1 - (mnl_cov$logLik[1] - nrow(mnl1.summary)) /
                                      mnl_null_lo,
                                    digits = 3),
                              "; NL = ",
                              round(1 - (mnl_exp$logLik[1] - nrow(mnl2.summary)) /
                                      mnl_null_lo,
                                    digits = 3))))
```

Variable	MNL—Covariates		MNL—Expansion	
	Estimate	p-value	Estimate	p-value
(Intercept):train	−0.4616	0.6288	−0.7241	0.2412
(Intercept):bus	−1.5305	0.1577	−1.3373	0.0541
(Intercept):car	−6.0352	0.0000	−5.1305	0.0000
vcost	−0.0087	0.2710	−00.0158	0.0333
travel	−0.0041	0.0000	−00.0049	0.0004
wait	−00.1012	0.0000	−00.0817	0.0000
income:train	−0.0667	0.0000	NA	NA
income:bus	−0.0284	0.0963	NA	NA
income:car	−0.0075	0.5710	NA	NA
size:train	1.1387	0.0002	NA	NA
size:bus	0.7745	0.0447	NA	NA
size:car	0.9224	0.0004	NA	NA
travel:income	NA	NA	−0.0001	0.0004
travel:size	NA	NA	0.0020	0.0000
wait:income	NA	NA	−0.0006	0.0089

Note
Log Likelihood: MNL—Covariates = −172.468; MNL—Expansion = −174.887
McFadden Adjusted R^2: MNL—Covariates = 0.35; NL = 0.352

Both models above suggest that there is at least some heterogeneity in behavior can be attributed to the two characteristics of the decision-makers investigated (income and party size). In particular, we can see from the expanded coefficients that the disutility of travel time and waiting time is greater at higher levels of income

```
# Create a data frame for plotting
df <- data.frame(income = seq(from = min(TM$income),
                              to = max(TM$income),
                              by = 1)) %>%
  mutate(time = coef(mnl_exp)['travel'] + coef(mnl_exp)['travel:income'] * income,
         wait = coef(mnl_exp)['wait'] + coef(mnl_exp)['wait:income'] * income) %>%
  pivot_longer(cols = -income,
               names_to = "variable",
               values_to = "coefficient")

# Plot
ggplot(df) +
  geom_line(aes(x = income,
                y = coefficient,
                color = variable))
```



However, is the inclusion of these variables sufficient to capture variations in taste? What if we are missing some other relevant variable, such as age? In the following section, we will revisit the specification of the models to test this idea.

10.10 Revisiting the Example

We will re-estimate the mixed logit models, but only after introducing the individual-level covariates as stand-alone variables and also as part of expansions. The expansion of the coefficient for waiting time includes a random component to give a random coefficient as follows:

$$\beta_{n,travel} = b_{travel} + b_{travel:income} income_n + b_{travel:size} size_n$$

and

$$\beta_{n,wait} = b_{wait} + b_{wait:income} income_n + \eta_{n,wait}$$

The models are estimated next

```
mixl_w1 <- gmnl(choice ~ vcost + travel + wait | income + size,
                 data = TM,
                 model = "mixl",
                 ranp = c(wait = "n"),
                 R = 50)
```

```
## Estimating MIXL model
```

```
mixl_w2 <- gmnl(choice ~ vcost + travel + travel:income + travel:size +
                     wait + wait:income | 1,
                data = TM,
                model = "mixl",
                ranp = c(wait = "n"),
                R = 50)
```

```
## Estimating MIXL model
```

and the results are summarized in the table below

```
mixl_w1.summary <- rownames_to_column(data.frame(summary(mixl_w1)$CoefTable),
                                       "Variable") %>%
  transmute(Variable,
            Estimate,
            pval = `Pr...z..`)

mixl_w2.summary <- rownames_to_column(data.frame(summary(mixl_w2)$CoefTable),
                                       "Variable") %>%
  transmute(Variable,
            Estimate,
            pval = `Pr...z..`)

mixl_table_2 <- mixl_w1.summary %>%
  full_join(mixl_w2.summary,
            by = "Variable")

kable(mixl_table_2,
      "latex",
      digits = 4,
      col.names = c("Variable",
                    "Estimate",
                    "p-value",
                    "Estimate",
                    "p-value")) %>%
  kable_styling() %>%
  add_header_above(c(" " = 1, "MIXL W-1" = 2, "MIXL W-2" = 2)) %>%
  footnote(general = c(paste0("Log-Likelihood: MIXL W-1 = ",
                              round(mixl_w1$logLik$maximum,
                                    digits = 3),
                            "; MIXL W-2 = ",
                            round(mixl_w2$logLik$maximum,
                                  digits = 3)),
                       # Calculate McFadden's  rho-2
                       paste0("McFadden Adjusted R^2: MIXL W-1 = ",
                              round(1 - (mixl_w1$logLik$maximum - nrow(mixl_w1.summary)) /
                                      mixl0$logLik$maximum,
                                    digits = 3),
                              "; MIXL W-2 = ",
                              round(1 - (mixl_w2$logLik$maximum - nrow(mixl_w2.summary)) /
                                      mixl0$logLik$maximum,
                                    digits = 3))))
```

Variable	MIXL W-1		MIXL W-2	
	Estimate	p-value	Estimate	p-value
train:(intercept)	0.1422	0.9187	−0.3663	0.6594
bus:(intercept)	−1.1624	0.4455	−1.1091	0.2417
car:(intercept)	−8.5571	0.0000	−7.2119	0.0000
vcost	−0.0104	0.3230	−0.0197	0.0495
travel	−0.0058	0.0001	−0.0065	0.0009
train:income	−0.0741	0.0004	NA	NA
bus:income	−0.0287	0.1784	NA	NA
car:income	−0.0038	0.8757	NA	NA
train:size	1.2021	0.0021	NA	NA
bus:size	0.8061	0.0914	NA	NA
car:size	1.1972	0.0098	NA	NA
wait	−0.1505	0.0000	−0.1179	0.0000
sd.wait	0.0638	0.0029	0.0599	0.0008
travel:income	NA	NA	−0.0001	0.0010
travel:size	NA	NA	0.0024	0.0002
wait:income	NA	NA	−0.0011	0.0213

Note
Log-Likelihood: MIXL W-1 = −163.542; MIXL W-2 = −165.626
McFadden Adjusted R^2: MIXL W-1 = 0.378; MIXL W-2 = 0.381

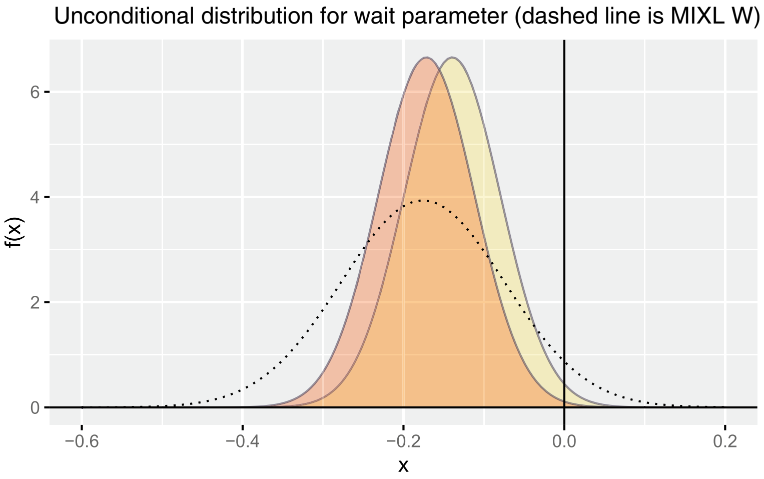
Given the expansion, the distribution of the random coefficient now is

$$\beta_{n_wait} \sim N\left(b_{wait} + b_{wait:income}income_n, \sigma^2_{wait}\right)$$

which means that we can calculate the distribution at different levels of income, as shown in the following figure for the first quartile (yellow) and the third quartile (red) of income. The dashed line is the distribution of willingness to pay according to model MIXL W.

```
# Define parameters for the distribution of willingness to pay
# Obtain quartiles
q <- quantile(TM$income, c(0, 0.25, 0.5, 0.75, 1))

# Define parameters for the distribution
mu_w <- coef(mixl_w)['wait']
sigma_w <- coef(mixl_w)['sd.wait']
```

```
# First quartile
mu_w2.1 <- coef(mixl_w2)['wait'] + coef(mixl_w2)["wait:income"] * q[2]
sigma_w2.1 <- coef(mixl_w2)['sd.wait']

# Third quartile
mu_w2.3 <- coef(mixl_w2)['wait'] + coef(mixl_w2)["wait:income"] * q[4]
sigma_w2.3 <- coef(mixl_w2)['sd.wait']

# Create a data frame for plotting
df_w <- data.frame(x =seq(from = -0.6,
                          to = 0.2,
                          by = 0.005)) %>%
  mutate(normal = dnorm(x,
                        mean = mu_w,
                        sd = sigma_w))

df_w2.1 <- data.frame(x =seq(from = -0.6,
                             to = 0.2,
                             by = 0.005)) %>%
  mutate(normal = dnorm(x,
                        mean = mu_w2.1,
                        sd = sigma_w2.1))
df_w2.3 <- data.frame(x =seq(from = -0.6,
                             to = 0.2,
                             by = 0.005)) %>%
  mutate(normal = dnorm(x,
                        mean = mu_w2.3,
                        sd = sigma_w2.3))

# Plot
ggplot() +
  geom_area(data = df_w2.1,
            aes(x = x,
                y = normal),
            fill = "yellow",
            alpha = 0.3) +
  geom_line(data = df_w2.1,
            aes(x = x,
                y = normal),
            alpha = 0.3) +
  geom_area(data = df_w2.3,
            aes(x = x,
                y = normal),
            fill = "red",
            alpha = 0.3) +
  geom_line(data = df_w2.3,
            aes(x = x,
                y = normal),
            alpha = 0.3) +
  geom_line(data = df_w,
            aes(x = x,
                y =normal),
            linetype = 3) +
  #ylim(c(0, 1/(2 * L) + 0.2 * 1/(2 * L))) + # Set the limits of the y axis
  geom_hline(yintercept = 0) + # Add y axis
  geom_vline(xintercept = 0) + # Add x axis
  ylab("f(x)") + # Label the y axis
  ggtitle("Unconditional distribution for wait parameter (dashed line is MIXL W)")
```

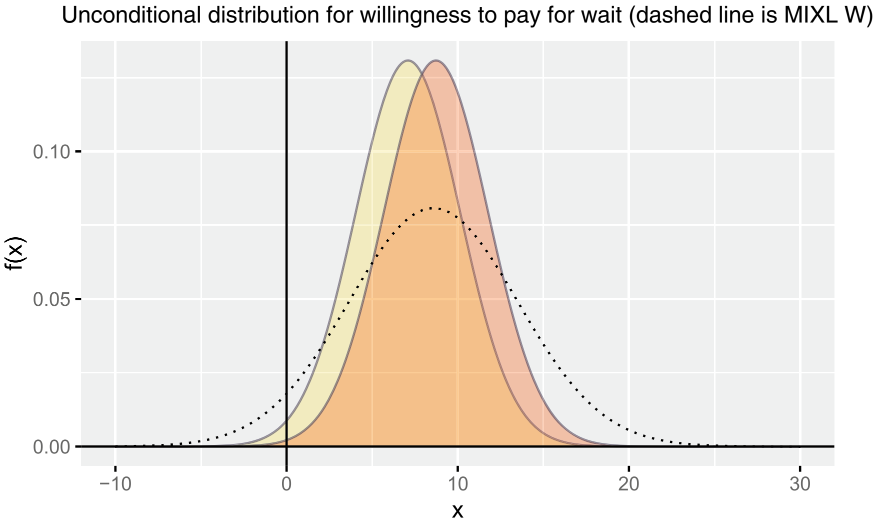


In a similar way, the distribution of the willingness to pay for waiting time according to model MIXL W-2 is

$$-\frac{\partial \text{vcost}}{\partial \text{wait}} \sim N\left(\frac{b_{wait} + b_{wait:income} income_n}{\beta_{vcost}}, \frac{\sigma^2_{wait}}{\beta^2_{vcost}}\right)$$

which is shown in the following figure for the first quartile (yellow) and the third quartile (red) of income. The dashed line is the distribution of willingness to pay according to model MIXL W.

```
# Define parameters for the distribution of willingness to pay
# Obtain quartiles
q <- quantile(TM$income, c(0, 0.25, 0.5, 0.75, 1))

# MIX W2 First quartile
mu_w2.1 <- (coef(mixl_w2)['wait'] + coef(mixl_w2)['wait:income'] * q[2]) *
  (1 / coef(mixl_w2)['vcost'])
sigma_w2.1 <- coef(mixl_w2)['sd.wait'] * sqrt((1 / coef(mixl_w2)['vcost'])^2)

# MIX W2 Third quartile
mu_w2.3 <- (coef(mixl_w2)['wait'] + coef(mixl_w2)['wait:income'] * q[4]) *
  (1 / coef(mixl_w2)['vcost'])
sigma_w2.3 <- coef(mixl_w2)['sd.wait'] * sqrt((1 / coef(mixl_w2)['vcost'])^2)

# MIX W
mu_w <- coef(mixl_w)['wait'] * (1 / coef(mixl_w)['vcost'])
sigma_w <- coef(mixl_w)['sd.wait'] * sqrt((1 / coef(mixl_w)['vcost'])^2)

# Create data frames for plotting
df_w2.1 <- data.frame(x =seq(from = -10,
                             to = 30,
                             by = 0.1)) %>%
  mutate(normal = dnorm(x,
                        mean = mu_w2.1,
                        sd = sigma_w2.1))

df_w2.3 <- data.frame(x =seq(from = -10,
                             to = 30,
                             by = 0.1)) %>%
  mutate(normal = dnorm(x,
                        mean = mu_w2.3,
                        sd = sigma_w2.3))

df_w <- data.frame(x =seq(from = -10,
                          to = 30,
                          by = 0.1)) %>%
  mutate(normal = dnorm(x, mean = mu_w, sd = sigma_w))

# Plot
ggplot() +
  geom_area(data = df_w2.1,
            aes(x = x,
                y = normal),
            fill = "yellow",
            alpha = 0.3) +
  geom_line(data = df_w2.1,
            aes(x = x,
                y = normal),
            alpha = 0.3) +
  geom_area(data = df_w2.3,
            aes(x = x,
                y = normal),
            fill = "red",
            alpha = 0.3) +
  geom_line(data = df_w2.3,
            aes(x = x,
                y = normal),
            alpha = 0.3) +
  geom_line(data = df_w,
            aes(x = x,
                y = normal),
            linetype = 3) +
  #ylim(c(0, 1/(2 * L) + 0.2 * 1/(2 * L))) + # Set the limits of the y axis
  geom_hline(yintercept = 0) + # Add y axis
  geom_vline(xintercept = 0) + # Add x axis
  ylab("f(x)") + # Label the y axis
  labs(title = "Unconditional distribution for willingness to pay for wait (dashed line is MIXL W)")
```



The results indicate that the introduction of individual-level covariates has absorbed some of the heterogeneity previously observed in the random coefficients, but it was not sufficient to completely account for variations in taste.

10.11 Final Remarks

The mixed logit modeling framework addresses two important issues that can arise in the case of the standard multinomial logit model: taste variations and non-proportional substitution patterns. As discussed above, the multinomial probit model is technically capable of doing this as well. So, what are the relative merits of these two modeling approaches? Both are computationally demanding and need numerical integration methods. In our view, the advantage of the mixed logit model is that the results are more intuitive and easier to interpret than the multinomial probit model. Unlike the probit model, where the covariance matrix is of the differences of utilities, in the mixed logit the covariance matrix relates directly to the error components and/or random coefficients. As well, it is possible to show (see Train Chap. 6) that the mixed logit model can approximate to any arbitrary degree any desired discrete choice model, including the multinomial probit model.

10.12 Exercises

1. What is the difference between an error component and a random coefficient in a mixed logit model?
2. In your own words, explain the IIA property as it pertains to the mixed logit model.
3. Load the following data set from the `mlogit` package:

