Politecnico di Milano
Scuola di Ingegneria Industriale e dell'Informazione
Corso di Studi in Ingegneria Matematica

`Oddsflow`: an interactive tool for the representation of estimated probabilities from discrete choice regression models

Relatore: Prof. Simone Vantini

Tesi di Laurea di:
Giulio Giorgio
matricola 800695

Anno Accademico 2014-2015

Contents

Introduction

Nowadays, a lot of problems involve the act of an agent (like a human being, a firm, a software, *etc.*) of making a choice among a set of distinct alternatives. Examples are countless: choosing a smartphone model given its characteristics in battery life and screen dimensions; following a college preparation class depending on scores get in different exams; placing a heating system considering installation and maintanance costs associated with each option.

One way to model this choice is to integrate it in a probabilistic framework. For each agent are given some attributes that can be observed by a researcher, whose task is to compute the probability for each alternative to be chosen by the agent. A large employed family of methods that provides those estimations is *discrete choice models*, theorized for the first time by Daniel McFadden (1973).

Discrete choice models allow the formulation of a wide variety of problems, including the well-known *multinomial logistic regression*. Indeed, it is a special case where only attributes of the chosen alternative are observed.

Due to this theory environment, in this work we use the term "choices" even when no agent expressed any preference. A lot of databases for which a logistic regression is requested are of this kind, since these problems are faced also in *cluster analysis*. A flower's species, the lake in which an alligator was captured or an employees' working class are not options that can be arbitrary chosen by an undefined entity. They are more likely considered as observations the researcher has to classify in known (or unknown) groups of objects. Nonetheless, the equations that rule the choice of an alternative over the others (or the belonging to a cluster) is neatly described by discrete choice models, thus this is going to be the framework used to describe the theory underneath.

The tools to represent the outcomes of a logistic regression (and, for extension, of a discrete choice analysis) are mostly limited to the binomial case of two alternatives from which to choose. When it comes to multinomial problems, in which the options an agent faces are more than two, the graphical representations lack. Despite one can commit on the same instruments

used for classical binary logistic regression, these are able to compare just a pair of alternatives at a time. A large set of them entails the production of a huge number of outputs, which are hardly to confront and interpret. On top of that, using methods from binary logistic regression to estimate and represent quantities of a multinomial regression leads an intrinsic error that, although small, sometimes cannot be ignored.

In this work we present the `Oddsflow` app: an original interactive tool to represent both estimated probabilities from a discrete choice analysis, and the influences that each attribute (i.e. the covariates in the model) has on the afore-mentioned probabilities.

For each explanatory variable in the model, one or more different `Oddsflow` plots are produced. Each alternative is placed as node of a complete digraph. Nodes are connected by edges depicting the regression coefficients that estimated the log odds between the pair of alternatives represented by vertices. The network then represents a sort of probability flux that, when the variable increases, flows to the nodes where the arrows point to, and leaves the nodes from which the arrows start. On each edge are also placed the estimation of the corresponding coefficient's expected value, its standard error and the associated Wald test's P-value.

Furthermore, the `Oddsflow` app allows the user to manage a basic discrete choice analysis. The user chooses the variables that will be exploited in the model confronting several discrete choice models on the basis of some index of interest. The next step is a panel where can be displayed multiple interactive `Oddsflow` plots, that instantly change depending on the covariate's values given by the user. All the graphs represented in this spot can be then exported and saved.

The thesis is organized as follows: Chapter 1 contains all the theory basis to comprehend the concept used. It starts with a section dedicated to discrete choice models, than describes some statistical inference tools, then some definitions from graph theory are recapped. Chapter 2 describes two methods used to represent multinomial logistic databases and regressions presented in the literature. Chapter 3 contains the explanation of how the `Oddsflow` plot works, what is meant to describe and its possible variants. In Chapter 4 the `Oddsflow` app is presented: from the variable selector, to the interactive `Oddsflow` plots, to the data uploader. Finally, in Chapter 5 some applications of the `Oddsflow` app in which its functional simplicity emerges are discussed.

Chapter 1

Theory Basis

1.1 Discrete Choice Models

In this chapter we consider two statistical methods for the analysis of discrete choice: *multinomial logit* and *conditional logit* models. Both models' aim is to analyze the choice of an individual among a set of alternatives and provide a set of probabilities associated to the choice of each alternative. Despise the fact that in the beginning the *multinomial logit* technique was more often employed, the growing spread of big databases made possible the adoption of the *conditional logit* model: in fact it requires a more complete set of regressors, which has to vary across alternatives, and not just across individuals. The applications for these techniques are countless: from the simple choice to take the bus or the car to go to work every morning, to the segmentation of a factory deadline work.

1.1.1 General Framework

The cornerstone of discrete choice models is a choice among a set of J options faced by an agent (i.e. person, firm, decision maker). The categorical response variable is called Y. The outcome of the decision in any given situation is denoted as y.

Taking a causal perspective, we can suppose that there are several factors (observed and unobserved), that determine the agent's choice. The attributes of the alternatives observed by the researchers are labeled $\mathbf{x}$, and the unobserved ε. The decision y depends on observed and unobserved factors through a function $y = h(\mathbf{x}, \varepsilon)$, called *behavioral process*. Given both $\mathbf{x}$ and ε, the agent's choice y is fully determined.

Our model will not be deterministic, due to the fact that ε is not observed. Agent's choices cannot be predicted exactly, but are integrated in a proba-

bilistic framework, which allows the derivation of the outcome probabilities. This framework consists in the assumption that ε is a random variable, with probability density $f(\varepsilon)$. The likelihood that the agent chooses one of the J possible outcomes is the probability that the behavioral process $h(\mathbf{x}, \varepsilon)$ equals to the corresponding y:

$$P(Y = y|\mathbf{x}) = P(\varepsilon : h(\mathbf{x}, \varepsilon) = y).$$

This probability can be expressed with an indicator function to get an integrable form. Define an indicator function:

$$\mathbb{1}_{\{h(\mathbf{x},\varepsilon)=y\}}$$

that takes value of 1 when the statement in the pedex is true, and 0 when the statement is false. Then $mathbbm1_{\{h(\mathbf{x},\varepsilon)=y\}} = 1$ when $\mathbf{x}$ and ε induces the agent to choose alternative y, and the corresponding probability is the expected value of this indicator function, over all the possible values of ε:

$$P(Y = y|\mathbf{x}) = P(\varepsilon : h(\mathbf{x}, \varepsilon) = y) = \int_{\varepsilon} \mathbb{1}_{\{h(\mathbf{x},\varepsilon)=y\}} f(\varepsilon) d\varepsilon.$$

Differences among models lie in specifications of $f(\varepsilon)$ and $h(\mathbf{x}, \varepsilon)$, and depending on these functions, the integral can be expressed in closed form. In these cases, the choice probability can be calculated exactly from the closed-form formula.

1.1.2 Random Utility Model

Discrete choice models are usually derived under an assumption of utility-maximizing behavior by the decision maker. Models derived that way are called *random utility models* (RUMs).

A decision maker, labeled n, faces a choice among J alternatives. Each choice j is paired with an utility level U_{nj}, which is assumed known just by decision maker. Instead, the researcher observes a value of utility V_{nj} (called *representative utility*) on the basis of explanatory variables, i.e. some observable attributes of the decision maker. This value is presumably different from U_{nj}, due to features that the researcher ignores. This difference is modelized as a random value ε_{nj}, defined as $\varepsilon_{nj} = U_{nj} - V_{nj}$.

The joint density of the random vector $\boldsymbol{\varepsilon}_n = (\varepsilon_{n1}, \ldots, \varepsilon_{nJ})$ is denoted $f(\boldsymbol{\varepsilon}_n)$. This density allows probabilistic statements about the decision maker's choice. The probability that decision maker chooses alternative i is:

$$\begin{aligned} P_{ni} &= P(U_{ni} > U_{nj}, \forall j \neq i) \\ &= P(V_{ni} + \varepsilon_{ni} > V_{nj} + \varepsilon_{nj}, \forall j \neq i) \\ &= P(\varepsilon_{nj} - \varepsilon_{ni} < V_{ni} - V_{nj}, \forall j \neq i). \end{aligned} \tag{1.1}$$

This probability is a cumulative distribution, namely the probability that each random term $\varepsilon_{nj} - \varepsilon_{ni}$ is lower than the observed quantity $V_{nj} - V_{ni}$. Assuming that $\boldsymbol{\varepsilon}_n$ is distributed with density $f(\boldsymbol{\varepsilon}_n)$, this cumulative probability equals to:

$$\begin{aligned} P_{ni} &= P(\varepsilon_{nj} - \varepsilon_{ni} < V_{ni} - V_{nj}, \forall j \neq i) \\ &= \int_{\varepsilon} \mathbb{1}_{\{\varepsilon_{nj} - \varepsilon_{ni} < V_{ni} - V_{nj}, \forall j \neq i\}} f(\boldsymbol{\varepsilon}_n) d\boldsymbol{\varepsilon}_n. \end{aligned} \tag{1.2}$$

1.1.3 Logit Models

The most preferred model used for discrete choice selection is logistic. It spread thanks to the closed form of the integral employed to compute estimated probabilities for each alternative and it is easily interpretable. Originally, the logit formula was derived by Luce (1959) assuming that the odds of choosing option i over j do not depend on the other alternatives (*independence of irrelevant alternatives*, IIA). McFadden (1973) proved that in the logit formula, unobserved utility ε has necessarily extreme value distribution.

The cumulative distribution function is:

$$F(x; \mu, \beta) = e^{-e^{-(x-\mu)/\beta}}.$$

This distribution is described by parameters μ (real number called *location*) and β (real non-negative number called *scale*).

The logit model is obtained by equation 1.2 assuming that each ε_{nj} is independent and identically distributed extreme value with $\mu = 0$ and $\beta = 1$. The probability density for each unobserved utility fraction is

$$f(\varepsilon_{nj}) = e^{-\varepsilon_{nj}} e^{e^{-\varepsilon_{nj}}}, \tag{1.3}$$

and the cumulative distribution is

$$F(\varepsilon_{nj}) = e^{-e^{-\varepsilon_{nj}}}. \tag{1.4}$$

The difference between two extreme value variables is distributed logistic. That is, if ε_{nj} and ,ε_{ni} are *i.i.d.* extreme value, then $\varepsilon^*_{nij} = \varepsilon_{nj} - \varepsilon_{ni}$ follows

the logistic distribution and

$$F(\varepsilon^*_{ni}) = \frac{e^{\varepsilon^*_{nij}}}{1 + \varepsilon^*_{nij}}. \tag{1.5}$$

The extreme value distribution for the errors (and hence the logistic distribution for the error differences) is nearly the same as assuming that the errors are independently normal. The extreme value distribution gives slightly fatter tails than a normal, which means that it allows for slightly more aberrant behaviour than the normal.

1.1.4 Derivation of Choice Probabilities

What follows is the derivation of choice probabilities (McFadden, 1973) under the assumptions given in the previous sections. The probability that decision maker n chooses alternative i is

$$\begin{aligned} P_{ni} &= P(V_{ni} + \varepsilon_{ni} > V_{nj} + \varepsilon_{nj}, \forall j \neq i) \\ &= P(\varepsilon_{nj} < \varepsilon_{ni} + V_{ni} - V_{nj}, \forall j \neq i). \end{aligned} \tag{1.6}$$

If ε_{ni} is considered given, (1.6) is the cumulative distribution for each ε_{nj} evaluated at $\varepsilon_{ni} + V_{ni} - V_{nj}$, which, according to (1.4) is

$$e^{-e^{-(\varepsilon_{ni}+V_{ni}-V_{nj})}}.$$

Since each ε is independent, this cumulative distribution over all $j \neq i$ is the product of the individual cumulative distributions:

$$(P_{ni}|\varepsilon_{ni}) = \prod_{j \neq i} e^{-e^{-(\varepsilon_{ni}+V_{ni}-V_{nj})}}. \tag{1.7}$$

Since ε_{ni} is not given, the unconditionally probability is the expected value of the previous expression with respect to ε_{ni}:

$$\begin{aligned} P_{ni} &= \int_{-\infty}^{+\infty} (P_{ni}|\varepsilon_{ni})\, f(\varepsilon_{ni}) d\varepsilon_{ni} \\ &= \int_{-\infty}^{+\infty} \left(\prod_{j \neq i} e^{-e^{-(\varepsilon_{ni}+V_{ni}-V_{nj})}} \right) e^{-\varepsilon_{ni}} e^{e^{-\varepsilon_{ni}}} d\varepsilon_{ni}. \end{aligned} \tag{1.8}$$

In the integral we find the expression for all alternatives, including the i

alternative.

$$\begin{aligned} P_{ni} &= \int_{-\infty}^{+\infty} \left(\prod_j e^{-e^{-(\varepsilon_{ni}+V_{ni}-V_{nj})}} \right) e^{-\varepsilon_{ni}} d\varepsilon_{ni} \\ &= \int_{-\infty}^{+\infty} e^{-\sum_j e^{-(\varepsilon_{ni}+V_{ni}-V_{nj})}} e^{-\varepsilon_{ni}} d\varepsilon_{ni} \\ &= \int_{-\infty}^{+\infty} e^{-e^{\varepsilon_{ni}} \sum_j e^{-(V_{ni}-V_{nj})}} e^{-\varepsilon_{ni}} d\varepsilon_{ni}. \end{aligned} \tag{1.9}$$

With the following change of variable:

$$t = e^{\varepsilon_{ni}} \Rightarrow dt = -e^{\varepsilon_{ni}} d\varepsilon_{ni}$$

The unconditional probability is the following integral:

$$P_{ni} = \int_0^{+\infty} e^{-t \sum_j e^{-(V_{ni}-V_{nj})}} dt,$$

which has a closed form:

$$\begin{aligned} P_{ni} &= \left[-\frac{e^{-t \sum_j e^{-(V_{ni}-V_{nj})}}}{\sum_j e^{-(V_{ni}-V_{nj})}} \right]_0^{+\infty} \\ &= \frac{1}{\sum_j e^{-(V_{ni}-V_{nj})}} \end{aligned}$$

and can be rewritten as the logit probability:

$$P_{ni} = \frac{e^{V_{ni}}}{\sum_j e^{V_{nj}}}. \tag{1.10}$$

1.1.5 Representative Utility Shape for Logit Models

Representative utility is usually linear in parameters. While working with discrete-choice logit models, two possible variables should be considered with respect to the categorical variable chosen as the response in the model: *individual specific* and *alternative specific* variables.

Suppose that a decision maker n chooses alternative j. Individual specific variables are variables independent from the alternative chosen. For each individual they share a common value among the J options. Alternative specific variables, instead, express values for all the J alternatives among the decision maker n can choose. For each individual there are J distinct values, each corresponding to a different alternative.

Then, depending on the coefficients that multiply them, variables can be of three types:

- alternative specific variables x_{ni} with a generic coefficient β,
- individual specific variables z_n with an alternative specific coefficient γ_i,
- alternative specific variables w_{ni} with an alternative specific coefficient δ_i.

The representative utility of decision maker n for alternative i is then:

$$V_{ni} = \alpha_i + \beta x_{ni} + \gamma_i z_n + \delta_i w_{ni}.$$

Only differences are relevant in the model described in the previous sections. This means that we are interested in the difference between the observed utilities of decision maker n among alternatives i and j:

$$V_{ni} - V_{nj} = (\alpha_i - \alpha_j) + \beta(x_{ni} - x_{nj}) + (\gamma_i - \gamma_j)z_n + (\delta_i w_{ni} - \delta_j w_{nj}). \quad (1.11)$$

Coefficients for individual specific values have to be alternative specific (and the intercept too), otherwise they would disappear in the differentiation. Besides, only differences between these coefficients are relevant and can be identified. For example, For example, with three alternatives 1, 2 and 3, the three coefficients γ_1, γ_2, γ_3 associated to an individual specific variable cannot be identified, but only two linear combinations of them. Therefore, a possible choice of normalization is to set the $\gamma_1 = 0$ and 1 is then called *baseline category*.

Coefficients for alternative specific variables may (or may not) be alternative specific. For example, transport time is alternative specific, but 10 minutes in public transport may not have the same impact on utility than 10 minutes in a car. In this case, alternative specific coefficients are relevant. Monetary time is also alternative specific, but in this case, one can consider than 1 euro is 1 euro whatever it is spent in car or in public transports 3. In this case, a generic coefficient is relevant (Croissant, 2013).

A model with all individual specific variables is called *multinomial logistic* model. If all variables are alternative specific, it will be called *conditional logistic*. When there are variables of both types, the model is called *mixed logistic*.

For multinomial logit models, the difference between two representative utilities among alternative i and j can be written as logarithmic odds ratio using (1.10)

$$\log \frac{P_{ni}}{P_{nj}} = \log \frac{e^{V_{ni}}}{e^{V_{nj}}} = V_{ni} - V_{nj} = (\alpha_i - \alpha_j) + (\gamma_i - \gamma_j)z_n. \quad (1.12)$$

Setting the J category as baseline (often the last one or the most common), multinomial logit models pair each response category with the baseline to describe simultaneously $J - 1$ log odds.

1.1.6 Coefficients Computation

Maximum likelihood fitting of logit models maximizes the likelihood subject to P_i simultaneously satisfying the $J-1$ equations that specify the model. Despite the differences between multinomial, conditional and mixed logistic, the models share a common likelihood (Hoffman and Duncan, 1988). For $n = 1, \ldots, N$, let $\mathbf{y}_n = (y_{n1}, \ldots, y_{nJ})$ represent the multinomial trial for decision maker n, where $y_{ni} = 1$ when the response is in category i and $y_{ni} = 0$ otherwise.

$$\log \ell = \sum_n \sum_i y_{ni} P_{ni}. \tag{1.13}$$

Via numerical methods that will not be mentioned in this work, the maximization of (1.13) subject to P_{ni} leads to estimate coefficients in 1.11.

1.2 Statistical Inference for Categorical Data

The distribution for the response variable has unknown parameter values. About them it is possible to make inference with some of the methods reviewed in this section.

1.2.1 Likelihood Functions and Maximum Likelihood Estimation

Under weak regularity conditions, such as the parameter space having fixed dimension with the true value falling in its interior, maximum likelihood estimators have desiderable properties. They have large-sample normal distributions, are asymptotically consistent, converging to the parameter as the number of observations increases, and are asymptotically efficient, producing large-sample standard errors no greater than those from other estimation methods (Agresti, 2002).

Given the data, for a chosen probability distribution the *likelihood function* is the probability of those data, treated as a function of the unknown parameter. The maximum likelihood (ML) estimate is the parameter value that maximizes this function. This is the parameter value under which the data observed have the highest probability of occurrence.