```
data("ModeCanada",
     package = "mlogit")
```

This data set includes information about travel by mode in the Montreal–Toronto corridor. Variable `noalt` is the number of alternatives available to each respondent. Filter all respondents with four alternatives and process the data for use in `gmnl`

```
MC <- mlogit.data(ModeCanada %>%
                    filter(noalt == 4),
                  choice = "choice",
                  shape = "long",
                  alt.levels = c("air",
                                 "train",
                                 "bus",
                                 "car"))
```

Estimate a mixed logit model using this data set. Justify your choice of variables. Which variables do you choose to have random coefficients and why?

4. Graphically show the random coefficient(s) of your model, both the unconditional and conditional distributions. Discuss the results.

References

Greene, William H. (2003). *Econometric analysis* (5th ed.). NJ: Prentice Hall.

Train, K. (2009). *Discrete choice methods with simulation* (2nd ed.). Cambridge: Cambridge University Press.

Chapter 11
Models for Ordinal Responses

Chaos is merely order waiting to be deciphered.

— José Saramago, The Double

Order and simplification are the first steps towards mastery of a subject.

— Thomas Mann

11.1 Ordinal Responses

In the preceding chapters, our focus has been on modeling choices in situations where decision-makers are faced with a set of binomial or multinomial alternatives (e.g., heating systems, commute mode, etc.). The data sets used in the examples represent choices made by individuals or households among different alternatives that had one thing in common: they were measured in a categorical but not ordinal scale (q.v., Chap. 1). As such, the alternatives did not follow a natural order or a logical sequence: a gas central system is not "higher" or "greater" than an electric room system. Similarly, car is not "higher" or "greater" than walking or public transport. This is what we call unordered choices.

In some cases, the alternatives that individuals choose from do indeed follow a natural, logical sequence. Think about restaurant, service, or product ratings for example. Customers are often asked to indicate their level of satisfaction or appreciation on a five point scale (one star to five stars). In such cases, individuals are asked to choose one from among five alternatives that follow a natural order: a five stars rating indicates a higher/greater satisfaction with the product/service than a four stars rating.

A. Páez and G. Boisjoly, *Discrete Choice Analysis with R*, Use R!,
https://doi.org/10.1007/978-3-031-20719-8_11

Ordinal scales are also categorical, but the categories follow a logical sequence. They are used in a variety of contexts, and can sometimes be a consequence of information reduction. Many surveys ask respondents about their income, but to avoid respondent burden and/or to preserve some privacy, instead of asking for a quantity, respondents are asked to choose their income category (e.g., $< 20,000$, $20,000 - 39,999$, $40,000 - 59,000$, etc.) The thing being measured (income) is an interval-ratio variable, but it is measured using a categorical-ordinal scale. Ordinal variables are commonly used in opinion surveys, as well as in health and life satisfaction studies. Their objective is to evaluate the perceptions, opinions, and attitudes of individuals. A few examples of applications of ordinal scales, typically based on Likert scales (see Chap. 1), are listed below

- "Strongly disagree" to "Strongly agree" (e.g., agreement with a political program)
- "Strongly dissatisfied" to "Strongly satisfied" (e.g., satisfaction with life)
- "Poor", "Fair", "Good", "Very Good", "Excellent" (e.g., self-assessed health)
- "Very Low", "Low", "Medium", "High", "Very High" (e.g., level of access to a service)
- "Terrible", "Poor", "Fair", "Good", "Excellent" (e.g., taste of a product).

These scales all have in common that they represent discrete observed outcomes. But in contrast with the unordered choice situations we have covered so far in the book, the discrete outcomes can be sorted by the magnitude of the response. Further, the ordered choices do not represent different alternatives or things per se: they are a measurement of the same entity, on a single dimension. In fact, ordered choices can be thought of as a translation of an underlying but unobserved continuous value into a set of discrete outcomes.

A practical example can help to illustrate this. To improve users' satisfaction and loyalty, public transport agencies regularly conduct satisfaction surveys, in which they ask users to rate their satisfaction with different aspects of their trips. One common question that public transport agencies are concerned with is on board comfort on vehicles:

What is your level of satisfaction with the comfort on board the buses?

- Very dissatisfied (1)
- Dissatisfied (2)
- Neutral (3)
- Satisfied (4)
- Very satisfied (5).

Without question, there is a natural order to the five categories mentioned above. The five categories represent the strength of satisfaction, which is, in fact, an entity on a continuous spectrum. However, unlike objective measures of other quantities (e.g., physical qualities), the intensity of the responses is not necessarily identical across individuals: one person's "very dissatisfied" may be emotionally more loaded than someone else's "very dissatisfied". This is not necessarily a problem, since what we care about is the opinion: as such, individuals have a strength of satisfaction that varies continuously on a spectrum, and we ask them to select the label that corresponds best to their perception of the entity being assessed. Basically, the ordered responses describe a continuous, one-dimensional scale using discrete observed outcomes. As discussed in Chap. 1, numbers are often associated with the different categories, however, it bears reminding that these numbers solely represent labels and not quantities.

In an effort to understand which factors influence users' satisfaction with comfort, a transit agency might want to model the responses to the question about comfort on buses. Could we model ordinal variables as multinomial variables? Technically, it is possible. There is nothing to prevent us from estimating a multinomial model with the categorical responses. That said, this is somewhat wasteful, since it throws away the information contained in the order of the categories. Furthermore, multinomial choice models assume that the alternatives are independent. Such assumption is not warranted in the case of ordinal variables: "very dissatisfied" is closer to "dissatisfied" than it is to "neutral", and neutral is closer to "satisfied" than to "very satisfied". A possible strategy could be to model the outcomes using a multinomial model with correlated alternatives (e.g., nested logit or probit). But conceptually that is problematic. You will recall that those models were derived on the principle that decision-makers compared the utility of various alternatives. However, as we argued above, an ordinal variable is better seen as the measurement of an underlying and unobserved unidimensional quantity. This means that there are not multiple utility functions because there is only one item that is being measured!

For the reasons above, it is not always appropriate to model ordinal outcomes as multinomial variables (although this may depend on one's preferences, see Bhat and Pulugurta 1998). Luckily, the discrete choice framework can be adapted with relative ease to analyze ordered responses. In this chapter, we describe a modeling framework applicable to ordinal variables that builds on the random utility theory and binary choice models discussed in earlier chapters.

Let us come back to our example about satisfaction to illustrate the basic mechanisms of this framework. In our example, the strength of satisfaction can be equated to a utility function that varies from $-\infty$ to ∞. In this case, the utility is derived from the on-board comfort when traveling by transit. Just like the case of unordered models, the utility is unitless. However, as noted above, in multinomial models we assume that each outcome has its own utility: decision-makers associate a utility with each of the alternatives. In the case of ordered responses, there is only one utility function per decision-maker, expressed as follows: the utility of individual n, ranging from $-\infty$ to ∞ is

$$-\infty \leq U_n \leq \infty$$

To understand the behavioral mechanisms underlying ordered responses and the associated single utility function, we return to the transformation from a continuous underlying satisfaction scale to an ordered discrete observed outcome. The utility of a specific individual has a value between $-\infty$ and ∞, but the response is expressed on a five point scale ("Strongly dissatisfied (1)" to "Strongly satisfied (5)"). The transformation from a single value on a continuous scale to the discrete outcomes can be expressed as follows:

$$Y_n = \begin{cases} \text{Very Dissatisfied (1)} & \text{if } -\infty < U_n \leq \lambda_{n1} \\ \text{Dissatisfied (2)} & \text{if } -\lambda_{n1} < U_n \leq \lambda_{n2} \\ \text{Neutral (3)} & \text{if } -\lambda_{n2} < U_n \leq \lambda_{n3} \\ \text{Satisfied (4)} & \text{if } -\lambda_{n3} < U_n \leq \lambda_{n4} \\ \text{Very Satisfied (5)} & \text{if } -\lambda_{n4} < U_n \leq \infty \end{cases}$$

Depending on the value of the unobserved utility with respect to the thresholds λ, an individual chooses one of the 5 points on the scale. For example, if individual n has a satisfaction that falls between λ_{n1} and λ_{n2}, they will choose the second level "Dissatisfied (2)". What we measure is therefore an approximation of the exact (but unobservable) strength of satisfaction or dissatisfaction of the individual.

The thresholds are central to ordered models in that they define what are the ranges of values of the utility that correspond to ordinal categories. The thresholds, which can vary across individuals, are parameters that are estimated in ordinal choice models. They thereby permit the formulation of a choice problem in line with the unordered choice model.

Using the formulation above, it is possible to model an individual choice as a series of binary choice models. Ordered choices models are in fact an extension of the models such as the binary logistic model introduced in Chap. 4. Put simply, the framework first assesses whether the individual's satisfaction is below λ_{n1}. If so, the answer is "Strongly dissatisfied". If not, the model moves to the other level, assessing whether the individual's satisfaction is below λ_{n2}. If so, the answer is "Dissatisfied", if not, the model moves to the other level. Every time, the model considers a choice between two levels. It does not mean that the individuals make their choice following this sequence (same comment as for nested logit), but rather this reflects how the ordered model can be decomposed into a series of linked binary choice models.

This short introduction provides a conceptual overview of ordinal responses and how they relate to the discrete choice framework that we used in the preceding chapters. In the remainder of the chapter, we will further develop this framework for modeling ordinal responses.

11.2 How to Use This Note

Remember that the source code used in this chapter is available. Throughout the notes, you will find examples of code in segments of text called *chunks*. This is an example of a chunk:

```
print("Too late to cry, too soon to laugh...")
```

```
[1] "Too late to cry, too soon to laugh..."
```

You can copy and paste the source code into your R or RStudio console, or create a script/notebook to save the code and any experiments you conduct.

11.3 Learning Objectives

In this chapter, you will learn about:

1. Ordinal responses.
2. Latent variables and ordinal models.
3. Application of the ordinal logit model.
4. Proportional odds.
5. Models for non-proportional odds.
6. Multivariate ordinal models.

11.4 Suggested Readings

- Hirk, R., Hornik, K., & Vana, L. (2020). mvord: An R package for fitting multivariate ordinal regression models, *Journal of Statistical Software, 4*, 1–41 https://doi.org/10.18637/jss.v093.i04
- Maddala, G. S. (1983). Limited-dependent and qualitative variables in econometrics, **Chapter 2, pp. 46–48**. Cambridge University Press.
- Train, K. (2009). Discrete choice methods with simulation, 2nd ed., **Chapter 7, pp. 159–163**. Cambridge University Press.

11.5 Preliminaries

Load the packages used in this chapter:

```
library(discrtr) # A companion package for the book Introduction to Discrete Choice Analysis with `R`
library(dplyr) # A Grammar of Data Manipulation
library(ggplot2) # Create Elegant Data Visualisations Using the Grammar of Graphics
library(ggridges) # Ridgeline Plots in 'ggplot2'
library(kableExtra) # Construct Complex Table with 'kable' and Pipe Syntax
library(mvord) # Multivariate Ordinal Regression Models
library(ordinal) # Regression Models for Ordinal Data
library(plyr) # Tools for Splitting, Applying and Combining Data
library(tidyr) # Tidy Messy Data
```

Load the data used in this chapter; the data are part of package {discrtr} and provide information about students that commuted to McMaster University in the fall of 2010:

```
data("mc_attitudes",
     package = "discrtr")
```

These data were collected as part of a university-wide travel survey that collected information about respondents (age, gender, whether they own a driver license, etc.) and responses to attitudinal statements. The data set was augmented with information from the census about the social environment (at the level of Dissemination Areas) and the physical environment (e.g., land uses, attributes of the transportation network). To learn more about the items in the table, use `?mc_attitudes`. The table includes responses to the following six attitudinal statements that refer to their neighborhoods and experience commuting to school:

- "There is a sense of community in my neighborhood" (`Community`)
- "I like to live in a neighborhood where there's a lot going on" (`Active_Neighborhood`)
- "I know my neighbors well" (`Neighbors`)
- "I feel safe and secure when walking in my neighborhood" (`Safe_Walk`)
- "Having shops and services within walking distance of my home is important to me" (`Shops_Important`)
- "I like traveling alone" (`Travel_Alone`)

Responses to the attitudinal statements variables were collected using a five point Likert scale, from "Strongly Disagree" to "Strongly Agree" with a neutral point:

```
mc_attitudes %>%
  select(Community:Travel_Alone) %>%
  summary()
```

```
            Community                Neighbors                Safe_Walk
 STRONGLY DISAGREE: 69   STRONGLY DISAGREE:233   STRONGLY DISAGREE:  5
 DISAGREE         :299   DISAGREE         :445   DISAGREE         : 67
 NEUTRAL          :414   NEUTRAL          :269   NEUTRAL          :169
 AGREE            :391   AGREE            :234   AGREE            :683
 STRONGLY AGREE   : 57   STRONGLY AGREE   : 49   STRONGLY AGREE   :306
         Shops_Important              Travel_Alone
 STRONGLY DISAGREE:  6    STRONGLY DISAGREE: 52
 DISAGREE         : 36    DISAGREE         :314
 NEUTRAL          :131    NEUTRAL          :476
 AGREE            :563    AGREE            :335
 STRONGLY AGREE   :494    STRONGLY AGREE   : 53
```

11.6 Modeling Ordinal Variables

The key difference between modeling ordinal variables and categorical (unordered) variables is that in the case of ordinal variables the alternatives are the *level* of the choice.

Consider for example the statement "There is a sense of community in my neighborhood". There are five responses:

- Strongly Disagree
- Disagree
- Neutral
- Agree
- Strongly Agree.

In multinomial choice situations, the alternatives are distinct entities: travel by helicopter or ferry. The response to an attitudinal statement does not refer to different entities, but rather the degree of agreement or disagreement with a single entity: the perception of a community by the respondent.

Putting this in the context of a decision-making rule, we can think of the decision-maker as responding to the choice situation based on how great the utility they gain from the situation is. Remember, in the case of ordinal responses, there is only one utility function per decision maker. We define the utility of decision-maker n as

$$U_n = f(Z_n)$$

where Z_n is a vector of attributes that describes the context of the decision-making situation and typically includes individual attributes of the decision-maker ($Z = [z_1, \cdots, z_k]$). As before (see Chap. 3), this utility can be decomposed into systematic and random utilities:

$$U_n = V(Z_n) + \epsilon_n$$

Continuing with the example, decision-maker n responds with "Strongly Disagree" if their utility is less than some value λ_1 (i.e., if their utility is "low"):

$$Y_n = SD \leftrightarrow U_n \leq \lambda_1$$

where λ_1 is a threshold that indicates the level of utility below which the respondent disagrees that there is a sense of community in their neighborhood. Notice that currently we are not subindexing λ by n, that is, we assume that the thresholds are identical for all decision-makers. Imagine for a moment that λ_1 is somehow known. In general it is not, but even if it was we do not know whether $U_n \leq \lambda_1$ is true because

U_n contains a random component ϵ_n. As we did in the case of multinomial choice models, we approach this by means of a probability statement:

$$P(Y_n = SD) = P(U_n \leq \lambda_1) = P(V_n + \epsilon_n \leq \lambda_1)$$

Rearranging the expression above we see that the probability of $Y = Y_{SD}$ equals the probability that the random utility is less than the systematic utility *when the systematic utility is lower than the threshold*:

$$P(Y_n = SD) = P(\epsilon_n \leq \lambda_1 - V_n)$$

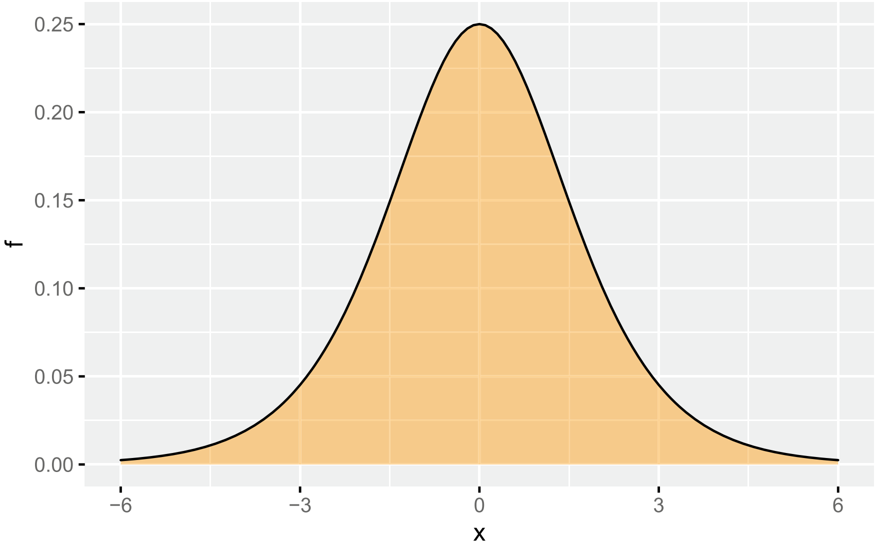
As before, the model that we obtain from this depends on the assumptions made about the random utility. For example, if we assume that the random utility follows the logistic distribution, the resulting model is the *ordinal logit model*. In contrast, if we assume that the random utility follows the normal distribution, the model that results is the *ordinal probit model*.

To illustrate the derivation of the model, we will work with the logistic distribution. This distribution is as follows (also see Chap. 3):

$$f(x; \mu, \sigma) = \frac{e^{-(x-\mu)/\sigma}}{(1 + e^{-(x-\mu)/\sigma})^2}$$

The logistic distribution is defined by two parameters. In the equation above, μ is the location parameter of the distribution and σ is the scale parameter. This distribution is plotted below (assuming that $\mu = 0$ and $\sigma = 1$):