In this section, the parameter for a generic problem is denoted θ and its ML estimate $\hat{\theta}$. The likelihood function is $\ell(\theta)$ and its logarithm, the log-likelihood (easier to work with since it is a sum rather than a product of terms) is $L(\theta) = \log \ell(\theta)$. For many models, $L(\theta)$ has a concave shape and $\hat{\theta}$ is the point at which the derivative equals 0. The ML estimate is

then the solution of the likelihood equation, $\partial L(\theta)/\partial\theta = 0$. Often, θ is multidimensional, denoted by $\boldsymbol{\theta}$, and $\hat{\boldsymbol{\theta}}$ is the solution of a set of likelihood equations. The standard error of $\hat{\theta}$, which is denoted SE, is asymptotically defined as the square root of the inverse of the information matrix .

1.2.2 Wald Test

The *Wald test* is a significance test of a null hypothesis $H_0 : \theta = \theta_0$ which exploits the large-sample normality of ML estimators. With non null standard error SE of $\hat{\theta}$, the test statistic

$$t = \frac{\hat{\theta} - \theta_0}{SE}$$

has an approximate standard normal distribution when $\theta = \theta_0$. One refers z to the standard normal table to obtain one- or two-sided P-values. Equivalently, for the two-sided alternative, t^2 has a chi-squared null distribution with 1 degree of freedom. The P-value is then the right-tailed chi-squared probability above the observed value. This type of test, using the standard error, is called a Wald test (Wald, 1943).

1.2.3 Likelihood Ratio Test

Let ℓ_0 be the maximum likelihood over the possible parameter values under H_0, and ℓ_1 the maximum likelihood over the larger set of parameter value without any additional assumption, validating a hypothesis H_1. The ratio $\Lambda = \ell_0/\ell_1$ cannot exceed 1. Wilks (1935) proved that the quantity called *likelihood-ratio test statistic*:

$$-2\log\Lambda = -2\log(\ell_0/\ell_1) \tag{1.14}$$

is distributed chi-squared as the number of observations goes to infinity. The degrees of freedom equal the number of parameters loss in the passage from H_1 to H_0 (the difference among their parameter space's dimensions).

1.2.4 McFadden's R^2

As in ordinary regression, R^2 describes the power of explanatory variables to predict the response, with $R^2 = 1$ for perfect prediction. A measure that directly uses the likelihood function is the one proposed by McFadden (1973).

Denote the maximized log likelihood L_M for a given model and L_0 for the null model containing only an intercept term, then

$$R^2_{McFadden} = 1 - \frac{\log \ell_M}{\log \ell_0} = 1 - \frac{L_M}{L_0}. \tag{1.15}$$

Since probabilities are no greater than 1, so log likelihood are nonpositive. Furthermore, an increasing in model's complexity expands the parameter's space, so the relation $0 \geq L_M \geq L_0$ always states.

To understand whether this definition makes sense, suppose first that the covariates in the current model give no predictive information about the outcome. This means that the likelihood value ℓ_M will not be much greater than the likelihood of the null model. Their ratio will be close to 1, and $R^2_{McFadden}$ will be close to 0. Next, if our model predicts perfectly all the variations in the response, it recreates with probability ≈ 1 all the choices made by decision makers. So the model's likelihood will be close to 1 and its logarithm close to 0, which leads to $R^2_{McFadden} = 1$.

1.3 Graphs

We now introduce some essential definitions from graph theory, in order to easily understand how the data representation will be structured. These objects were chosen for their shape, which shines when it comes to make connections between objects intuitive and clear. For this reason, no algorithm or theorem will be considered.

Definition 1. A *graph* is an ordered pair $G = (V, E)$ comprising a set V of *vertices* or *nodes* together with a set E of *edges* or *lines*. (Biggs et al., 1986)

Definition 2. A *directed graph* is a graph where the edges have a direction associated with them.

Definition 3. A *complete digraph* is a directed graph in which every pair of distinct vertices is connected by an edge.

Chapter 2

Graphical Representations

The graphical representation of a multinomial logistic regression has always been a little painful. Imagine the common case of a dataset with many explanatory variables and a response variable with a lot of alternatives: the classic tools to represent any kind of information are too dispersive or inefficient, due to their derivation from the binary logistic regression.

In this section we take a tour to some common logistic data (and regression) representation to highlight their limits and find a way to exceed them.

2.1 Graphical Contingency Tables

The *contingency tables*, term introduced by Pearson (1904), are tables which cells contains the frequency counts of the outcomes for a sample. Table 2.1 is an example from an article that studied effects on racial characteristics on whether persons convicted of homicide received the death penalty (Agresti, 2002).

As we can see from it, the table summarizes the entire data set (provided that all variables are categorical) in a combination of rows and columns depending om the number of alternatives in the variables. This instrument is fundamental to approach the statistical analysis on a categorical set of data, but since it gathers a superficial information, its utility quickly ceases.

For example, if we were searching for a form of biased or discriminant behavior based on the race in the death penalty verdict, the sole amount of people belonging to the two races for which a court chose the death penalty would be obviously not enough, even for a preliminary investigation.

We can also draw a *graphical contingency table*, like in Figure 2.1, in which the area of each square represents the portion of outcomes, split by their

Table 2.1: Death Penalty Verdict by Defendant's Race and Victims' Race

Victims' Race	Defendant's Race	Death Penalty	
		Yes	No
White	White	53	414
	Black	11	37
Black	White	0	16
	Black	4	139

Source: M. L. Radelet and G. L. Pierce, *Florida Law Rev.* **43:** 1-34 (1991).

belonging to one of the alternatives of the variables. Neither the graphical

Figure 2.1: *Graphical contingency table* of data set in Table 2.1

approach helps to collect anything relevant about our purpose, since it is just a different way to look at the same object.

Authors often fill contingency tables with also a column in which the frequency of one of the alternative of the response variable can be read. It is possible to redesign the squares in the *graphical contingency table* to let their areas be somehow dependent from the frequencies of the adjoined column, but the new figure will not be likely requested without its counterpart, since

it is good use that relative frequencies are presented with the absolute ones.

In Table 2.2 appears a list (called "Percent Yes") with probabilities that the defendant is sentenced to death based on his race and on his victims' race.

Table 2.2: Death Penalty Verdict by Defendant's Race and Victims' Race with Probabilities based on Races

1 Victims' Race	Defendant's Race	Death Penalty		Percent Yes
		Yes	No	
White	White	53	414	11.3
	Black	11	37	22.9
Black	White	0	16	0.0
	Black	4	139	2.8

Source: M. L. Radelet and G. L. Pierce, *Florida Law Rev.* **43:** 1-34 (1991).

Our quick study may then end with a logistic regression which we discover with that there was a sort of racial discrimination in death sentences in the time the data were been collected.

The next example shows how the previous graphical tool becomes an inefficient way to start an analysis when the number of variables and alternatives in categorical variables increases. Figure 2.2 is the graphical contingency table from a study of factors influencing the primary food choice of alligators (Agresti, 2002). It used 219 alligators captured in four Florida lakes, and the response variable is the primary food found in the animals' stomach. This has five categories: fish, invertebrate, reptile, bird, other; the data are also classified according to the lake of capture, the gender and the length (≤ 2.3 meters long, > 2.3 meters long).

In the graph it is clear how the various dimensions of the squares make the interpretation pretty hard, being the classes of animals impossible to compare in just a glance. Furthermore, the explanatory variables in this model are still few; considering databases or problems with a lot more, committing to this graph would probably bring nothing than confusion.

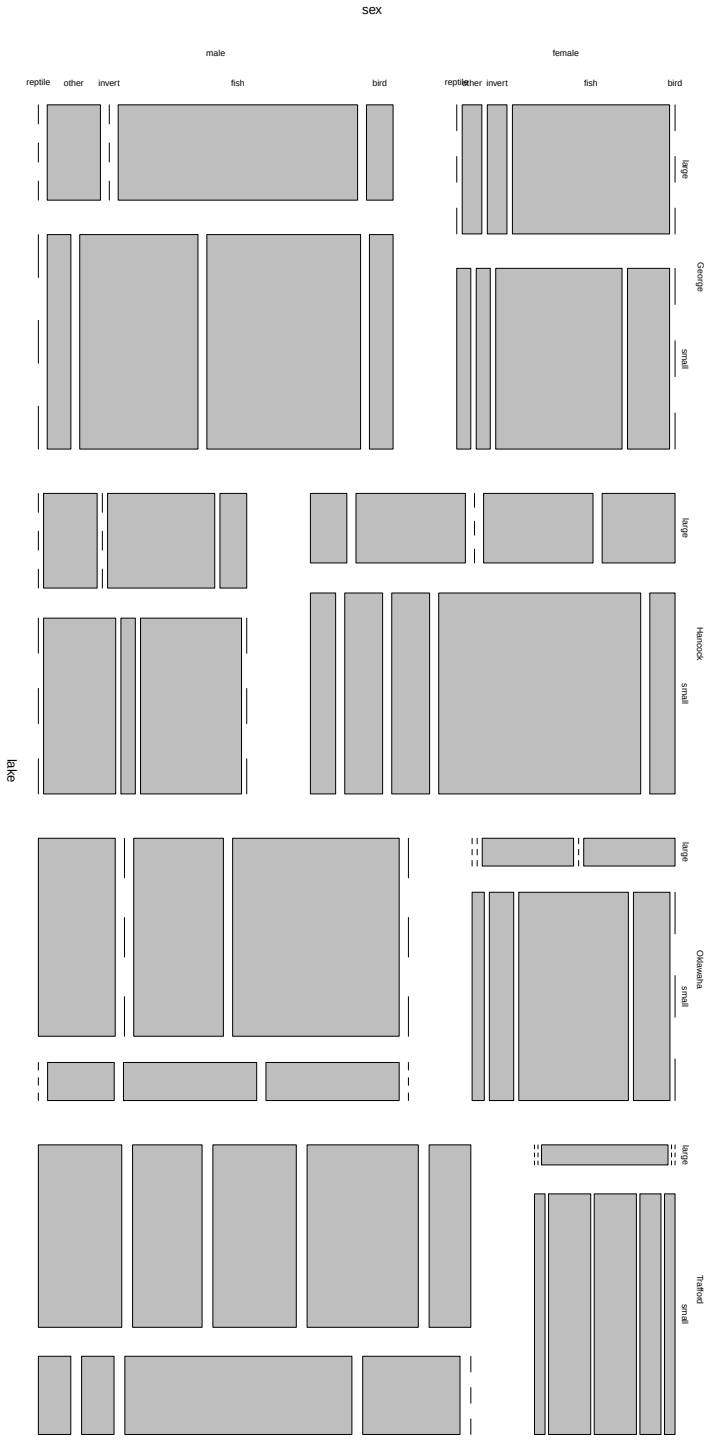


Figure 2.2: *Graphical contingency table* for the `Alligator` data set.

2.2 Estimated Probabilities

The aim of our work is to provide an immediate graphic tool to read and understand the results from the prevision part of a logistic regression. In this case the response is categorical, so the model used for the analysis will return J probabilities that a decision maker n (with his set of values for the explanatory variables in the data set) will choose each one of the J alternatives.

If one (or more) of these explanatory variables are quantitative, it is informative to plot the estimated probabilities associated with each alternative. The other predictors are left constant while the one that will be represented varies among his range. For each quantitative regressor, the graph consists so in an overlapping of curves, one for each option.

In the next example (UCLA: Statistical Consulting Group), the options of students entering high schools can be modeled using career and social indicators. The individual may choose among general program, academic program and vocational program. We consider the writing score as the only explanatory variable.

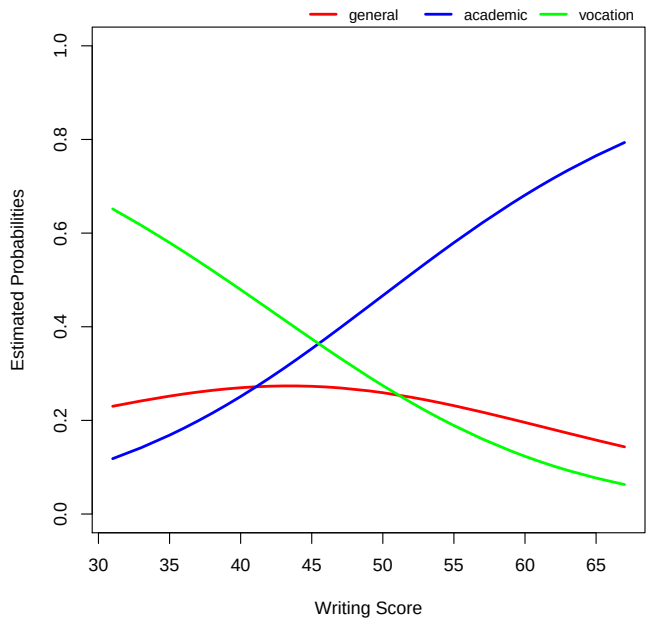


Figure 2.3: Estimated Probabilities for the `student` database

As shown in Figure 2.3 an increase in the writing score raises the probability that a student selects an academic program over a vocational or general one. In the picture there is no information about the regression coefficients

and their statistical relevance in the explaining of the phenomenon (for example in the form of a P-value).

We may try to guess that, taking the general program as baseline category, the log odd between the academic and the baseline program increases when the writing score increases. This would suggest a positive regression coefficient $\gamma_{academic}$ corresponding to the writing score variable, but does not help to quantify it, nor gives any clue on its relevance in the model's estimation.

The plot becomes gradually harder to read and interpret when the number of alternatives in the response variable increases. However, this representation has its limit in the choice of fixed values for all other explanatory variables. Leading univariate regression models and plotting estimated probability plots for them might be useful to locate the intervals of interest in which study the probability behavior. But a high number of explanatory variables makes this estimation progressively harder, and forces the researcher to visually compare a lot of these plots.

Chapter 3

The `Oddsflow` plot

In this chapter we present an original graphical representation for the estimated probabilities of a mixed logistic regression. It exploits the shape of a complete digraph and allows to relate in the same figure the estimated probabilities and the regression coefficients, given with their statistical relevance according to the Wald test (Chapter 1).

The software used for the regression is the R package `mlogit` (Croissant, 2013), the one used to produce plot graphs is the `igraph` package (Csardi and Nepusz, 2006).

3.1 The `Oddsflow` plot's structure

The idea behind the `Oddsflow` plot is an interactive and updatable graph $G = (V, E)$ with a vertex for each alternative in the response categorical variable and a directed edge for each pair of vertices. We first consider the multinomial logistic model, in which all variables are individual specific.

Suppose that the user wants to display the `Oddsflow` plots for a hypothetical multinomial regression model. They are asked to indicate all the values for the explanatory variables of the decision maker n for which the probabilities of falling in one of the J categories are requested. Then they select the variables whose regression coefficients will be displayed. For each variable one or more `Oddsflow` plots are produced, depending on its continuous or discrete nature. Vertices and edges have a peculiar representation for the sake of plots' interpretation:

- every node is represented by a circle which area depends on the alternative's estimated probability. This probability is written close to each node;

- every edge is represented by an arrow which shaft's width increases with the absolute value of the chosen variable's regression coefficient $\gamma_{ij} = \gamma_i - \gamma_j$. The coefficient is printed in the middle of the arrow and is followed by its standard deviation and a combination of symbols (taken from the classic summary output in R) synthesizing its Wald test's P-value[1].

In case of only continuous explanatory variables the arrows are designed as follows (the case with discrete variables is treated in the next section). For each couple of nodes, the log odd model between the alternatives represented by the nodes is extracted from the regression model. The alternative chosen as the baseline in the model will be indicated as j, while the other as i. Then the coefficient γ_{ij} relative to the variable for which the `Oddsflow` plot is drawn is considered: if it is positive, the arrow that joins the two nodes will start from the one representing the baseline alternative j and will point to the node representing i. Otherwise, the arrow will be plotted starting from the node representing i and pointing to j. Each edge then always represents the positive contribute of the variable to the log odd of the alternative which the arrow points to, over the alternative which the arrow starts from. For this reason, the estimated coefficients printed on edges are taken without their signs, since they are summarized yet by edges' orientations.

This process is repeated for each couple of alternatives/nodes. The final outcome is a set of complete digraphs (one for each explanatory variable) with heterogeneous vertices and edges resembling a flow network.

Assume now that we would like to know how the estimated probabilities of a multinomial logistic regression vary when an arbitrary model's explanatory variable increases or decreases its value: the `Oddsflow` plot retains the needed information. Augmenting that value would increase the areas of the nodes pointed by the arrows, and decrease the areas from which the arrows start.

This representation allows the interpretation according to every arrow is associated with a sort of transferred probability flow, which shifts from an alternative to another depending on the variation in the explanatory variable. But what discussed in this work is not a flow in any mathematical or engineered meaning: the application of algorithms from operations research or optimization is useless, since the graph structure is here employed just for its properties in visualizing links between objects.

Nonetheless, varying the explanatory variables' values induces a changing in the nodes areas that follows the indications given by the arrows. The user can experience this interactivity with the `Oddsflow` app (Chapter 4) and

[1]The R notation is: "$***$" for values < 0.001, "$**$" for values < 0.01, "$*$" for values < 0.05, "." for values < 0.1, " " for values < 1.

imagine that a metaphorical flow of probability passes from an alternative to another after an external variation.

Inside this network, the probability does not get wasted, since all values depicted close to the nodes always sum to 1. Finally, the node which only edges start from can be seen as a source node, and the one which only edges arrives as a sink node. In fact, if the value tends to positive (negative) infinity, all the estimated probability will be concentrated in the alternative represented by the sink (source) node.

3.2 An Individual Specific Example: The `student` Database

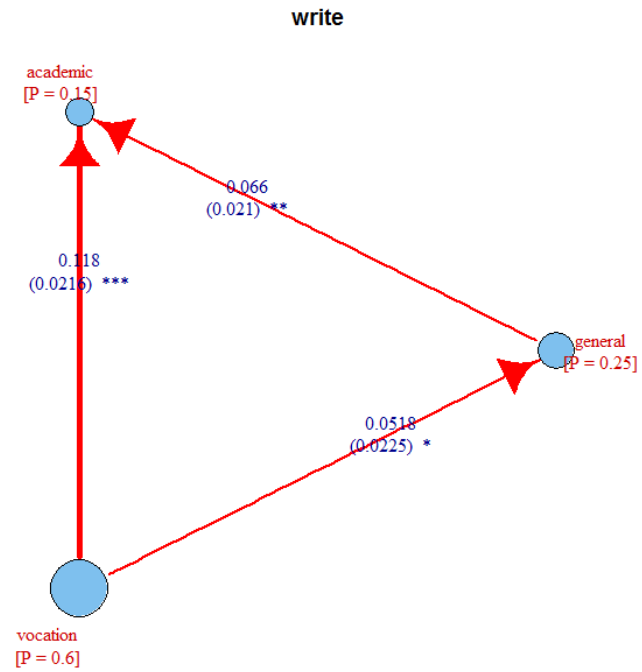
An example with individual specific regressors is the students' program choice (UCLA: Statistical Consulting Group). The model's goal is to estimate the probability of choosing a program given the writing score of an undefined exam and the socioeconomic index (ses).