```
df <- data.frame(x = rep(seq(-6, 6, 0.1))) %>%
  mutate(f = dlogis(x,
                    location = 0,
                    scale = 1))

ggplot() +
  geom_area(data = df,
            aes(x = x,
                y = f),
            color = "black",
            fill = "orange",
            alpha = 0.5)
```



The distribution expressed in terms of the utilities can be written as

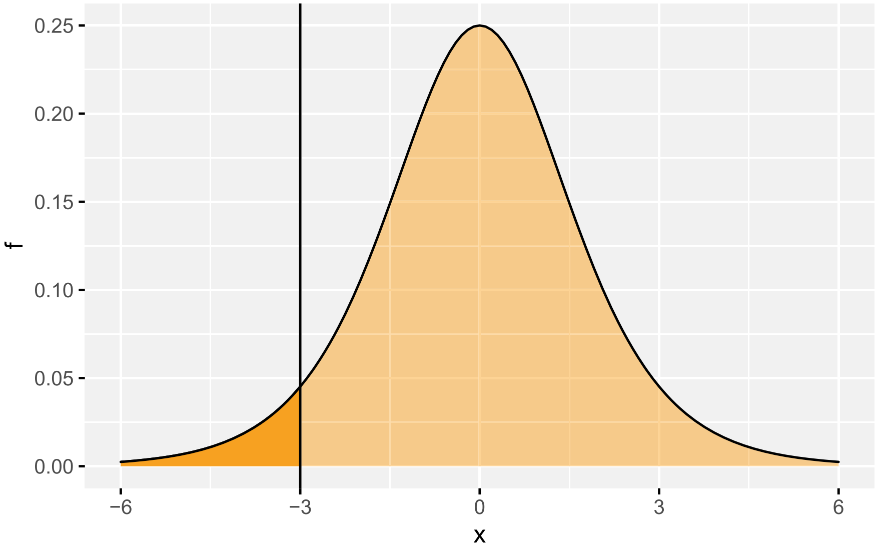
$$f(x;\mu,\sigma) = \frac{e^{-(\epsilon_n-\lambda_1-V_n)/\sigma}}{(1+e^{-(\epsilon_n-\lambda_1-V_n)/\sigma})^2}$$

The random variable is the random utility, and the location parameter of the distribution is the systematic utility. As we know, the integral of the logistic distribution has a closed form (i.e., an analytical solution). This integral is

$$F(x;\mu,\sigma) = \frac{e^{(\lambda_1-V_n)/\sigma}}{1+e^{(\lambda_1-V_n)/\sigma}}$$

This is equivalent to finding the area under the curve to the left of λ_1. Suppose that $\lambda_1 = -3$, then the area under the curve is the darker part of the curve to the left of -2:

```
df <- data.frame(x = rep(seq(-6, 6, 0.1))) %>%
  mutate(f = dlogis(x,
                    location = 0,
                    scale = 1))

ggplot() +
  geom_area(data = df,
            aes(x = x,
                y = f),
            color = "black",
            fill = "orange",
            alpha = 0.5) +
  geom_area(data = df %>%
              filter(x <= -3),
            aes(x = x,
                y = f),
            color = "black",
            fill = "orange",
            alpha = 1) +
  geom_vline(xintercept = -3)
```



In this way, we can express the probability of decision-maker n responding with "Strongly Disagree" as (and notice how this expression is formally a binary logit probability):

$$P(Y_n = SD) = \frac{e^{(\lambda_1 - V_n)/\sigma}}{1 + e^{(\lambda_1 - V_n)/\sigma}}$$

This, however, gives only the probability of one of the possible responses. Consider next what happens when the utility is above the threshold for strongly disagreeing: given their ordinal nature, the response then must be at least "Disagree". However, if the utility is not yet large enough for the individual to state that they are "Neutral" about the statement, then the response *must* be "Disagree". The utility corresponding to "Disagree" must lie between the thresholds where the utility is sufficiently low to "Strongly Disagree" or sufficiently high to be "Neutral":

$$Y_n = D \leftrightarrow \lambda_1 < U_n \leq \lambda_2$$

with $\lambda_1 \leq lambda_2$.

The associated probability statement is now:

$$P(Y_n = D) = P(\lambda_1 < U_n \leq \lambda_2) = P(\lambda_1 < V_n + \epsilon_n \leq \lambda_2)$$

This can be rewritten as

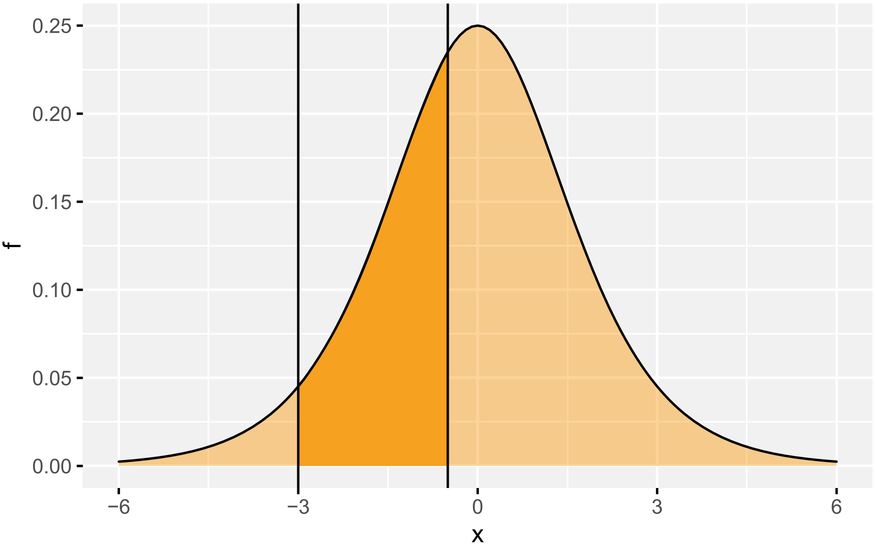
$$P(Y_n = D) = P(\lambda_1 - V_n < \epsilon_n \leq \lambda_2 - V_n) = P(\epsilon_n \leq \lambda_2 - V_n) - P(\epsilon_n < \lambda_1 - V_n)$$

And as we saw above, $P(\epsilon_n < \lambda_1 - V_n)$ is simply $P(Y = Y_{SD})$. In the case of the logistic model, this probability is

$$P(Y_n = D) = \frac{e^{(\lambda_2 - V_n)/\sigma}}{1 + e^{(\lambda_2 - V_n)/\sigma}} - \frac{e^{(\lambda_1 - V_n)/\sigma}}{1 + e^{(\lambda_1 - V_n)/\sigma}}$$

which is equivalent to: (1) the difference between two binary logit probabilities; and (2) the area under the curve to the left of λ_2 *minus* the area under the curve to the left of λ_1, as shown below for $\lambda_2 = -0.5$:

```r
ggplot() +
  geom_area(data = df,
            aes(x = x,
                y = f),
            color = "black",
            fill = "orange",
            alpha = 0.5) +
  geom_area(data = df %>%
              filter(x >= -3 & x <= -0.5),
            aes(x = x,
                y = f),
            color = "black",
            fill = "orange",
            alpha = 1) +
  geom_vline(xintercept = c(-3, -0.5))
```



Other probabilities are derived in the same fashion. For example, the probability of “Neutral” is

$$P(Y_n = N) = P(\lambda_2 - V_n < \epsilon_n < \lambda_3 - V_n) = P(\epsilon_n < \lambda_3 - V_n) - P(\epsilon_n < \lambda_2 - V_n)$$

which in the case of the logit model becomes:

$$P(Y_n = N) = \frac{e^{(\lambda_3 - V_n)/\sigma}}{1 + e^{(\lambda_3 - V_n)/\sigma}} - \frac{e^{(\lambda_2 - V_n)/\sigma}}{1 + e^{(\lambda_2 - V_n)/\sigma}}$$

In the general case when there are J ordered responses, say $k_1, k_2, \ldots, k_i, \ldots, k_J$, these probabilities are:

$$\begin{array}{l}
P(Y_n = k_1) = \frac{e^{(\lambda_1 - V_n)/\sigma}}{1+e^{(\lambda_1 - V_n)/\sigma}} \\
P(Y_n = k_2) = \frac{e^{(\lambda_2 - V_n)/\sigma}}{1+e^{(\lambda_2 - V_n)/\sigma}} - \frac{e^{(\lambda_1 - V_n)/\sigma}}{1+e^{(\lambda_1 - V_n)/\sigma}} \\
\ldots \\
P(Y_n = k_i) = \frac{e^{(\lambda_i - V_n)/\sigma}}{1+e^{(\lambda_i - V_n)/\sigma}} - \frac{e^{(\lambda_{i-1} - V_n)/\sigma}}{1+e^{(\lambda_{i-1} - V_n)/\sigma}} \\
\ldots \\
P(Y_n = k_J) = 1 - \sum_{i=1}^{J-1} P(Y_n = k_i)
\end{array}$$

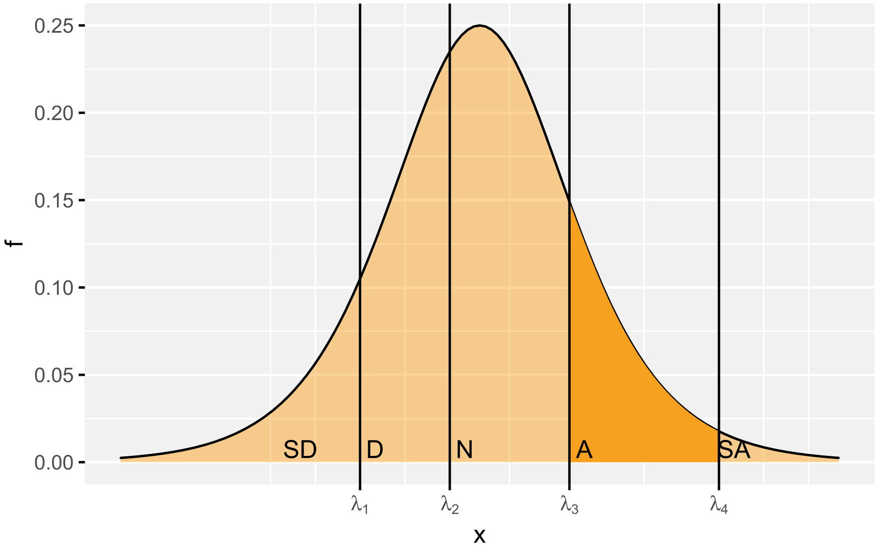
with $-\infty \equiv \lambda_1 \leq \lambda_2 \leq \cdots \lambda_i \leq \cdots \lambda_{J-1} \equiv \infty$, and:

$$\sum_{i=1}^{J} P(Y_n = k_i) = 1$$

This model is variously called *ordered logit model*, *ordinal logistic regression*, or *cumulative link model* (e.g., with a logit link). The last probability expression (i.e., $P(Y_n = k_J)$) is simply the reminder of the area under the curve after we aggregate the probabilities for responses k_i with $i = 1, \ldots, J - 1$.

To conclude this illustration with the response to the statement about sense of community, the following plot shows the probabilities of an uninformative model (i.e., $V_n = 0$, no information is going into defining the systematic utility), assuming that we know that the values of the thresholds are $\lambda_1 = -2$, $\lambda_2 = -0.5$, $\lambda_2 = 1.5$, $\lambda_2 = 4$:

```r
df <- data.frame(x = rep(seq(-6, 6, 0.1))) %>%
  mutate(f = dlogis(x,
                    location = 0))

lj <- c(-2, -0.5, 1.5, 4)

ggplot() +
  geom_area(data = df,
            aes(x = x,
                y = f),
            color = "black",
            fill = "orange",
            alpha = 0.5) +
  geom_area(data = df %>%
              filter(x >= lj[3] & x <= lj[4]),
            aes(x = x,
                y = f),
            color = NA,
            fill = "orange",
            alpha = 1) +
  geom_vline(xintercept = lj) +
  geom_text(aes(label = c("SD", "D", "N", "A", "SA"),
                x = c(-3, lj + 0.25)),
            y = 0.0075) +
  scale_x_continuous(breaks = lj,
                     labels = expression(lambda[1],
                                         lambda[2],
                                         lambda[3],
                                         lambda[4]))
```



In the plot above the darker area is the probability of the response being "Agree". The uninformative model with $Vn = 0$ and $\sigma = 1$, simplifies the probabilities to the area under the curve to the left of each threshold:

$$P(Y \leq k_i) = \frac{e^{\lambda_i/\sigma}}{1 + e^{\lambda_i/\sigma}}$$

11.7 But What About Those Thresholds?

Up to this point we illustrated the probabilities as if we knew the values of the thresholds λ_i. In practice we seldom know what those parameters are. However, it is easy to see that they are, in our current formulation, simply constants. So, we can imagine that the thresholds are absorbed by the systematic utilities V_n as if they were constant terms (similar to the alternative-specific constants in multinomial models). This helps to see that the thresholds are *estimable* parameters, along with any other parameters in the systematic utility, which as we know usually takes the following form:

$$V_n = \beta Z_n$$

11.8 And What About the Scale Parameter?

The scale parameter σ can be normalized to 1. Similar to the case of multinomial models, this normalization does not affect the cardinality of the utilities and thus does not unduly affect the decision-making rule.

11.9 Example

We now proceed to illustrate the ordinal logistic model with an example. For this we will use the variable `Community` (for "There is a sense of community in my neighborhood"). Sense of community has been studied by planners (e.g., Talen 1999; Rogers and Sukolratanametee 2009), community psychologists (e.g., McMillan and Chavis 1986), sociologists (e.g., Grannis 2009), and transportation researchers (e.g., Whalen et al. 2012), among others, and is of interest because it relates to quality of life, the effectiveness of collective action, and well-being.

Grannis (2009), in his theory of T-communities, posits a sequence of events that lead to the formation of communities of neighbors. Geographical co-location is a necessary condition for the emergence of communities, but the formation of relationships of trusts hinges on casual encounters. Thus, it is necessary to be in proximity to others, but also to see them, to say hello to them, to eventually chat with them, before relationships of trust are established. From a transportation and planning perspective, a question is how contextual variables such as development density and the use of different modes of transportation can influence sense of community.

Here, we will estimate a model for the categorical variable for sense of community using variables that describe the decision-maker (their `age` and whether they have access to a private `vehicle`); and attributes of their social environment (the proportion of non-Canadian residents in their neighborhood, `Rate_Non_Canadian`, and the proportion of workers who commute by public transportation, `Rate_Public`). Note that here "neighborhood" means a Dissemination Area, the smallest census geography with socio-demographic information publicly available in Canada.

With these variables, the following utility function becomes:

$$V_n = f(age_n, vehicle_n, Rate\_Non\_Canadian_n, Rate\_Public_n)$$

The model is estimated using the function `clm` (for "cumulative link model") from package {ordinal}:

```
mod_community_clm <- clm(Community ~ age + vehicle +
                           Rate_Non_Canadian + Rate_Public,
                         data = mc_attitudes)

summary(mod_community_clm)
```

```
formula: Community ~ age + vehicle + Rate_Non_Canadian + Rate_Public
data:    mc_attitudes

 link  threshold nobs logLik   AIC     niter max.grad cond.H 
 logit flexible  1230 -1671.97 3359.94 6(0)  3.20e-09 1.7e+05

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)   
age                0.03319    0.01168   2.842  0.00449 **
vehicleYes        -0.35811    0.11058  -3.239  0.00120 **
Rate_Non_Canadian -2.10287    0.81057  -2.594  0.00948 **
Rate_Public       -1.79157    0.78901  -2.271  0.02317 * 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:
                           Estimate Std. Error z value
STRONGLY DISAGREE|DISAGREE  -2.6743     0.3045  -8.781
DISAGREE|NEUTRAL            -0.6801     0.2848  -2.388
NEUTRAL|AGREE                0.7655     0.2846   2.690
AGREE|STRONGLY AGREE         3.2824     0.3139  10.458
```

By default, the scale parameter is normalized (i.e., set to 1). We can check the status of the estimation by inspecting the `convergence` to verify that the model successfully converged:

```
mod_community_clm$convergence
```

```
$code
[1] 0

$messages
[1] "successful convergence"

$alg.message
[1] "Absolute and relative convergence criteria were met"
```

The output of the model includes the values of the estimated parameters, including the threshold coefficients (i.e., λ_i). The positive coefficient for age indicates that older individuals tend to be more agreeable about the statement "there is a sense of community in my neighborhood". The negative coefficient for `vehicleYes` means that individuals who have individual access to a private vehicle are less likely to agree that there is a sense of community in their neighborhood. Individuals in neighborhoods with higher proportions of non-Canadian residents and commuters who travel by transit are also less likely to agree that there is a sense of community in their neighborhoods.

The parameter estimates can be interpreted as the factor by which the distribution of the systematic utilities is shifted on the x-axis. Let us unpack this. In the following chunk of code, we begin by creating a few hypothetical individual profiles. First, we will retrieve the coefficients estimated above (both those in the utility function and the thresholds):

```
# Coefficients of the utility function
beta_n <- mod_community_clm$coefficients[5:8]

# Thresholds
l_j <- mod_community_clm$coefficients[1:4]
```

Given these coefficients, we start with a hypothetical individual who is 21 years old, who does *not* have individual access to a private vehicle, and who lives in a neighborhood with zero non-Canadian residents and transit commuters:

```
# Profile 1: Age = 21, No vehicle, other variables at zero
p1 <- c(21,
        0,
        0,
        0)
```

Next, we will simulate a second profile, where the only difference is that the person has individual access to a private vehicle:

```
# Age = 21, vehicle, other variables at zero
p2 <- c(21,
        1,
        0,
        0)
```

Given those inputs, we can calculate the utility of these two hypothetical individuals:

```
V_1 <- beta_n %*% p1
V_2 <- beta_n %*% p2
```

Again, the systematic utility is equivalent to the location parameter of the distribution, and we assumed that $\sigma = 1$ (i.e., we normalized the scale parameter). In this way, we can calculate the distribution for the two hypothetical profiles as

```
# Calculate the values of the logistic distribution for profile 1 in the range defined by x
dV_1 <- data.frame(x = rep(seq(-2.9, 4.4, 0.1)),
                   profile = "Profile 1") %>%
  mutate(f = dlogis(x,
                    location = V_1))

# Calculate the values of the logistic distribution for profile 2 in the range defined by x
dV_2 <- data.frame(x = rep(seq(-2.9, 4.4, 0.1)),
                   profile = "Profile 2") %>%
  mutate(f = dlogis(x,
                    location = V_2))

# Bind the distributions for the two profiles
dV <- rbind(dV_1,
            dV_2)
```

To illustrate the shifts in the location of the distribution due to various variables, we will begin by creating a base plot to compare the distributions.

```
# Create a base plot to compare distributions
base_plot <- ggplot() +
  # Plot the thresholds
  geom_vline(xintercept = mod_community_clm$coefficients[1:4]) +
  # Label the responses
  geom_text(aes(label = c("SD", "D", "N", "A", "SA"),
                x = c(-3, l_j + 0.25)),
            y = 0.75) +
  # Label the ticks in the x axis with the threshold coefficients
  scale_x_continuous(name = "x",
                     breaks = l_j,
                     labels = expression(lambda[1],
                                         lambda[2],
                                         lambda[3],
                                         lambda[4]))
```

Figure 11.1 shows the shift in the distribution between profiles 1 and 2 and due to `vehicle`. The values of the utilities for the two profiles (the location parameters of the distributions) are annotated in the plot with red dashed lines. As seen in the figure, the distribution for profile 2 (who has access to a vehicle) is shifted by the magnitude of the coefficient for `vehicle` (approximately −0.3581), and the direction of the shift is to the left due to the negative sign of the coefficient.

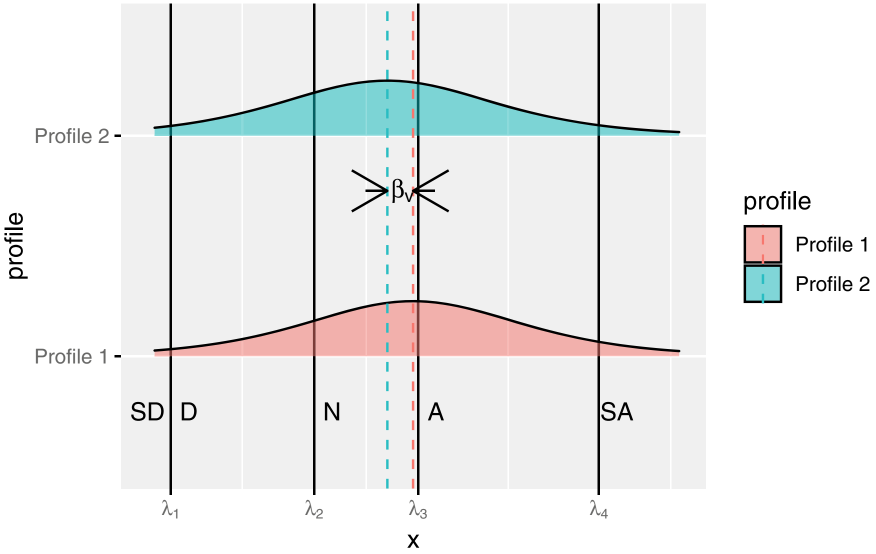


Fig. 11.1 Comparing the distribution of hypothetical profiles 1 and 2

```
# Render the plot and annotate the effect of `vehicle`
base_plot  +
  # Plot the distributions as ridges
  geom_ridgeline(data = dV,
                 aes(x = x,
                     y = profile,
                     fill = profile,
                     group = profile,
                     height = f),
                 alpha = 0.5) +
  # Mark the center of the two distributions
  geom_vline(data = data.frame(profile = c("Profile 1", "Profile 2"),
                               x = c(V_1, V_2)),
             aes(xintercept = x,
                 color = profile,
                 group = profile),
             linetype = "dashed") +
  # Annotate the shift in the location of the distribution due to
  # having access to a vehicle
  annotate("segment",
           x = V_1 + 0.3, xend = V_1,
           y = 1.75, yend = 1.75,
           arrow = arrow(),
           color = "black") +
  annotate("segment",
           x = V_2 - 0.3, xend = V_2,
           y = 1.75, yend = 1.75,
           arrow = arrow(),
           color = "black")+
  annotate("text",
           parse = TRUE,
           label = "beta[v]",
           x = V_1 - 0.15,
           y = 1.75,
           color = "black")
```

The effect of the shift in the location of the distribution is to change the areas under the curve with respect to the various thresholds: other things being equal, the individual with access to a vehicle (profile 2) has a higher probability of responding “Strongly Disagree”, whereas the individual without vehicle (profile 1) has a higher probability of responding “Strongly Agree”. It is not necessarily straightforward what the shift means for other responses, but to illuminate this question we can calculate those probabilities by using the `predict` function using our hypothetical profiles:

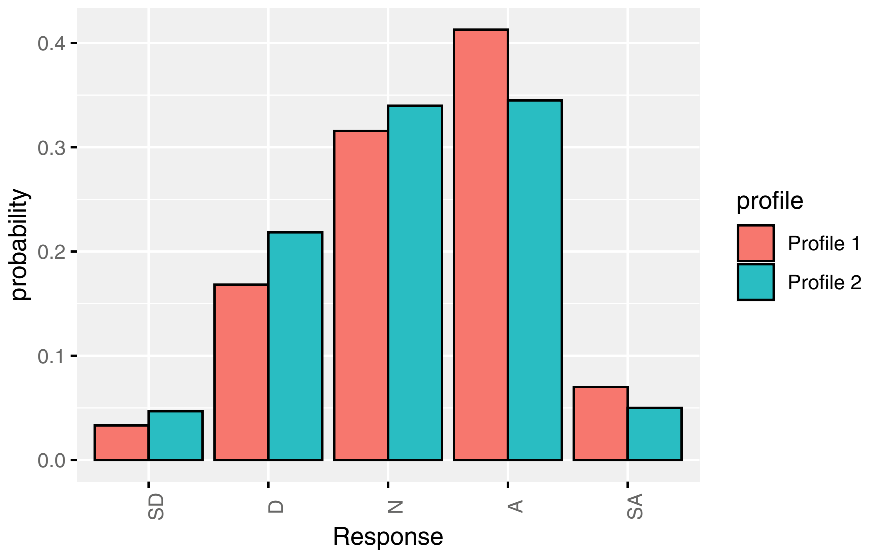
```
profiles <- data.frame(profile = c("Profile 1",
                                   "Profile 2"),
                       age = 21,
                       vehicle = c("No", "Yes"),
                       Rate_Non_Canadian = 0,
                       Rate_Public = 0)
```

```
profiles <- cbind(profiles,
                  predict(mod_community_clm,
                          newdata = profiles))

profiles %>%
  select(-c(age, vehicle, Rate_Non_Canadian, Rate_Public))
```

```
    profile fit.STRONGLY DISAGREE fit.DISAGREE fit.NEUTRAL fit.AGREE
1 Profile 1            0.03320134    0.1682626   0.3156495 0.4128011
2 Profile 2            0.04682896    0.2183811   0.3398485 0.3448960
  fit.STRONGLY AGREE
1         0.07008558
2         0.05004543
```

The sum of the probabilities for each profile adds up to 1, as expected. A bar chart helps to understand the differences between the probabilities for every response between the two profiles:

```
profiles %>%
  select(-c(age, vehicle, Rate_Non_Canadian, Rate_Public)) %>%
  # Pivot longer to gather all probabilities in a sigle column
  pivot_longer(cols = -profile,
               names_to = "Response",
               values_to = "probability") %>%
  # Code the response in order
  mutate(Response = factor(Response,
                           levels = c("fit.STRONGLY DISAGREE",
                                      "fit.DISAGREE",
                                      "fit.NEUTRAL",
                                      "fit.AGREE",
                                      "fit.STRONGLY AGREE"),
                           labels = c("SD", "D", "N", "A", "SA"),
                           ordered = TRUE)) %>%
  ggplot() +
  geom_col(aes(x = Response,
               y = probability,
               fill = profile),
           color = "black",
           position = "dodge") +
  theme(axis.text.x = element_text(angle = 90))
```



As noted earlier, the probability of responding “Strongly Disagree” is larger for profile 2 and the probability of responding “Strongly Agree” is larger for profile 1. But we see that the patterns of probabilities can be markedly non-linear: the probabilities do not increase or decrease monotonically along the dimension of the responses.

The preceding example shows the effect of a single variable (and a relatively simple one, since it was a dummy variable). More generally, the shift is proportional to the coefficient, as we will illustrate next with a third profile. This profile is based on profile 2, but now transit commuters in the neighborhood are set to the in-sample median value of this variable (approximately 11.76%):

```r
# Define profile 3
p3 <- c(21,
        1,
        0,
        median(mc_attitudes$Rate_Public))

# Calculate the utility
V_3 <- beta_n %*% p3

# Calculate the values of the logistic distribution for profile 2 in the range defined by x
dV_3 <- data.frame(x = rep(seq(-2.9, 4.4, 0.1)),
                   profile = "Profile 3") %>%
  mutate(f = dlogis(x,
                    location = V_3))

# Bind the distributions for the two profiles
dV <- rbind(dV,
            dV_3)
```

Figure 11.2 compares the probability distributions of the three profiles in our example:

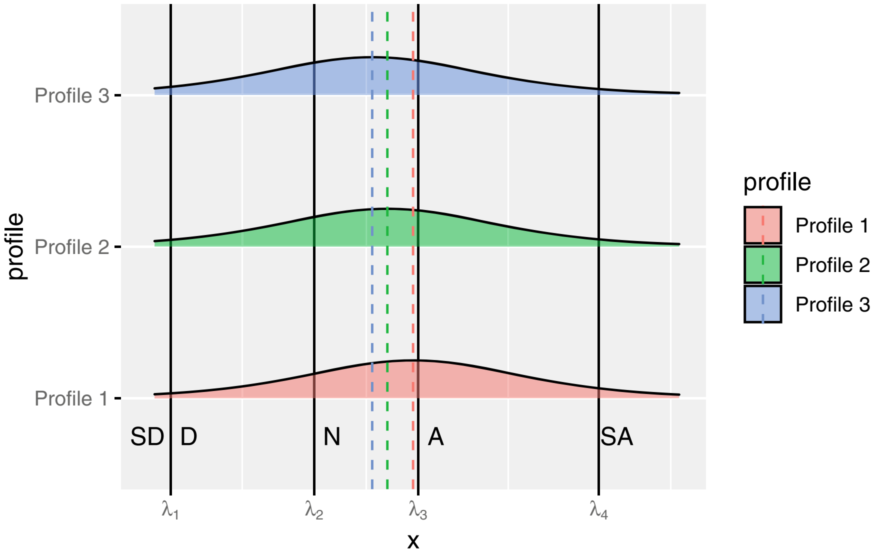


Fig. 11.2 Comparing the distribution of hypothetical profiles 1, 2, and 3