The continuous `Oddsflow` plot

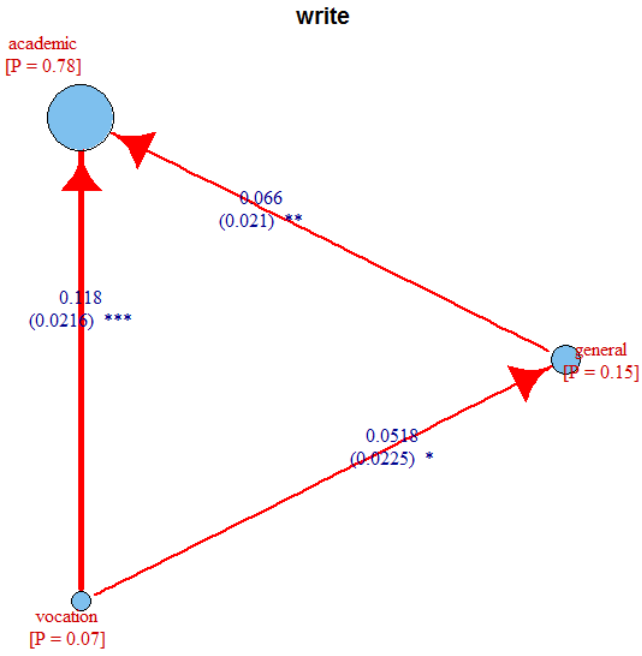
Figure 3.1 shows that increasing the writing score, the probability of choosing an academic program instead of a general or vocational one increases, since both the arrows from the two other alternatives are pointed to the academic vertex. The opposite situation affects the vocational program's probability decreasing, while the general program's case is different.

Comparing the coefficients corresponding to edges that leave and arrive from a vertex, it is possible to get an idea of the estimated probability for the values at the endpoints of the range interval of the explanatory variable. If the sum of the leaving edges' coefficients exceed the sum of the arriving edges' coefficients, at the right interval's endpoint we can expect a negative difference of probability flow, that makes the estimated probability value lesser than the one computed at the left endpoint. The same states in the opposite way, generating a positive difference. In the `student` example, since general has a leaving edge of value 0.066 and an arriving edge of value 0.0518, there will be a probability negative gap between the lowest writing score in the database and the highest. This is confirmed by the curve in Figure 2.3.

Each `Oddsflow` plot recaps some values estimated by the regression model that was employed to produce it. Concerning the `student` example, Figure 3.1 gives some information about the three log odds that the model is



(a) writing score = 34



(b) writing score = 66

Figure 3.1: `Oddsflow` plots for the students database with writing score = 34 and writing score = 66.

designed to describe:

$$\begin{aligned}
\log \frac{P_{academic}}{P_{vocation}} &= \alpha_{academic} + 0.118\ z_{write} \\
\log \frac{P_{academic}}{P_{general}} &= \alpha_{academic} + 0.066\ z_{write} \\
\log \frac{P_{vocation}}{P_{general}} &= \alpha_{vocation} - 0.0518\ z_{write}.
\end{aligned} \tag{3.1}$$

Where $\alpha_{vocation}$ in the first equation and $\alpha_{general}$ in the second and third equations are set to 0, since they are constant contributes for the baseline category.

As described in the previous section, regression coefficients' signs are integral with arrows' directions. Log odds with positive z_{write} coefficients increase when z_{write} increases, so arrows are directed from the alternative on the denominator to the one on the numerator. On the other side, an increase in z_{write} leads to a decrease in those log odds (like $\log \frac{P_{vocation}}{P_{general}}$) in which z_{write} has a negative coefficient. Probability of choosing a vocation program over a general one would decrease, and the arrow from vocational node to general node is drawn in the exact opposite on respect of the other two.

Some of these contributes may be omitted on the basis of the regression coefficients' Wald test P-values. For example, considering significant a term with P-value equal to 0.001 or less, only the log odds between academic and general program are statistically influenced by the writing score. The other two log ratios may be considered constant.

All these equations are fully expressible with the `Oddsflow` plot and app (Chapter 4). The former gives the multiplicative coefficients, the latter all the constants α.

The Discrete `Oddsflow` plot

Among explanatory variables there may be some categorical, for which the same continuous representation cannot be used. Their variation (in the non-ordinal case) is not monotonous, but just a switch between alternatives without any hierarchy. This means that it is impossible to estabilish if a change in these covariates is an increasing or a decreasing. To keep the same structure, discrete variables' `Oddsflow` plots depict effects of indicator variables, which shift from 0 to 1 depending on two of the explanatory variables' alternatives.

Thus, for each qualitative regressor with K alternatives, every possible switch among each alternative has to be represented. For $K > 1$, the amount

of these non-repeating pairs is equal to

$$N = \sum_{k=1}^{K-1} k. \tag{3.2}$$

For this reason, N `Oddsflow` plots are generated to map all possible switches from any pair of options leading to variations in the estimated probabilities.

An example of non repeating pair is the switch "on → off", which is treated as "off → on" because the two log odds are equal except for the coefficients' signs:

$$\log \frac{P_{on}}{P_{off}} = -\log \frac{P_{off}}{P_{on}}.$$

To present the graphical outcome, we focus on another formulation of the `student`'s problem. Among all disposable covariates, it is possible to select the social economic status (called *ses*, a discrete variable) and use it to estimate the probability of choosing one of the three programs. Since ses has three alternatives (*low, middle* and *high*), $K = 3$. N is then equal to 3 and `Oddsflow` produces 3 graph plots.

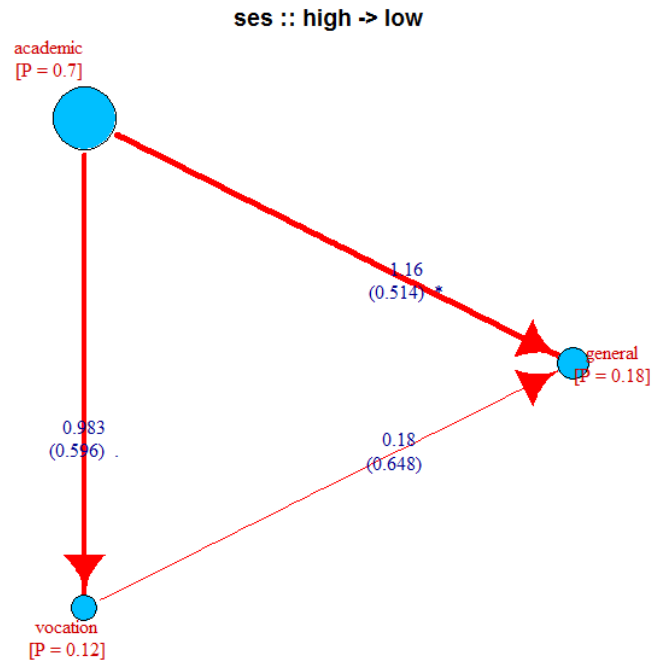
The arrows in Figure 3.2 represent the regression coefficients for a dummy variable indicating the social economic status in the regression formula. According to the plot's title, the dummy takes value 1 when the individual has a low social economic status, and zero when has high;

Modifying this variable causes log odds' increasings and decreasings as already explained: for example, switching from high to low ses raises the log odds of general and vocational program against academic program, and increase the relative estimated probability.

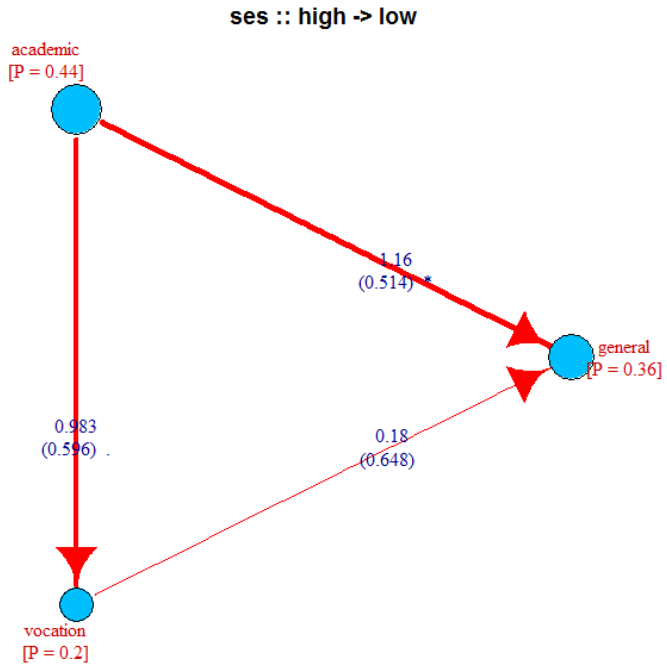
From this model's `Oddsflow` plots (Figures 3.2 and 3.3) can be derived the following log odds:

$$\begin{aligned}
\log \frac{P_{academic}}{P_{vocation}} &= \alpha_{academic} - 0.983\ \mathbb{1}_{\{high \to low\}} - 1.27\ \mathbb{1}_{\{high \to middle\}} \\
\log \frac{P_{academic}}{P_{general}} &= \alpha_{academic} - 1.16\ \mathbb{1}_{\{high \to low\}} - 0.63\ \mathbb{1}_{\{high \to middle\}} \\
\log \frac{P_{vocation}}{P_{general}} &= \alpha_{vocation} - 1.16\ \mathbb{1}_{\{high \to low\}} + 0.645\ \mathbb{1}_{\{high \to middle\}}
\end{aligned} \tag{3.3}$$

The alternative high is automatically chosen by R as baseline category for the explanatory variable ses. Figure 3.4 is then redundant to collect all the regression coefficients, since their plots are generated setting low as

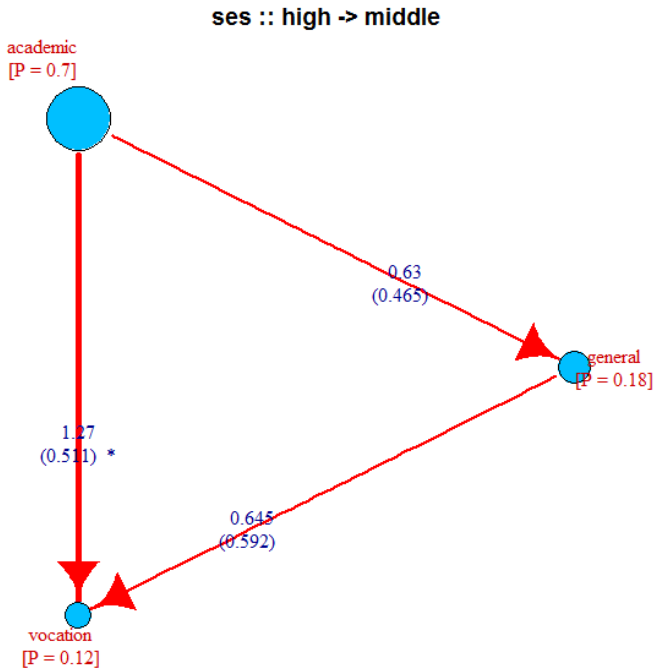


(a) ses = high

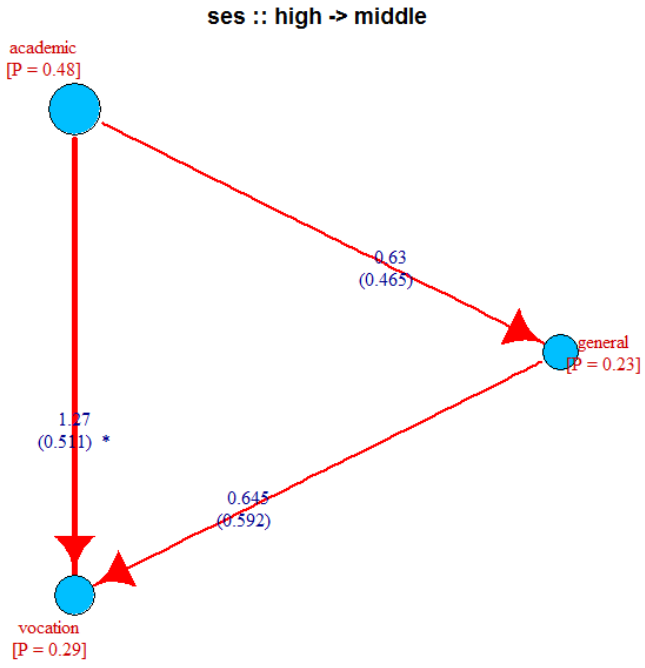


(b) ses = low

Figure 3.2: `Oddsflow` plots for the dummy variable $\mathbb{1}_{\{high \to low\}}$: $\mathbb{1}_{\{high \to low\}} = 1$ when ses = low; $\mathbb{1}_{\{high \to low\}} = 0$ when ses = high in the `student` data set.

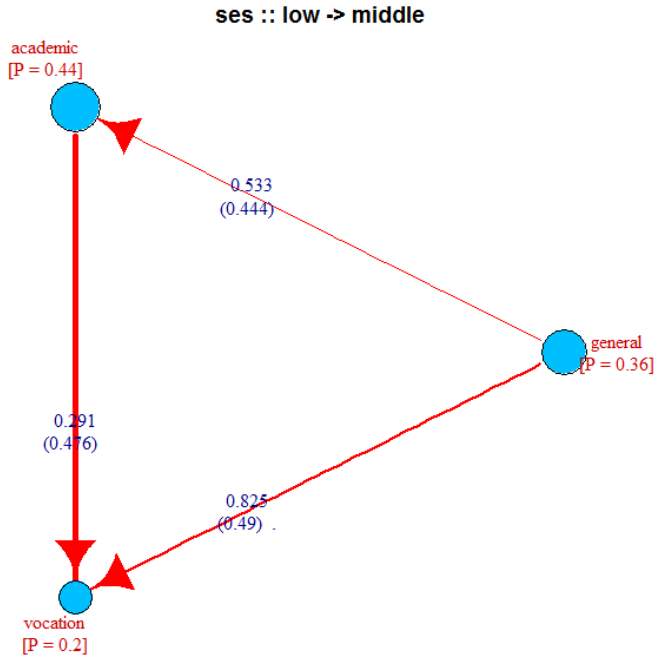


(a) ses = high

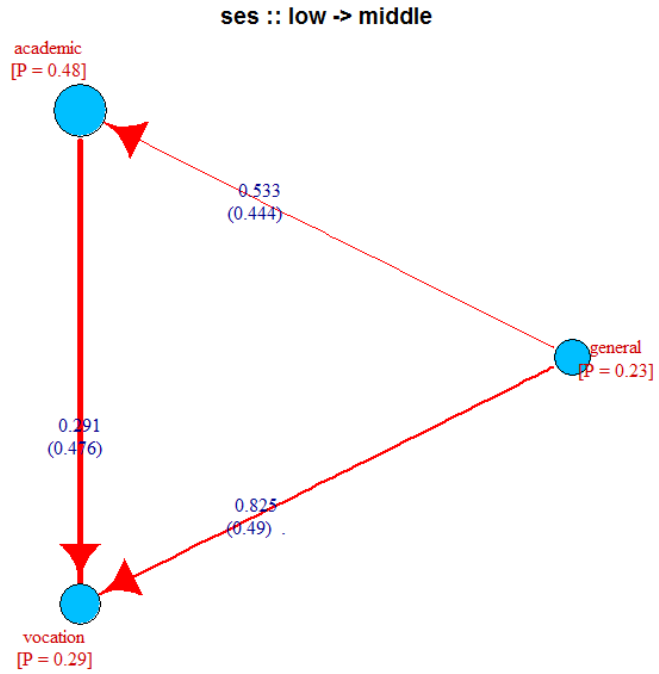


(b) ses = middle

Figure 3.3: `Oddsflow` plots for the dummy variable $\mathbb{1}_{\{high \to middle\}}$: $\mathbb{1}_{\{high \to middle\}} = 1$ when ses = middle; $\mathbb{1}_{\{high \to middle\}} = 0$ when ses = high in the `student` data set.



(a) ses = low



(b) ses = middle

Figure 3.4: `Oddsflow` plots for the dummy variable $\mathbb{1}_{\{low \to middle\}}$: $\mathbb{1}_{\{low \to middle\}} = 1$ when ses = middle; $\mathbb{1}_{\{low \to middle\}} = 0$ when ses = low in the `student` data set.

baseline category for the switches. They are still included among disposable `Oddsflow` plots for the model, since the user may be interested in studying switches that are not just the ones given by default.

3.3 An Alternative Specific Example: the Fishing Database

As seen in Chapter 1, alternative specific explanatory variables may have coefficients independent from the baseline alternative chosen for the regression. The probability flow through the alternatives is then trivial, since the estimated probabilities are directly (or indirectly, depending on coefficients' signs) proportional to variations in the corresponding alternative's regressor value.

For these variables whose coefficients are alternative specific, the `Oddsflow` plot drops the arrows representation and depicts just the nodes and their regression coefficients with standard deviation and P-value indications on their right side. The color of each node reflects the respective coefficient's value in the model, emphasizing with graduating shades the alternative's influence. On the left side of each plot is a color palette to suggest the distances between coefficients' values. For positive values, the color shades from yellow (the zero) to red, for negative values from light blue (the zero) to purple.

The `Fishing` data set (Cameron and Trivedi, 2005) contains choice of recreational fishing mode, in which data may depend on both the individual and the alternative. The response variable is the place where the decision maker chooses to fish (*boat, beach, charter* and *pier*). The database has two alternative specific variables (the catching rate and the price asked to fish in the place selected), and one individual specific variable (the income).

In Figure 3.5 the alternative specific coefficients for the catch variable are displayed. They allow the user to confront the different values for each choice, select those with P-value lower than a desired threshold and check which alternative contributes most in the representative utility shape.