```
# Render the plot and add the distributions
base_plot  +
  # Plot the distributions as ridges
  geom_ridgeline(data = dV,
                 aes(x = x,
                     y = profile,
                     fill = profile,
                     group = profile,
                     height = f),
                 alpha = 0.5) +
  # Mark the center of the three distributions
  geom_vline(data = data.frame(profile = c("Profile 1", "Profile 2", "Profile 3"),
                               x = c(V_1, V_2, V_3)),
             aes(xintercept = x,
                 color = profile,
                 group = profile),
             linetype = "dashed")
```

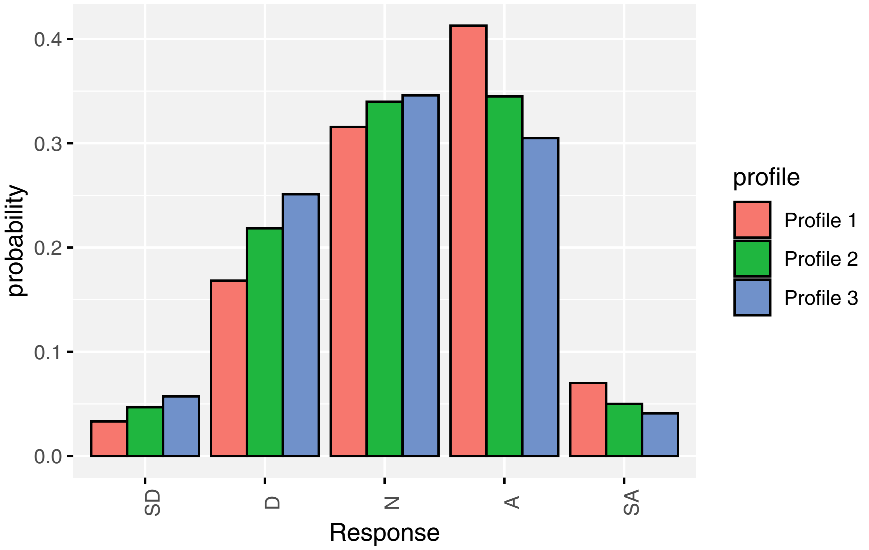
Inspection of the distribution for profile 3 reveals that the effect of living in a neighborhood with a median number of workers who commute by transit is to further shift the distribution to the left (due to the negative sign of the coefficient for `Rate_Public`). The magnitude of the shift is proportional to the magnitude of the coefficient, as follows:

```
mod_community_clm$coefficients["Rate_Public"] * median(mc_attitudes$Rate_Public)
```

```
Rate_Public
 -0.2107731
```

The probabilities of each response for the profiles are as seen in the following bar chart:

```r
profiles <- data.frame(profile = c("Profile 1",
                                   "Profile 2",
                                   "Profile 3"),
                       age = 21,
                       vehicle = c("No",
                                   "Yes",
                                   "Yes"),
                       Rate_Non_Canadian = 0,
                       Rate_Public = c(0,
                                       0,
                                       median(mc_attitudes$Rate_Public)))

profiles <- cbind(profiles,
                  predict(mod_community_clm,
                          newdata = profiles))

# Prepare data for plotting
profiles %>%
  select(-c(age, vehicle, Rate_Non_Canadian, Rate_Public)) %>%
  # Pivot longer to gather all probabilities in a single column
  pivot_longer(cols = -profile,
               names_to = "Response",
               values_to = "probability") %>%
  # Code the response in order
  mutate(Response = factor(Response,
                           levels = c("fit.STRONGLY DISAGREE",
                                      "fit.DISAGREE",
                                      "fit.NEUTRAL",
                                      "fit.AGREE",
                                      "fit.STRONGLY AGREE"),
                           labels = c("SD", "D", "N", "A", "SA"),
                           ordered = TRUE)) %>%
  # Plot
  ggplot() +
  geom_col(aes(x = Response,
               y = probability,
               fill = profile),
           color = "black",
           position = "dodge") +
  theme(axis.text.x = element_text(angle = 90))
```



11.10 Proportional Odds Property of the Ordinal Logistic Model

A property of the model introduced in the preceding section is that it gives *proportional odds*. Consider the probability that the response is less than or equal to a certain level of the categorical response (i.e., $P(Y \leq Y_j)$). As we saw, this was the area under the curve to the left of the corresponding threshold. In the ordinal logit model (assuming that the scale has been normalized):

$$P(Y \leq k_i) = \frac{e^{(\lambda_i - V_n)}}{1 + e^{(\lambda_i - V_n)}}$$

The odds ratio given this probability is as follows:

$$\frac{P(Y_n \leq k_i)}{1 - P(Y_n \leq k_i)} = \frac{\frac{e^{(\lambda_i - V_n)}}{1+e^{(\lambda_i - V_n)}}}{1 - \frac{e^{(\lambda_i - V_n)}}{1+e^{(\lambda_i - V_n)}}}$$

The expression above is the chances of $Y_n \leq k_i$ relative to $Y_n > k_i$. We can manipulate the expression above to obtain a simpler formulation of this odds ratio:

$$\frac{\frac{e^{(\lambda_i - V_n)}}{1+e^{(\lambda_i - V_n)}}}{\frac{1+e^{(\lambda_i - V_n)} - e^{(\lambda_i - V_n)}}{1+e^{(\lambda_i - V_n)}}} = e^{(\lambda_i - V_n)}$$

When $V_n = \lambda_j$ the odds ratio is $e^0 = 1$, i.e., the odds of $Yn \leq k_i$ are equal to the odds of $Yn > k_i$. As $V_n \ll \lambda_i$ $\lambda_i - V_n$ tends to a very large negative number, and therefore the odds ratio tends to zero (the chances of $Y_n < k_i$ tend to vanish). In contrast, when $V_n \gg \lambda_i$ $\lambda_i - V_n$ becomes an increasingly large positive number; accordingly, the odds ratio tends to grow.

To illustrate this effect, suppose that $V_n = \lambda_i + \kappa$, where κ is an arbitrary constant:

$$e^{(\lambda_i - (\lambda_i + \kappa))} = e^{\kappa}$$

Adding *any* value to the systematic utility is, as we have seen, equivalent to shifting the location of the distribution. When $\kappa < 0$ the odds ratio is less than 1, and when $\kappa > 0$ the odds ratio is greater than 1. Let $\kappa = 0.47$. In this case, the odds of $Y_n \leq k_i$ are approximately 1.6 times greater than the odds of $Y_n > k_i$:

```
exp(0.47)
```

```
[1] 1.599994
```

Somewhat surprisingly, as seen above, changes in the odds ratio do not really depend on λ_j (the threshold coefficient vanishes from the odds ratio when we think of it in terms of shifting the location of the distribution). This property is called *proportional odds*. Changes in the odds ratio are proportional to κ. This can be expressed as well in terms of the utilities. Suppose that we ask what is the change in the odds ratio of $Y \leq k_i$ between two individuals, say n and m, whose utilities differ by an amount $\kappa = V_m - V_n$. In this case:

$$\frac{e^{(\lambda_i - V_n)}}{e^{(\lambda_i - V_m)}} = e^{(\lambda_i - \lambda_i - V_n + V_m)} = e^{(V_m - V_n)} = e^{\kappa}$$

The change in the odds ratio is a constant irrespective of the level of the response. This property of the model is reminiscent of the Independence from Irrelevant Alternatives (IIA) property of the multinomial logit model. In the case of IIA, changes in the odds ratio were independent of any other alternatives in the choice set; in the present case, the odds ratio is independent of the level of the response. In other words, the ratio There are different modifications of the model that relax the proportional odds property, as discussed in the following section.

11.11 Non-proportional Odds Models

Proportional odds in the model discussed above impose a certain rigidity in the way the model can describe the data: the difference in odds ratios between two decision-makers is a constant no matter the level of the response. In other words, the effect of a variable (let us say car ownership in the example above) is the same between SD and D than between D and N.

Fortunately, there are a number of ways to circumvent this property of the model.

11.11.1 Parameterizing the Thresholds

A relatively simple way to relax the proportional odds property is by parameterizing the thresholds. As seen in the example about sense of community, the thresholds in the basic ordinal logit model are constants: while the distribution shifts location in response to the attributes in V_n, the thresholds remain in place. Suppose instead that the thresholds are allowed to vary as follows:

$$\lambda_i(W_n) = f(W_n)$$

where W_n is a vector of attributes that describe shifts in the thresholds. As an example, suppose that the thresholds are parameterized using the variable `gender`:

$$\lambda_i(gender) = \lambda_{i1} + \lambda_{i2}gender$$

This gives two sets of thresholds, λ_i^m for men and λ_i^w:

$$\begin{aligned}\lambda_i^m &= \lambda_{i1}\\ \lambda_i^w &= \lambda_{i1} + \lambda_{i2}\end{aligned}$$

The odds ratios of two decision-makers, woman (w) and man (m) now are only constant across responses j if the difference between thresholds is a constant:

$$\frac{e^{(\lambda_j^w - V_w)}}{e^{(\lambda_j^m - V_m)}} = e^{(\lambda_j^w - \lambda_j^m - V_w + V_m)}$$

This is implemented for our model for `community` in the next chunk of code (the argument `nominal = ~ gender` is the formula for the thresholds):

```
mod_community_flex_1 <- clm(Community ~ age + vehicle +
                              Rate_Non_Canadian + Rate_Public,
                            nominal = ~ gender,
                            data = mc_attitudes)

summary(mod_community_flex_1)
```

```
formula: Community ~ age + vehicle + Rate_Non_Canadian + Rate_Public
nominal: ~gender
data:    mc_attitudes
```

```
 link  threshold nobs logLik   AIC     niter max.grad cond.H 
 logit flexible  1230 -1668.20 3360.39 6(0)  3.42e-09 1.7e+05

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)   
age                0.03525    0.01177   2.995  0.00274 **
vehicleYes        -0.34498    0.11086  -3.112  0.00186 **
Rate_Non_Canadian -2.00618    0.81379  -2.465  0.01369 * 
Rate_Public       -1.85352    0.79094  -2.343  0.01911 * 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:
                                      Estimate Std. Error z value
STRONGLY DISAGREE|DISAGREE.(Intercept)  -2.7698     0.3568  -7.762
DISAGREE|NEUTRAL.(Intercept)            -0.5623     0.3028  -1.857
NEUTRAL|AGREE.(Intercept)                1.0044     0.3039   3.306
AGREE|STRONGLY AGREE.(Intercept)         3.4745     0.3743   9.283
STRONGLY DISAGREE|DISAGREE.genderWoman   0.2289     0.2633   0.869
DISAGREE|NEUTRAL.genderWoman            -0.1096     0.1295  -0.846
NEUTRAL|AGREE.genderWoman               -0.3005     0.1257  -2.390
AGREE|STRONGLY AGREE.genderWoman        -0.2192     0.2884  -0.760
```

The results now include threshold coefficients labeled as “(Intercept)”: those correspond to λ_{i1}. Then, there is a set of threshold coefficients associated with the variable `gender`: those correspond to λ_{i2}. The results indicate that the difference between thresholds is not a constant: the shift in the thresholds for women is in some cases negative and in some cases positive. We might ask of these coefficients whether they are significantly different from zero: if none of them are, the model would collapse to the proportional odds model, since $\lambda_i^w = \lambda_{i1} + \lambda_{i2}$ collapses to $\lambda_i^m = \lambda_{i1}$ if $\lambda_{i2} = 0$ (in this example, we see that the only threshold for women that is significantly different from zero is for “NEUTRAL|AGREE”. Figure 11.3 illustrates the results of the model. Since gender is not a variable in V_n, the location of the distribution does not shift with this attribute, but the thresholds do, one of them significantly so.

```r
# Coefficients of the utility function
beta_n <- mod_community_flex_l$coefficients[9:12]

# Thresholds
l_j <- mod_community_flex_l$coefficients[1:8]

# Profile 1: Age = 21, No vehicle, other variables at zero
p1 <- c(21, 0, 0, 0)

# Calculate the utility of these two hypothetical individuals
V_1 <- beta_n %*% p1

# Calculate the values of the logistic distribution for profile 1 in the range defined by x
dV_1 <- data.frame(x = rep(seq(-3.2, 4.7, 0.1))) %>%
  mutate(f = dlogis(x,
                    location = V_1))

ggplot() +
  # Plot the distribution
  geom_area(data = dV_1,
            aes(x = x,
                y = f),
            color = "black",
            fill = "orange",
            alpha = 0.5) +
  # Plot the thresholds
  geom_vline(data = data.frame(profile = rep(c("Man", "Woman"), each = 4),
                               l_j = c(l_j[1:4], l_j[1:4] + l_j[5:8])),
             aes(xintercept = l_j,
                 color = profile)) +
  scale_x_continuous(name = "x",
                     breaks = c(l_j[1:4], l_j[1:4] + l_j[5:8]),
                     labels = expression(lambda[1][m],
                                         lambda[2][m],
                                         lambda[3][m],
                                         lambda[4][m],
                                         lambda[1][w],
                                         lambda[2][w],
                                         lambda[3][w],
                                         lambda[4][w])) +
  theme(axis.text.x = element_text(angle = 90))
```

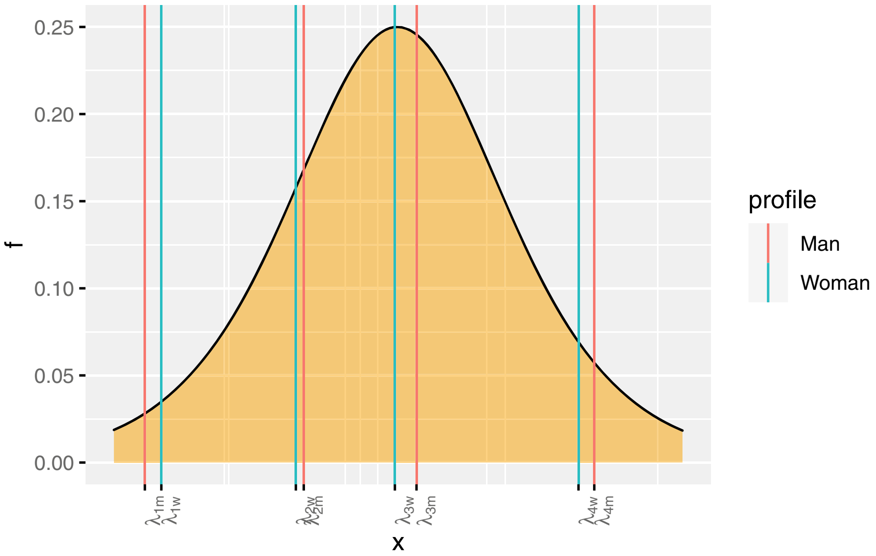


Fig. 11.3 Model with parameterized thresholds for two versions of hypothetical profile 1: woman and man

11.11.2 Parameterizing the Scale

Previously we said that the scale could be normalized without affecting the order of the utilities. It is also true that the variance can be parameterized to allow for non-constant scale. In general terms, this is done as follows:

$$\sigma(S_n) = \sigma_n = e^{\eta S_n}$$

where S_n is a vector of attributes that describe variations in the scale. The functional form is an exponential to ensure that the scale is strictly positive. Suppose for example that we parameterize the scale using the `visa` status of students (i.e., domestic and international) as follows:

$$\sigma_n = e^{\eta_1 visa}$$

If `visa` is 0 when the student is domestic (D) and 1 when the student is international (I), we can see that

$$\begin{aligned}\sigma_D &= 1\\ \sigma_I &= e^{\eta_1}\end{aligned}$$

This model can be estimated using function `clm()` in package {ordinal} by using the argument `scale`:

```
mod_community_ncs <- clm(Community ~ age + vehicle +
                              Rate_Non_Canadian +
                              Rate_Public,
                            scale = ~ visa,
                            data = mc_attitudes)

summary(mod_community_ncs)
```

```
formula: Community ~ age + vehicle + Rate_Non_Canadian + Rate_Public
scale:   ~visa
data:    mc_attitudes

 link  threshold nobs logLik   AIC     niter max.grad cond.H
 logit flexible  1230 -1670.31 3358.63 10(0) 1.27e-12 1.7e+05

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
age                0.03303    0.01155   2.859  0.00424 **
vehicleYes        -0.35461    0.10872  -3.262  0.00111 **
Rate_Non_Canadian -1.92234    0.79908  -2.406  0.01614 *
Rate_Public       -1.68778    0.77931  -2.166  0.03033 *
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
log-scale coefficients:
                  Estimate Std. Error z value Pr(>|z|)
visaInternational  -0.2702     0.1452  -1.861   0.0627 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Threshold coefficients:
                          Estimate Std. Error z value
STRONGLY DISAGREE|DISAGREE -2.6268     0.3020  -8.698
DISAGREE|NEUTRAL           -0.6505     0.2819  -2.308
NEUTRAL|AGREE               0.7782     0.2813   2.766
AGREE|STRONGLY AGREE        3.2781     0.3103  10.564
```

In this example, we can reject the null hypothesis that $\sigma_I = \sigma_D = 1$ with $p < 0.1$. A log-scale coefficient converts into the scale parameter as follows:

```
exp(mod_community_ncs$coefficients[9])
```

```
visaInternational
        0.7632213
```

where we see that the scale of the distribution of domestic students is less than 1. To illustrate the results of the model we simulate another profile (profile 4) by setting age to 21, vehicle to "No", rate of non-Canadian residents in the neighborhood to the in-sample media, and rate of public transportation to zero. In Fig. 11.4 we see

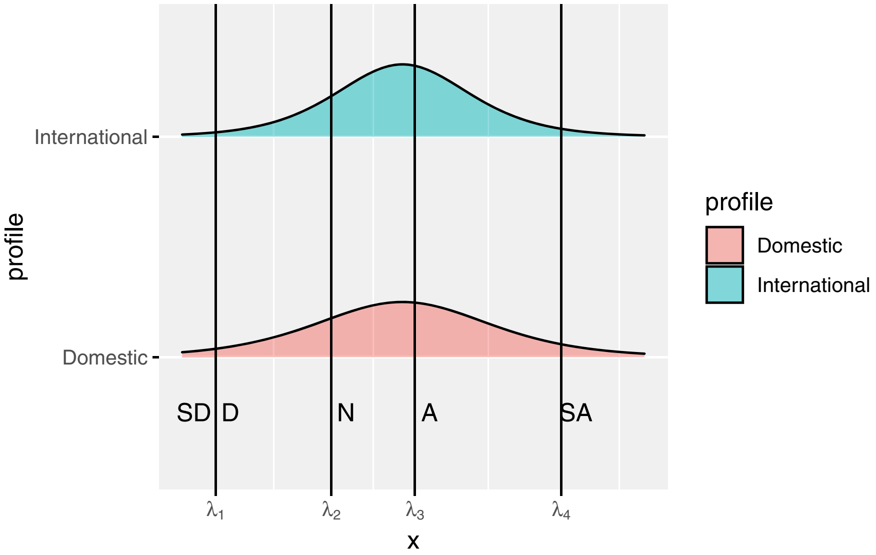


Fig. 11.4 Model with parameterized scale for two versions of hypothetical profile 4: domestic and international student

that the distribution is more compact for international students, and as a consequence their probability of responding more strongly to the attitudinal statement is lower.

```r
# Coefficients of the utility function
beta_n <- mod_community_ncs$coefficients[5:8]

# Thresholds
l_j <- mod_community_ncs$coefficients[1:4]

# Scale
s_I <- exp(mod_community_ncs$coefficients[9])

# Profile 4: Age = 21, No vehicle, other variables at zero
# proportion of non-Canadian residents is the in-sample median
p1 <- c(21, 0, median(mc_attitudes$Rate_Non_Canadian), 0)

# Calculate the utility of these two hypothetical individuals
V_1 <- beta_n %*% p1

# Calculate the values of the logistic distribution for profile 1
# when student is Domestic
dV_D <- data.frame(x = rep(seq(-3.2, 4.7, 0.1)),
                   profile = "Domestic") %>%
  mutate(f = dlogis(x,
                    location = V_1))

# Calculate the values of the logistic distribution for profile 1
# when student is International
dV_I <- data.frame(x = rep(seq(-3.2, 4.7, 0.1)),
                   profile = "International") %>%
  mutate(f = dlogis(x,
                    location = V_1,
                    scale = s_I))

# Bind the distributions for the two profiles
dV <- rbind(dV_D,
            dV_I)

# Render the plot and add the distributions
ggplot() +
  # Plot the distributions as ridges
  geom_ridgeline(data = dV,
                 aes(x = x,
                     y = profile,
                     fill = profile,
                     group = profile,
                     height = f),
                 alpha = 0.5) +
  # Plot the thresholds
  geom_vline(xintercept = mod_community_ncs$coefficients[1:4]) +
  # Label the responses
  geom_text(aes(label = c("SD", "D", "N", "A", "SA"),
                x = c(-3, l_j + 0.25)),
            y = 0.75) +
  # Label the ticks in the x axis with the threshold coefficients
  scale_x_continuous(name = "x",
                     breaks = l_j,
                     labels = expression(lambda[1],
                                         lambda[2],
                                         lambda[3],
                                         lambda[4]))
```

11.12 Multivariate Ordinal Models

An interesting situation arises when the analyst has multiple ordinal variables that they wish to analyze as joint outcomes. For example, the analyst might be interested in the number of out-of-home activities by household heads, while considering the interactions between them (e.g., Scott and Kanaroglou 2002). In this case, the ordinal outcomes might not be independent due to sharing of responsibilities: one of the household heads might generate fewer episodes if the total mobility needs for the household are compensated by an increased number of episodes by the other household head. A similar situation arises in the research of Benedini et al. (2020), who investigated the number of trips by bicycle for different purposes (i.e., work and non-work). The ordinal outcomes (i.e., trip-making intensity by purpose) are likely not independent simply by the nature of the phenomenon: more work-related trips leave less time for more non-work-related trips.

In cases like these the outcomes might not be independent, and therefore it might be more appropriate to model them jointly. Ordinal models with multiple outcomes are called *multivariate ordinal models*.

In the general case, suppose that there are O ordinal outcomes of interest:

$$\begin{gathered} Y_{n1} \\ \dots \\ Y_{no} \\ \dots \\ Y_{nO} \end{gathered}$$

Each of these outcomes can be modeled using the ordinal framework introduced in this chapter:

$$Y_{no} = k_{oi} \leftrightarrow \lambda_{o,i-1} < U_{no} \leq \lambda_{o,i}$$

In this formulation $\lambda_{o,0} \equiv -\infty$ and $\lambda_{o,J_o+1} \equiv \infty$ (J is the number of categorical levels of outcome o). The models are derived following the same logic of models for univariate ordinal responses. However, the models are tied by a correlation structure that depends on the distribution of the random utility. Normally distributed random utilities are naturally accommodated in a multivariate normal distribution. In the case of multivariate logistic ordinal models, a multivariate logit link results from the univariate logistic margins with a copula t with a pre-specified number of degrees of freedom (see Hirk et al. 2020).

To return the example at hand, it is reasonable to expect that a person who agrees that there is a sense of community in their neighborhood is also likely to agree that they know their neighbors well. We have two ordinal responses that are modeled as follows:

$$\begin{aligned} Y_{n,Community} &= k_{Community,i} \leftrightarrow \lambda_{Community,i-1} < U_{n,Community} \leq \lambda_{Community,i} \\ Y_{n,Neighbors} &= k_{Neighbors,i} \leftrightarrow \lambda_{Neighbors,i-1} < U_{n,Neighbors} \leq \lambda_{Neighnors,i} \end{aligned}$$

with

$$\begin{aligned} V_{n,Community} &= f(age_n, vehicle_n, Rate\_Non\_Canadian_n, Rate\_Public_n) \\ V_{n,Neighbors} &= f(age_n, vehicle_n, Rate\_Non\_Canadian_n, Rate\_Public_n) \end{aligned}$$

Multivariate ordinal models are implemented in package {mvord}. The relevant function to estimate models of this kind is `mvord()`.

The following chunk of code shows the code to estimate a bivariate model of variables `Community` and `Neighbors`. The multiple ordinal outcomes are specified using `MMO2()`. The constant is set to zero (recall that the thresholds act as constant terms in ordinal models). Notice that the link function in this example is for a probit model:

```
mod_bivariate <- mvord(formula = MMO2(Community, Neighbors) ~ 0 + age + vehicle +
                           Rate_Non_Canadian +
                           Rate_Public,
                         link = mvlogit(df = 8L),
                         # {mvord} does not like tbl or tbl_df objects:
                         # convert to plain data.frame
                         data = data.frame(mc_attitudes))
summary(mod_bivariate)
```

```
Call: mvord(formula = MMO2(Community, Neighbors) ~ 0 + age + vehicle +
    Rate_Non_Canadian + Rate_Public, data = data.frame(mc_attitudes),
    link = mvlogit(df = 8L))

Formula: MMO2(Community, Neighbors) ~ 0 + age + vehicle + Rate_Non_Canadian +
    Rate_Public

   link threshold nsubjects ndim    logPL   CLAIC   CLBIC fevals
mvlogit  flexible      1230    2 -3239.52 6513.51 6601.68    839

Thresholds:
                                     Estimate Std. Error z value  Pr(>|z|)
Community STRONGLY DISAGREE|DISAGREE -2.69135    0.29322 -9.1786 < 2.2e-16 ***
Community DISAGREE|NEUTRAL           -0.69339    0.26648 -2.6021  0.009266 **
Community NEUTRAL|AGREE               0.74247    0.26517  2.8000  0.005110 **
Community AGREE|STRONGLY AGREE        3.27354    0.29658 11.0378 < 2.2e-16 ***
```

```
Neighbors STRONGLY DISAGREE|DISAGREE -1.55617    0.26772 -5.8126 6.150e-09 ***
Neighbors DISAGREE|NEUTRAL            0.20525    0.26098  0.7865  0.431597
Neighbors NEUTRAL|AGREE               1.29728    0.26246  4.9429 7.699e-07 ***
Neighbors AGREE|STRONGLY AGREE        3.40117    0.30386 11.1931 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Coefficients:
                      Estimate Std. Error z value  Pr(>|z|)
age 1                 0.032620   0.010701  3.0484 0.0023010 **
age 2                 0.044504   0.010722  4.1507 3.314e-05 ***
vehicleYes 1         -0.358602   0.112166 -3.1971 0.0013884 **
vehicleYes 2         -0.657084   0.113304 -5.7993 6.659e-09 ***
Rate_Non_Canadian 1 -2.083539   0.811324 -2.5681 0.0102266 *
Rate_Non_Canadian 2 -2.973022   0.851910 -3.4898 0.0004833 ***
Rate_Public 1        -1.867729   0.774509 -2.4115 0.0158870 *
Rate_Public 2        -3.716425   0.793642 -4.6827 2.831e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error Structure:
                          Estimate Std. Error z value  Pr(>|z|)
corr Community Neighbors 0.530169   0.025642  20.676 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Of note in this case is the correlation between the ordinal outcomes `Community` and `Neighbors`. The *p*-value of this parameter is sufficiently low that we can reject the null hypothesis that the correlation between the outcomes is zero. Had that been the case, it would be more efficient to re-estimate the models independently as univariate ordinal outcomes instead of jointly.

The coefficients for each outcome are interpreted similarly to those of a univariate ordinal model. It is possible to calculate the *joint probabilities* of the outcomes too. These are the probabilities of jointly observing outcomes $k_{i,Community}$ and $k_{j,Neighbors}$. As an example, the joint probabilities of an individual in the sample can be calculated by creating a fitting grid with all combinations of outcomes for `Community` and `Neighbors`, and the attributes of the chosen individual:

```r
# Select an individual in the sample to fit the probabilities
n_ind <- 1

# Create a grid of values for chosen individual
fit_grid_ind <- expand.grid(Community = c("STRONGLY DISAGREE",
                                          "DISAGREE",
                                          "NEUTRAL",
                                          "AGREE",
                                          "STRONGLY AGREE"),
                            Neighbors = c("STRONGLY DISAGREE",
                                          "DISAGREE",
                                          "NEUTRAL",
                                          "AGREE",
                                          "STRONGLY AGREE"),
                            # Retrieve the attributes of the n_ind record
                            # in the table
                            age = mc_attitudes$age[n_ind],
                            vehicle = mc_attitudes$vehicle[n_ind],
                            Rate_Non_Canadian = mc_attitudes$Rate_Non_Canadian[n_ind],
                            Rate_Public = mc_attitudes$Rate_Public[n_ind])
```

Table 11.1 Fitted joint probabilities for in-sample individual

Community	Neighbors_SD	Neighbors_D	Neighbors_N	Neighbors_A	Neighbors_SA
Community_SD	0.01705	0.01234	0.00419	0.00238	0.00039
Community_D	0.03543	0.07788	0.04166	0.02384	0.00246
Community_N	0.02295	0.10683	0.10197	0.08130	0.00831
Community_A	0.01060	0.06836	0.11016	0.17266	0.03552
Community_SA	0.00090	0.00422	0.00831	0.02820	0.02210

Table 11.1 shows the fitted joint probabilities for each combination of responses for the selected in-sample record:

```
# Join the prediction grid to the predicted probabilities
joint.probs <- data.frame(fit_grid_ind %>%
                            select(Community, Neighbors),
                          # Use `predict()` and the prediction grid
                          # to predict joint probabilities
                          joint.prob = predict(mod_bivariate,
                                               type = "prob",
                                               newdata = fit_grid_ind)) %>%
  # Revalue the ordinal responses for ease of presentation
  mutate(Community = revalue(Community,
                             c("STRONGLY DISAGREE" = "Community_SD",
                               "DISAGREE" = "Community_D",
                               "NEUTRAL" = "Community_N",
                               "AGREE" = "Community_A",
                               "STRONGLY AGREE" = "Community_SA")),
         Neighbors = revalue(Neighbors,
                             c("STRONGLY DISAGREE" = "Neighbors_SD",
                               "DISAGREE" = "Neighbors_D",
                               "NEUTRAL" = "Neighbors_N",
                               "AGREE" = "Neighbors_A",
                               "STRONGLY AGREE" = "Neighbors_SA"))) %>%
  pivot_wider(names_from = Neighbors, values_from = joint.prob)

joint.probs %>%
  kable("latex",
        digits = 5,
        caption = "\\label{tab:joint-probs-in-sample-individuals}
        Fitted joint probabilities for in-sample individual")
```

We can verify that the joint probabilities add up to 1:

```
joint.probs %>%
  select(-Community) %>%
  sum()
```

```
[1] 1
```

As seen in Table 11.1, the fitted probabilities indicate that the chosen individual tends to agree with both attitudinal statements: that there is a sense of community in their neighborhood and that they know their neighbors well. The joint probabilities of disagreeing or strongly disagreeing with these statements are considerably lower. And the lowest joint probabilities are for dissonant responses, that is, those were the respondent might agree with one statement but disagree with the other.