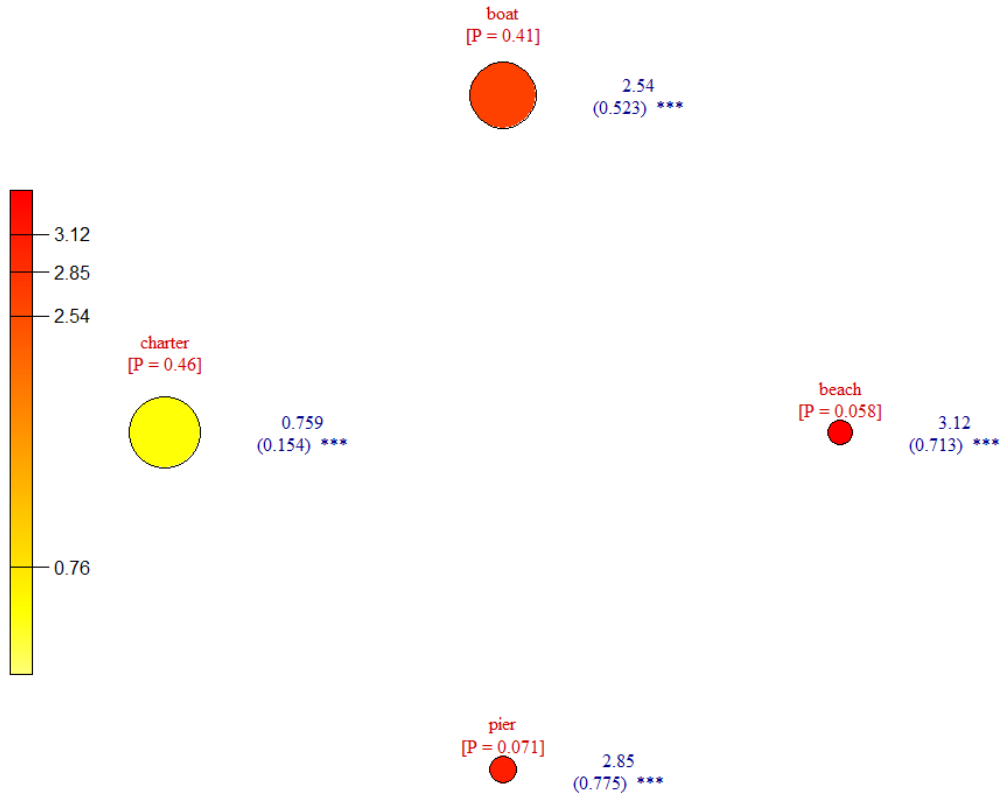


Figure 3.5: `Oddsflow` plot for the catching rate in the Fishing dataset.

Chapter 4

The `Oddsflow` app

The primary issue of our work is the development of an interactive tool to compute estimated probabilities from a discrete choice regression model. The `Oddsflow` plot introduced in Chapter 3 is the static version of the namesake application, which gives the user some tool to make variable selection and the possibility to manipulate the new subject's attributes to compute and compare the estimated probabilities, varying an arbitrary number of explanatory regressors.

The software used to develop the interactive part is Shiny (RStudio Team, 2014), an extension to Rstudio (RStudio Team, 2015), one of the most powerful `R` language IDE.

4.1 The Variable Selector

Once the user submitted the data set to the `Oddsflow` app, the first interactive interface displayed is the variable selector. It allows to choose the categorical variable going to be the response and the explanatory ones among all the others. If needed, a regression summary can be visualized to evaluate both the regressors' statistical relevance and how the model fits the data via some indicators supplied by the `mlogit` package.

According to the database's shape, a different variable selector is employed. As described in Chapter 1, a variable is called alternative specific provided that the attributes of all the alternatives are expressed (not just the attributes for the chosen option). Regarding databases with both alternative and individual specific variables, our software is limited to those with just one categorical variable, that is the response. Furthermore, is possible to set a generic coefficient (a constant) for each alternative specific variable, or an alternative specific one.

Variable Selector with only Individual Specific Variables

For this section, the `student` dataset (UCLA: Statistical Consulting Group) will be used as sample. Figure 4.1 shows the variable selector's start.

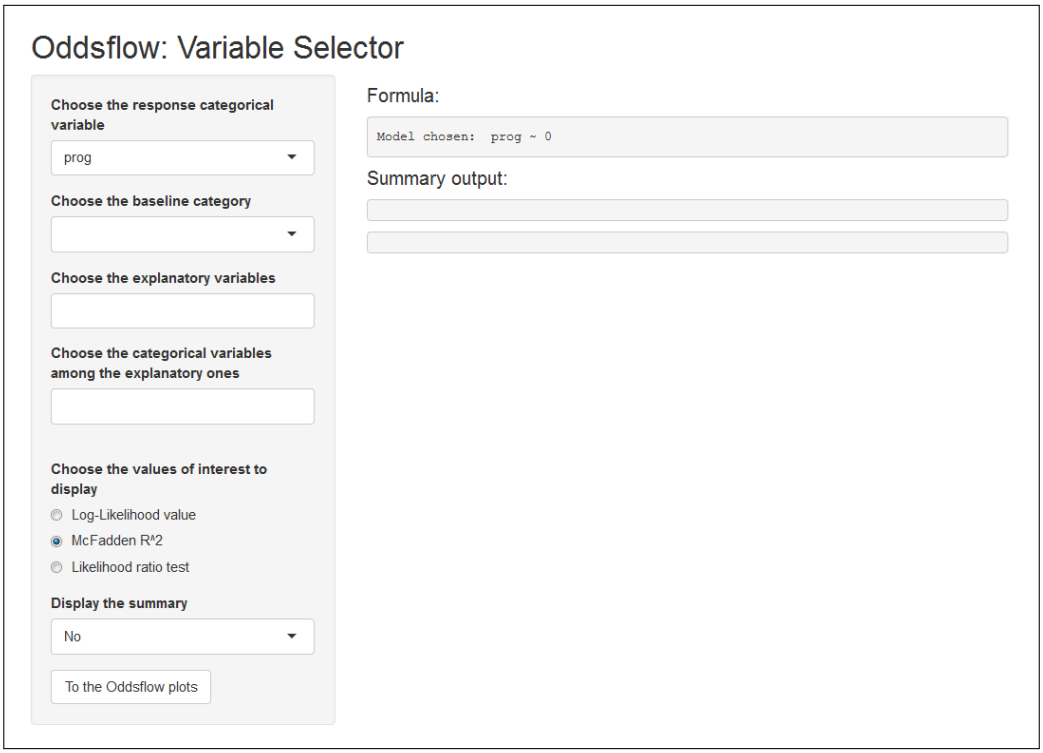


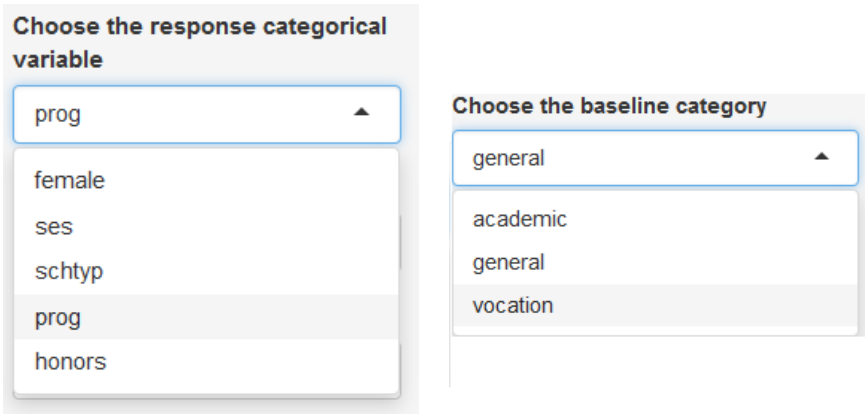
Figure 4.1: The first step of a variable selection procedure with the `Oddsflow` app.

The first *widget* (the web element that the user can interact with) from the top allows to select the response variable among all the categorical ones in the dataset. Below the user can choose which baseline category is going to be set for the logistic regression, launched by filling all the selector's fields. The functioning of these two widgets is depicted in Figure 4.2.

The choice of a baseline category is only relevant for the summary outputs reported in the variable selector. It is irrelevant in the next steps of the `Oddsflow` app.

The following widgets let the user choose the explanatory variables. Once selected, they appear in the white box and are simply added to the right side of the final formula, which the regression will be triggered for.

Depending on the analyzed data, a categorical variable may appear as a numeric array or as an array of strings. It is possible to detail which variable

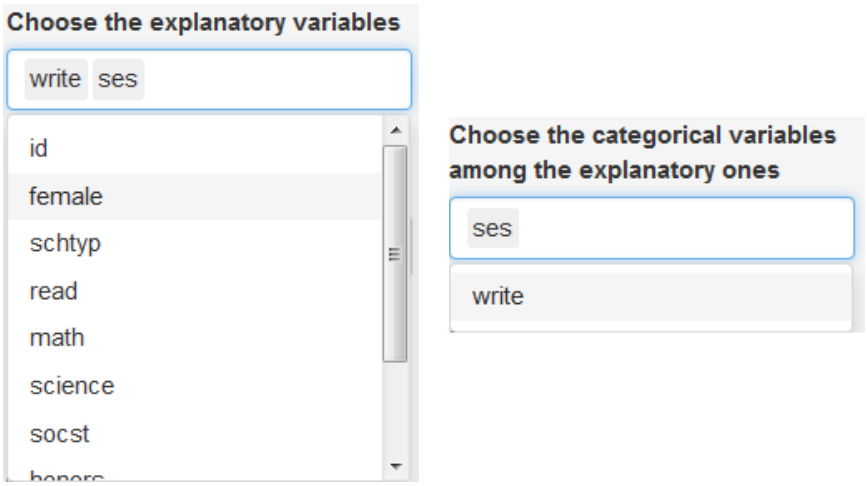


(a) *Categorical Response Selector* (b) *Baseline Category Selector*

Figure 4.2: Steps to select the response variable.

is categorical and which is continuous with the widget in Figure 4.3b. The software chooses by default the variables that `R` recognizes as categorical.