We can learn more about the underlying preferences towards the various responses by looking at the effect of the attributes. This is similar to the computing the marginal effects: how do the joint probabilities change when the values of the attributes change? To explore this question we proceed to create a prediction grid with all combinations of the ordinal responses, two different ages two different vehicle ownership status, and the Rate_Non_Canadian and Rate_Public set to the in-sample median. This gives four different profiles that vary by age and vehicle availability:

```
pred_grid <- expand.grid(Community = c("STRONGLY DISAGREE",
                                       "DISAGREE",
                                       "NEUTRAL",
                                       "AGREE",
                                       "STRONGLY AGREE"),
                         Neighbors = c("STRONGLY DISAGREE",
                                       "DISAGREE",
                                       "NEUTRAL",
                                       "AGREE",
                                       "STRONGLY AGREE"),
                         # Age = 20 is the value for the first quartile in the sample
                         # and age = 23 is the value for the third quartile
                         age = c(20, 23),
                         vehicle = levels(mc_attitudes$vehicle),
                         Rate_Non_Canadian = median(mc_attitudes$Rate_Non_Canadian),
                         Rate_Public = median(mc_attitudes$Rate_Public))
```

The prediction grid is used as an input in conjunction with the model to compute the joint probabilities for each pair of ordinal outcomes at each level of the independent variables:

```
# Create a data frame with the prediction grid and
# the predicted joint probabilities
pred_grid <- data.frame(pred_grid,
                        joint.prob = predict(mod_bivariate,
                                             type = "prob",
                                             newdata = pred_grid)) %>%
  # Convert to factors and revalue for presentation
  mutate(age = factor(age,
                      levels = c(20, 23),
                      labels = c("Age = 20",
                                 "Age = 23")),
         vehicle = revalue(vehicle,
                           c("Yes" = "Vehicle",
                             "No" = "No Vehicle")),
         Community = revalue(Community,
                             c("STRONGLY DISAGREE" = "SD",
                               "DISAGREE" = "D",
                               "NEUTRAL" = "N",
                               "AGREE" = "A",
                               "STRONGLY AGREE" = "SA")),
         Neighbors = revalue(Neighbors,
                             c("STRONGLY DISAGREE" = "SD",
                               "DISAGREE" = "D",
                               "NEUTRAL" = "N",
                               "AGREE" = "A",
                               "STRONGLY AGREE" = "SA")))
```

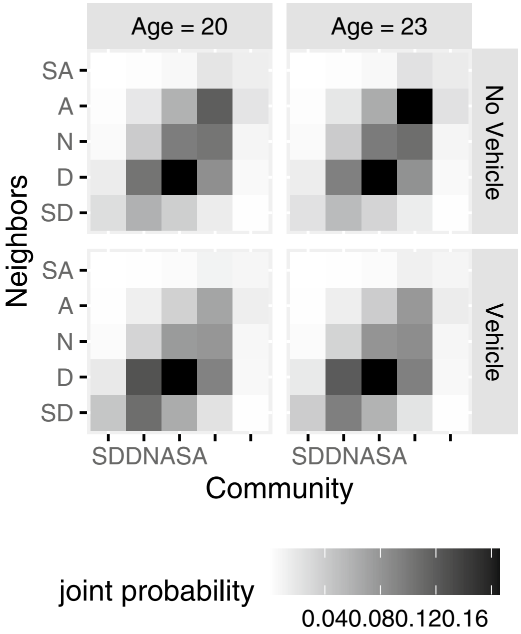
At this point, verifying that the joint probabilities for each profile add up to one is an appropriate sanity check:

```
pred_grid %>%
  group_by(age, vehicle) %>%
  summarize(prob = sum(joint.prob),
            .groups = "drop")
```

```
  prob .groups
1    4    drop
```

Tile plots in conjunction with faceting (an exploratory visualization technique discussed in Chap. 2) are useful to inspect the joint probabilities of the four profiles we created above:

```
pred_grid %>%
  ggplot(aes(x = Community, y = Neighbors)) +
  geom_tile(aes(fill = joint.prob)) +
  scale_fill_gradient(name = "joint probability",
                      low="white",
                      high="black") +
  coord_equal() +
  theme(legend.position = "bottom") +
  facet_grid(vehicle ~ age)
```



It is possible to see that responses tend to be consonant: respondents tend to jointly agree that there is a sense of community in their neighborhood *and* that they know their neighbors well, or not. We do notice that there is a marked tendency to disagree with the statements when the individual has access to a vehicle (the sign of the coefficient for this variable is negative in both equations). As well, there appears to be a somewhat fainter preference for agreeing with the statements in the case of the profile with `age` set to 23 (the sign of this coefficient is positive in both equations). We can more easily see the effect of age and access to a vehicle by inspecting the odds ratios.

Figure 11.5 plots the odds ratios for the two different age profiles. We see that at a given age, the probability of jointly agreeing that there is a sense of community and strongly agreeing that the individual knows the neighbors well of an individual *without* individual access to a vehicle is almost two times that of an individual with individual access to a vehicle. Individuals *without* access to a vehicle are only half as likely to jointly disagree that there is a sense of community in the neighborhood and that they know their neighbors well compared to individuals with access to a vehicle.

```
pred_grid %>%
  # Pivot the table to create columns of joint probabilities
  # for `No Vehicle` and `Vehicle`
  pivot_wider(names_from = vehicle,
              values_from = joint.prob) %>%
  # Calculate the ratio of the joint probability for the two
  # profiles
```

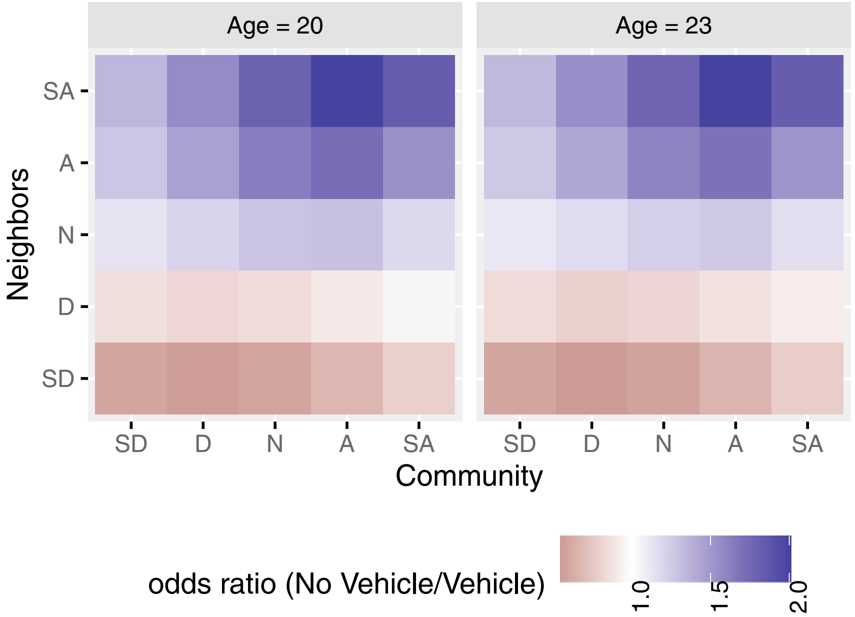


Fig. 11.5 Plot of odds ratios by age for No Vehicle/Vehicle status

```
  mutate(or = `No Vehicle`/Vehicle) %>%
  # Plot as tiles
  ggplot(aes(x = Community,
             y = Neighbors)) +
  geom_tile(aes(fill = or)) +
  # Use a divergent fill scale with midpoint 1
  scale_fill_gradient2(name = "odds ratio (No Vehicle/Vehicle)",
                       midpoint = 1) +
  coord_equal() +
  theme(legend.position = "bottom",
        legend.text = element_text(angle = 90)) +
  # Facet by age
  facet_grid(~ age)
```

A similar plot can be seen in Fig. 11.6 where the odds ratios are for the two different vehicle access profiles when age changes. As expected, older individuals are more likely to agree/strongly agree that there is a sense of community and that they know their neighbors well compared to younger individuals (approximately 15% more likely, according to the odds ratios). In contrast, older individuals are about 10–15% less likely to disagree/strongly disagree with the attitudinal statements.

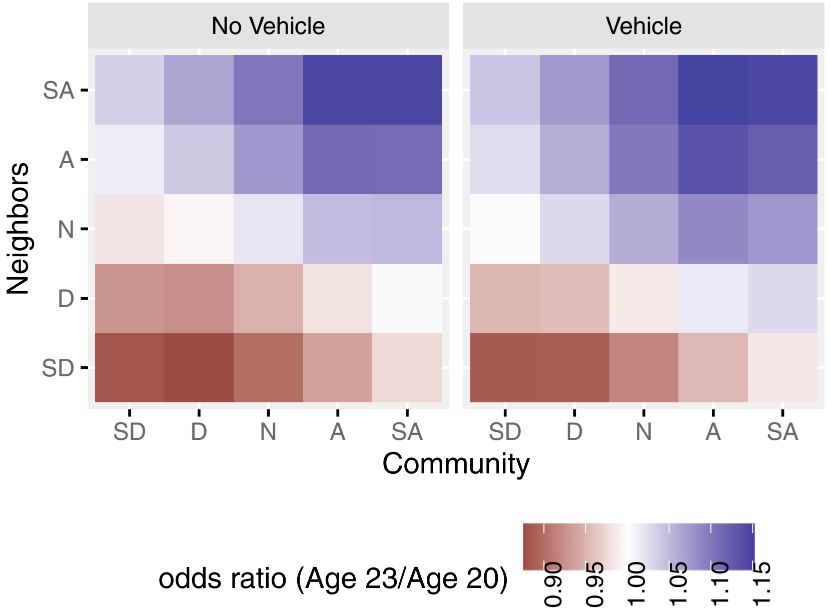


Fig. 11.6 Plot of odds ratios by age for Age 23/Age 20

```
pred_grid %>%
  pivot_wider(names_from = age, values_from = joint.prob) %>%
  mutate(or = `Age = 23`/`Age = 20`) %>%
  ggplot(aes(x = Community, y = Neighbors)) +
  geom_tile(aes(fill = or)) +
  scale_fill_gradient2(name = "odds ratio (Age 23/Age 20)",
                       midpoint = 1) +
  coord_equal() +
  theme(legend.position = "bottom",
        legend.text = element_text(angle = 90)) +
  facet_grid(~ vehicle)
```

It is important to remark here that the multivariate ordinal model discussed here displays the proportional odds property of univariate ordinal models. As in the univariate case, multivariate ordinal models have been extended into more sophisticated technical frameworks, which include among other possibilities parameterizing the correlations between ordinal responses. The reader is encouraged to consult Hirk et al. (2020) for technical details and implementation in package {mvord}.

11.13 Exercise

1. Why is it important to distinguish between non-ordinal and ordinal categorical variables in the analysis of discrete choices?
2. When parameterizing the thresholds λ_i, is it possible to use variables that are part of the systematic utility V_n? Explain.
3. When parameterizing the scale σ, is it possible to use variables that are part of the systematic utility V_n? Explain.
4. Show the relationships of the model with parameterized scale to the proportional odds property.
5. Load the following data set from the `discrtr` package:

```
data("mc_modality",
     package = "discrtr")
```

This data set was used by Lavery et al. (2013) to investigate the *modality* of students, staff, and faculty traveling to McMaster University. The modality is the number of modes of transportation that respondents recognize as feasible alternatives for travel to school (this variable is `modality` in the table).

Model the modality using an ordinal logistic model. Justify your modeling decisions. Is a model with parameterized thresholds and/or scale justified? Discuss.

References

Benedini, D. J., Lavieri, P. S., & Strambi, O. (2020). Understanding the use of private and shared bicycles in large emerging cities: The case of Sao Paulo, Brazil. *Case Studies on Transport Policy, 8*(2), 564–575. https://doi.org/10.1016/j.cstp.2019.11.009.

Bhat, C. R., & Pulugurta, V. (1998). A comparison of two alternative behavioral choice mechanisms for household auto ownership decisions. *Transportation Research Part B, 32*(1), 61–75.

Grannis, R. (2009). *From the ground up: Translating geography into community through neighbor networks*. Princeton: Princeton University Press.

Hirk, R., Hornik, K., & Vana, L. (2020). Mvord: An R package for fitting multivariate ordinal regression models. *Journal of Statistical Software, 93*(4), 1–41. https://doi.org/10.18637/jss.v093.i04.

Lavery, T. A., Páez, Antonio, & Kanaroglou, Pavlos S. (2013). Driving out of choices: An investigation of transport modality in a university sample. *Transportation Research Part A: Policy and Practice, 57,* 37–46.

McMillan, David W., & Chavis, David M. (1986). Sense of community: A definition and theory. *Journal of Community Psychology, 14*(1), 6–23.

Rogers, G. O., & Sukolratanametee, S. (2009). Neighborhood design and sense of community: Comparing suburban neighborhoods in Houston Texas. *Landscape and Urban Planning, 92*(3–4), 325–34. https://doi.org/10.1016/j.landurbplan.2009.05.019.

Scott, D. M., & Kanaroglou, P. S. (2002). An activity-episode generation model that captures interactions between household heads: Development and empirical analysis. *Transportation Research Part B-Methodological, 36*(10), 875–896. https://doi.org/10.1016/S0191-2615(01)00039-X.

Talen, E. (1999). Sense of community and neighbourhood form: An assessment of the social doctrine of new urbanism. *Urban Studies, 36*(8), 1361–1379. <Go to ISI>://000081660600007.

Whalen, K. E., Paez, A., Bhat, C., Moniruzzaman, M., & Paleti, R. (2012). T-communities and sense of community in a university town: Evidence from a student sample using a spatial ordered-response model. *Urban Studies, 49*(6), 1357–1376. https://doi.org/10.1177/0042098011411942.

Epilogue

We don't receive wisdom; we must discover it for ourselves after a journey that no one can take for us or spare us.

— Marcel Proust

One's destination is never a place, but rather a new way of looking at things.

— Henry Miller

Models, Models, Models

We started this book with a fundamental premise in mind: that models are useful tools that help us understand interesting aspects of the world. The technique and art of working with models is to know what and how to simplify. Simplification, of course, means that by definition models are always wrong, in the sense that they never fully capture the thing being modeled. It is precisely this weakness of models which lends them their strength: by concentrating on a limited number of possibly many facets of a phenomenon, models bring reality to the level of our powers of cognition. René Magritte's delightful pictorial paradox that simultaneously is a pipe and not a pipe, is a powerful illustration of the nature of models.[1] The aim of a modeler, therefore, is to capture the *essence* of something, be it to please the senses or to clarify a problem, but ultimately with the objective of enlightening the mind. Irrespective of whether a model is physical (a sculpture) or conceptual (a mental map), creating models requires three fundamental ingredients: materials, tools, and

[1] https://en.wikipedia.org/wiki/The_Treachery_of_Images.

A. Páez and G. Boisjoly, *Discrete Choice Analysis with R*, Use R!,
https://doi.org/10.1007/978-3-031-20719-8

technical expertise. In this book we aimed to cover all three in reference to modeling one particular phenomenon: individual decision-making.

There are numerous models that explain various aspects of decision-making. The focus of this book has been on discrete choice analysis, a family of techniques derived from concepts that emerged in micro-economics and that have proven useful in numerous fields and applications. At the core of discrete choice analysis are *models*, and our aim has been to cover the three complementary ingredients of modeling in an intuitive manner, supported with the use of numerous examples, extensive use of coding, and data. In this epilogue we would like to briefly recap a journey of discovery, one that, if you have worked on the examples, no one took for you but yourself.

The Ingredients of Modeling

Discrete choice models are mathematical and statistical in nature. The *materials* come in the form of data. Models in this family of techniques are about decisions that concern distinct—indivisible and mutually exclusive—alternatives. Accordingly, the data comprise at least one discrete variable that represents individual choices. Further, alternatives can be, and usually are, characterized by a set of attributes (for example, the speed and cost of smartphones as in the example introduced in Chap. 3). These attributes are bundled together and therefore individuals cannot pick and choose attributes from different alternatives, say the null price of the old phone together with the high speed of the new phone. They have to select a single alternative with all its attributes, many of whom are not explicitly priced. This is what characterizes a discrete choice. Since most of us are usually exposed to quantitative data analysis, becoming familiar with the nature of discrete data is a key prerequisite to modeling discrete choices. Certainly, the quality of a model depends on the quality of the materials used, as well as how well the modeler understands them. This is why the book began with the fundamentals of data in Chap. 1 and exploratory data analysis in Chap. 2.

The *tools* for discrete choice analysis consist first of the conceptual and mathematical frameworks that generations of experts have spent time developing. These frameworks offer us an approach to representing reality in a simplified manner and builds on our evolving understanding of behavioral mechanisms. Representing complex phenomena with "simple" mathematical models requires that we make assumptions that simplify and generalize. It is often possible to relax our assumptions, although this usually comes at a price, for instance linked to computational challenges and the utility of a model to give meaningful behavioral interpretations. Ignoring these assumptions can lead to models that are inappropriate, and therefore not helpful or worse, deceiving. Understanding the theoretical framework and how it is applied is, therefore, key to generating relevant models. In our view, this is where computational tools, in our case the `R` language and associated packages, are essential. Computational tools allow a modeler to implement the conceptual and mathematical frameworks underlying discrete choice analysis at a speed and scale that would otherwise be prohibitive. Equipped in this way, a modeler can easily experiment with tangible examples to develop a gut feeling for the behavior of models and the underlying mechanisms of interest.

Having developed skills with the data and the tools, a modeler acquires the last ingredient of modelling: *technical expertise*. With today's powerful computers and software tools, running a model in `R` (or any other language for that matter) is relatively simple. Anyone with some programming literacy and internet search skills can run models. However, without an understanding of the underlying fundamental concepts and assumptions, such tools can be misused and lead to confusion and frustration at best, or to gleeful mistakes at worse. This book, while covering only a sample of models and behaviors, has aimed to provide an understanding of the fundamental concepts and tools behind discrete choice analysis. In doing so, we aspired to equip the reader with a sharp understanding of the power, limits, and implications of models, as well as the modeling tools and assumptions that go into model construction, specification, and estimation.

It is important to remember that model building is an *art*. The three ingredients mentioned before are complementary, and a good, that is, a *useful* model, can only be obtained with appropriate data and a skilled modeler who understands their data and uses the appropriate tools. As with any art, there are guidelines but typically no strict rules about how the craft of modelling. But just as a brush is not an appropriate tool for sculpting marble, it is only by understanding the tools that a modeler makes informed decisions to develop models that are appropriate for a given application.

From Fundamental Concepts to Sophisticated Models

After introducing key ideas about models, data, and exploratory data analysis, the book delved into the fundamental concepts of discrete choice analysis in Chap. 3. A first simplification of reality in discrete choice analysis relates to the behavioral mechanism used by people who make choices—in other words, the decision rule(s) used by individuals to decide between various discrete alternatives. The framework used, called *utility maximization*, is strongly rooted in neoclassical economics, in particular a micro-economic theory of consumer behavior. This theory posits that individuals select the alternatives that provide them with the greatest utility. Utility here refers to a summary indicator of the pleasure, enjoyment, or attractiveness associated with an alternative. This framework also assumes that individuals make choices in a rational manner, meaning in a consistent fashion: when presented with the same choice situation, they consistently choose the same alternative. Therefore, within this framework, choices are all about rational trade-offs. Individuals weigh the attributes of the different alternatives against one another, and pick the alternative with the overall combination of attributes that gives them the greatest utility.

While individuals are assumed to act in a rational manner and always select the alternative that provides them with the greatest utility, it is impossible for the modeler to observe all elements of the alternatives that enter into the decision-making process. To deal with this limitation, the utility is decomposed into two components: the *systematic utility* and the *random utility*. The systematic utility corresponds to the elements that can be observed. The random utility reflects the unknown elements—or uncertainties - associated with the choices. The random utility component is the foundation of random utility modelling: this part of the utility function is at the core of the probabilistic statements used to derive discrete choice models. Before a model can be derived, however, we must make some assumptions about the distribution of the random utility.

Accordingly, a second simplification used to develop probabilistic choice models relates to the distribution of the random utility. As mentioned above, assumptions regarding the random utility are necessary to obtain operational models. A first model, the *binary logit* model, was introduced in Chap. 4 as the workhorse of discrete choice analysis. This model assumes a specific distribution of the random utility (the Extreme Value Type 1–EV Type I). Why this distribution? The EV Type I distribution is chosen chiefly for practical reasons. The difference between two random utilities assumed to follow the EV Type I distribution (which is central to the calculation of probabilities) follows the logistic distribution. This distribution is convenient for estimation purposes because its integral has an analytical solution, or a *closed form*. The EV Type I distribution is also a probable distribution of the random utility in terms of behavior of a population, being somewhat similar to the normal distribution (albeit with fatter tails). Building on these foundations, Chap. 4 presented the logit probability function of the binary logit models, and then expands it to the multinomial logit.

After familiarizing ourselves with the framework of the logit model, we proceeded to use it in Chaps. 5 and 6 to discuss some important practical aspects of discrete choice modelling, many of which are transferable to other discrete choice models. The *specification and estimation* of logit models were addressed in Chap. 5, whereas Chap. 6 discussed examples of *behavioral insights* that can be gained from such models. As discrete choice analysis is anchored in a strong theoretical foundations, these chapters illustrate how discrete choice analysis can be used to derive meaningful behavioral interpretations and to forecast the implications of changes in the system as individuals adapt their behavior. In this way, Chaps. 4–6 provide the basis to work with the logit model and its potential for applications.

The ease of implementation of the logit model reflects the idealized conditions used to derive it. A consequence of these conditions is the axiom known as *Independence of Irrelevant Alternatives (IIA)*. This axiom basically states that a choice between two alternatives is independent of any other alternatives in the choice set. While this property is potentially useful, it only is appropriate when a model is fully specified, meaning that there are no hidden correlations among utility functions. A manifestation of IIA is patterns of substitution that are proportional. Alas, as the red bus-blue bus paradox, this property can be a liability when the utility functions have unaccounted-for correlations. The EV Type I distribution, with its convenient and simple closed form, does not accommodate correlations. To allow for more flexibility, several more sophisticated models, building on the same fundamental concepts of random utility modeling were introduced. Chapters 7–10 were devoted to introducing models that allow for (i) *non-proportional substitution patterns* and (ii) *taste heterogeneity*, thereby relaxing the assumptions behind the multinomial logit model.

In Chap. 7, the *nested logit* was introduced. This model is a particular case of the family of *Generalized Extreme Value* (GEV) models. The strength of the nested logit model is that it accounts for non-proportional substitution patterns, while remaining relatively simple in terms of implementation and interpretation. The *probit model* was then presented in Chap. 8. The probit model is distinct from the preceding models in that it assumes a normal distribution of the random utility (remember these terms are central components of the probabilistic statements). Further, the probit model is based on a joint distribution of the random utility terms that includes a covariance matrix. In other words, the distribution of the random utility terms of the different alternatives are not assumed a priori to be independently and identically distributed. In doing so, the probit model allows for flexible correlation structures and therefore for non-proportional substitution patterns. This model is, however, technically and computationally more demanding and the interpretation of the correlation structure is far from intuitive. Whereas probit models are not as commonly used as other models for discrete choice modelling, conceptually they can serve as a basis to develop models that allow for more flexibility, while being more intuitive. In this way, Chap. 8 introduces the readers to the covariance matrix later used in the mixed logit model in Chap. 10.

Early in the book we discussed the possibility of variations in behavior attributable to socio-economic, demographic and/or attitudinal factors. For example, some individuals might be more risk averse (e.g., more willing to choose a variable interest rate on a mortgage) than others. The simplest way to incorporate effects of this kind is to introduce relevant attributes of the decision-makers in the utility functions. Chapters 9 and 10 cover two modeling approaches to deal with *taste heterogeneity* in possibly more sophisticated manners, namely the *latent class logit* model and the *mixed logit model*.

Starting with the latent class logit model, Chap. 9 delved into the issue of taste variation. The latent class logit model is an intuitive approach to segment a sample based on differing tastes. This model directly builds on the multinomial logit model, with the addition of a latent class model, used to segment the sample. The latent class logit model estimates distinct parameters for each class of individuals, and calculates a class-membership probability too. The example presented in Chap. 9 on mode choice illustrates how different coefficients (and behavioral interpretation) are obtained for individuals in different classes. For example, the willingness to pay was calculated for three different classes of travelers; the results illustrate how risk-averse individuals may have a higher willingness to pay (to avoid the risk, here measured in terms of fatality rate). The latent class logit thereby allows to capture taste heterogeneity by grouping individuals into classes, while remaining straightforward in terms of implementation and interpretation.

Another approach to capturing taste heterogeneity was presented in Chap. 10: the *mixed logit*. This model, which also builds on the multinomial logit model, presents a "continuous" approach to the latent class model, where the parameters vary across individuals based on a continuous probability distribution. The probabilities are then calculated based on a weighted average of all possible values of the parameters, here defined by the integral of the density function of the parameters. This approach allows for a more flexible consideration of taste heterogeneity across all individuals. By combining a flexible correlation structure with the multinomial framework, the mixed logit accommodates taste heterogeneity, while providing outcomes from which behavioral insights can be derived. For example, in the example related to air travel in Australia, the results obtained allow to derive a distribution of the willingness to pay to save waiting time.

Following an intuitive approach to introducing variations in model specifications, these two chapters also illustrates how the multinomial logit model can be used to capture variations in taste based on individual characteristics, through *covariates and expansion coefficients*. These chapters illustrate multiple possibilities that exist to specify a model by adjusting some of the blocks of the discrete choice analysis framework, illustrating and comparing different approaches that are available to the modeler to capture taste heterogeneity. Here again, the skills of the modelers are crucial in selecting an appropriate specification and understanding its implication.

It is important to note here that latent class logit and mixed logit models also automatically result in non-flexible substitution patterns. Our division of topics into *substitution patterns* and *taste heterogeneity* is not necessarily either or, but a choice we made in terms of how to present the materials. As readers develop expertise with discrete choice analysis they will come to realize that there are numerous ways to journey and a fuller map of the landscape of discrete choice analysis will hopefully emerge.

The last topic covered in the book is somewhat different. In Chaps. 4 and Chaps. 7–10 we were concerned with *unordered* discrete choices. Ordered choices are another relevant aspect that requires careful consideration. In Chap. 11 we argued that ordinal responses can be modeled as unordered responses, but that this practice is not appropriate due to the loss of information (the natural sequence of the responses is lost) and because this information loss can lead to hidden correlations between alternatives. Perhaps as importantly, ordinal choices are conceptually different from unordered responses: while in multinomial choice situations each alternative is distinct and has its own utility function, in ordinal responses the choice refers to the level or intensity of a single response, so the model is based on a single utility function. For this reason, in Chap. 11, we discussed how the random utility framework can be adapted to deal with ordinal responses. The *ordinal logit* and *ordinal probit* models are presented. The building bricks of these models are a series of differences between binary models separated by *threshold parameters* that indicate the boundary between different levels or categories of responses. Ordinal models illustrate yet again the ability of the discrete choice framework to accommodate a variety of choice structures and situations. Further, in Chap. 11, again, we illustrated how heterogeneity in taste can be captured through variations in scale, something that other multinomial models can be adjusted to do as well.

The End of the Journey?

Understanding decision-making is essential to build more sustainable societies. Anchored in strong theoretical foundations which provide the ground for behavioral insights, discrete choice analysis directly contributes to understanding decision-making. This book has illustrated through various examples how discrete choice models can be used to derive meaningful behavioral interpretations that can directly inform policy-making.

Coming back to the fundamental ingredients of discrete choice modelling, it is important to recognize that these are in constant evolution. *Data* available to researchers continues to evolve with different data collection techniques and the emergence of massive data. These new data sources offer numerous opportunities for improving our understanding of social and economic processes, but also require more powerful computers and methods. Similarly, the *tools* evolve as the body of knowledge expands, together with computational possibilities. As new data sources emerge and tools continue to be developed, a skilled modeler needs to keep up to date with their *technical expertise*. The field of discrete choice analysis will continue to evolve to contribute to a greater understanding of individual choices. As an example, the emergence of machine learning techniques has inspired new research that aims to strengthen discrete choice analysis through the integration of data-driven techniques (Cranenburgh et al. 2022). Here, it is important to bear in mind the trade-offs that exist between fully flexible, but opaque, models vs transparent specifications, resting on understandable assumptions that provide interpretable parameters. A broader debate exists on this very topic (see Rudin 2019) with some pointing at the risks of trying to interpret what may essentially be uninterpretable modeling mechanisms. The strong theoretical roots of discrete choice analysis are perhaps one of the quintessential examples of interpretable methods. That said, it is possible that combining discrete choice analysis with emerging machine learning techniques will contribute to meaningful improvements to informing decision-making.

We aimed to write a text that would engage the reader from Chap. 1 with numerous practical examples, starting from the preliminary analysis of data through model estimation and post-estimation exploratory analysis of model results. If we achieved this goal, the reader will have become familiar with a wide set of tools that can be used to study decision-making behavior as well as with the theoretical foundations of choice modelling. The field of choice modeling is vast and in constant evolution. The models and tools presented in this book are, therefore, far from exhaustive. It is our hope that readers who complete this book will not think of it as a destination, but as a new way of looking at and understanding things.

In this sense, we must clearly identify some possible points of departure for further learning, things that we decided to exclude from this text to keep it as general an introduction as possible. Accordingly, our treatment of discrete choice analysis concentrated on *utility maximization* to the exclusion of other possible decision-making mechanisms, including *regret minimization* (Chorus 2010). We have also assumed throughout that data refer to *revealed preferences*, that is, actual behavior retrospectively recorded. *Stated preferences* in contrast refer to data obtained from controlled experiments where individuals are asked to state their *hypothetical* choices. Stated preferences, while they provide an interesting avenue to collect data on choice situations that cannot be observed, pose some interesting practical and conceptual challenges. These, along with opportunities for research, are covered in magnificent depth by Louvier et al. (2000). Further, we did not cover in great depth data collection, either in revealed or stated preference situations. Finally, we concentrated on one language and a relatively small set of packages, to the exclusion of other packages such as {apollo} (see Hess and Palma 2019) which offers more refined control of model specification and estimation but that we deem more technically demanding. As well, we did not cover other languages such as Python[2] or Julia.[3] These languages might be topics for different books at a later time.

References

Chorus, Caspar G. (2010). A new model of random regret minimization. *European Journal of Transport Infrastructure Research, 10*(2), 181–96.

Hess, S., & Palma, D. (2019). Apollo: A flexible, powerful and customisable freeware package for choice model estimation and application. *Journal of Choice Modelling, 32,* 100170. https://doi.org/10.1016/j.jocm.2019.100170.

Louviere, J. J., Hensher, D. A., & Swait, J. D. (2000). *Stated choice methods: Analysis and applications*. Cambridge: Cambridge University Press.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence, 1*(5), 206–15. https://doi.org/10.1038/s42256-019-0048-x.

van Cranenburgh, S., Shenhao Wang, A. V., Pereira, F., & Walker, J. (2022). Choice modelling in the age of machine learning - discussion paper. *Journal of Choice Modelling, 42,* 100340. https://doi.org/10.1016/j.jocm.2021.100340.

[2] https://biogeme.epfl.ch/.

[3] https://github.com/mattwigway/DiscreteChoiceModels.jl.