For completeness of the presentation, in Figure 4.3 the writing score (a continuous variable) and the social economic status (a categorical variable) has been selected.



(a) *Explanatory Variables Selector* (b) *Detailing on the variables' nature*

Figure 4.3: Steps to select the explanatory variables.

If the "Display the summary" widget is switched on "Yes", the result will be the screen in Figure 4.4. The user may eventually base his variable selection on some values of interest expressed in the `mlogit` package summary, without

displaying the whole of it. These indicators appear in the first line of the "Summary output" panel.

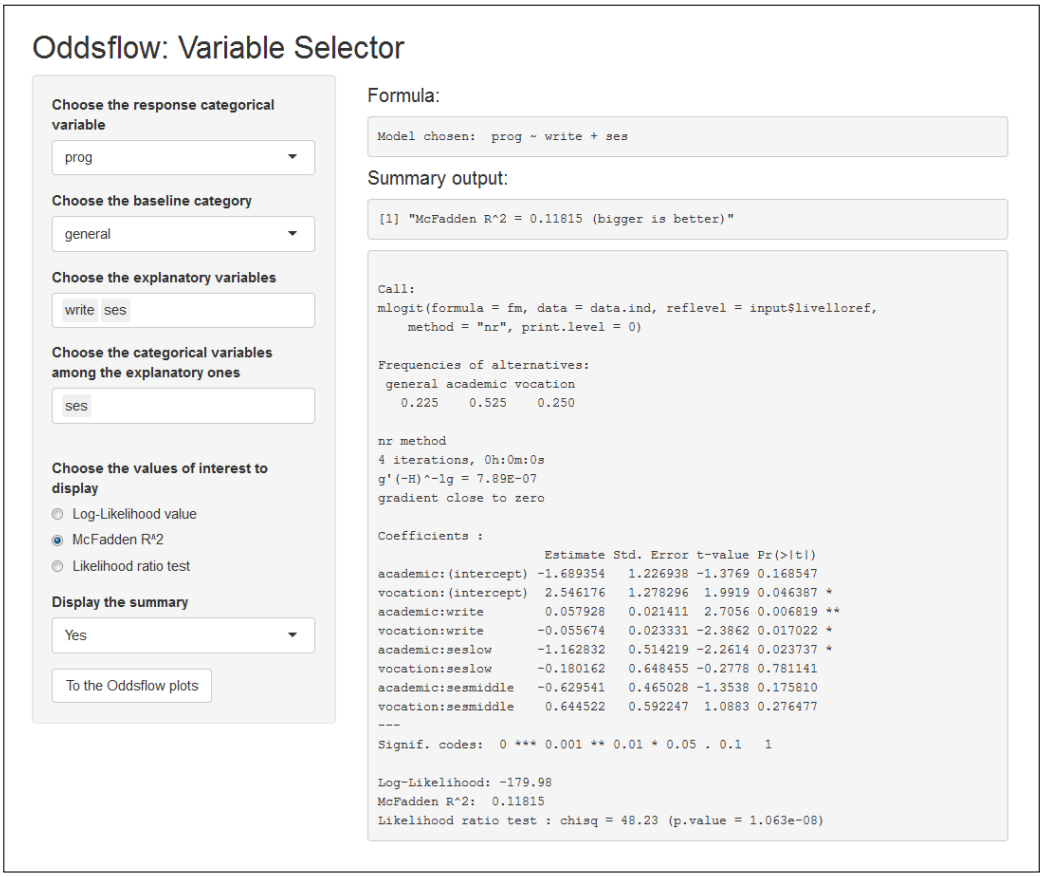


Figure 4.4: The result of a variable selection procedure with the `Oddsflow` app.

The R summary (with its P-values and indexes of interest) is automatically updated every time the user changes something in the model formulation, like removing an unnecessary explanatory variable or setting a different baseline alternative.

The interactivity inherent in the variable selector has been designed to be friendly to the non-expert user, who's taking count of how the model fits the data just via some indicators or is interested in the `Oddsflow` plot only. Nonetheless a skilled operator may delve a little bit deeper in the analysis confronting more than an indicator at time, or basing his selection on statistical evidences explained by P-values. As anticipated in Chapter 3, by changing the baseline alternatives the user can also draw the constant

coefficients useful to complete equations (3.1) and (3.3).

When it has come to an acceptable model, clicking on the "To the Oddsflow plots" button brings to the interactive `Oddsflow` plots.

Variable Selector with Alternative Specific Variables

The structure of a dataset with at least one alternative specific variable is fairly different from an usual set of data. In fact, for some variables there are information on all the possible response's alternatives, also on those that have not been chosen. In our work, we focus on datasets with one categorical variable (the response) and an arbitrary number of continuous explanatory variables, which can be alternative or individual specific.

For this section, the sample employed is the `Fishing` dataset (Cameron and Trivedi, 2005). Figure 4.5 shows the variable selector's start for the case with alternative specific variables.

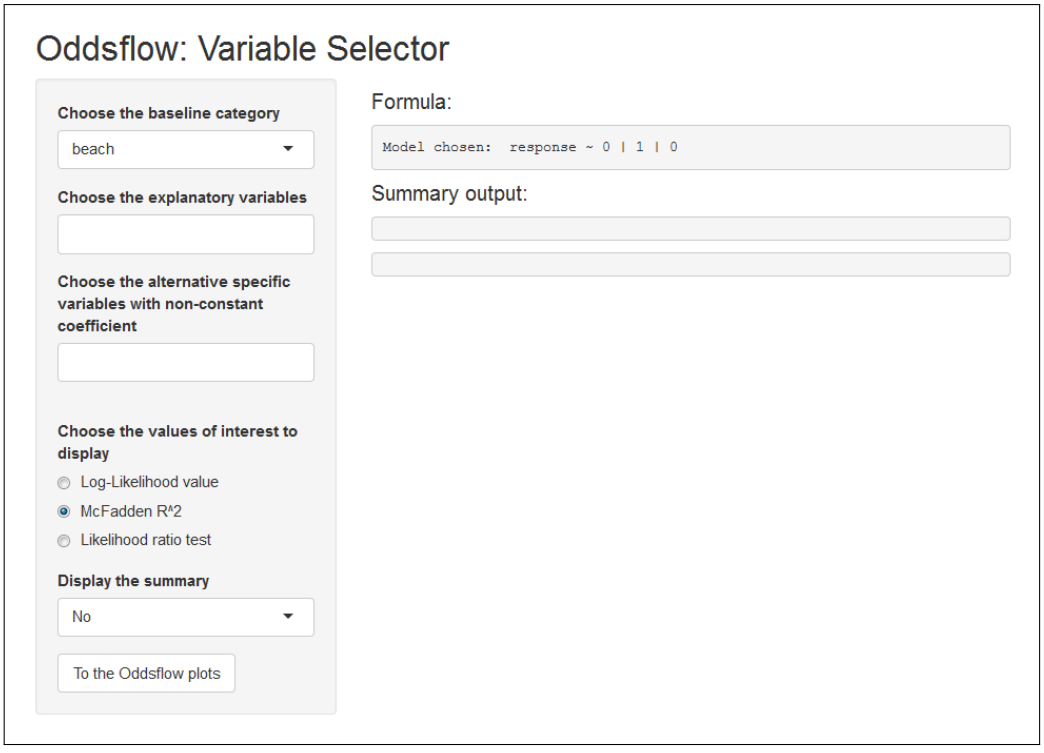
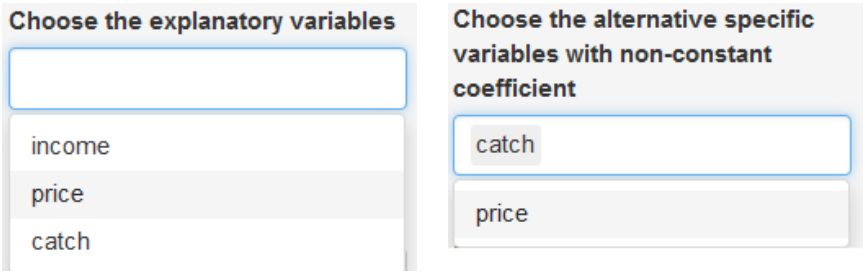


Figure 4.5: The first step of a variable selection procedure with the `Oddsflow` app (case with alternative specific variables).

The differences with Figure 4.1 are the inability to choose the model's response variable and the detailing on the alternative specific variables. These

can be introduced in the formula with a generic coefficient, or with a coefficient dependent on the categories. The "Choose the alternative specific variables with non-constant coefficient" widget automatically fetch for the alternative specific among the explanatory variables, and can establish which regressor is needed a constant coefficient for.



(a) *Explanatory Variables Selector* (b) *Detailing on the alternative specific variables' nature*

Figure 4.6: Steps to select the explanatory variables (case with alternative specific variables).

On the top of the right panel is the model formula, updated like the summary after each alteration in the widgets. Its structure is introduced in the R package `Formula` (Zeileis and Croissant, 2010) and in the right side the regressors are put in the following order: on the left of the first separator are the alternative variables with generic coefficient, between the separators are the individual specific variables with alternative specific coefficients and on the right of the second separator are the alternative specific variables with alternative specific coefficients. The "Summary output" of the example's model returns the same information carried by the previous example. The final step is displayed in Figure 4.7.

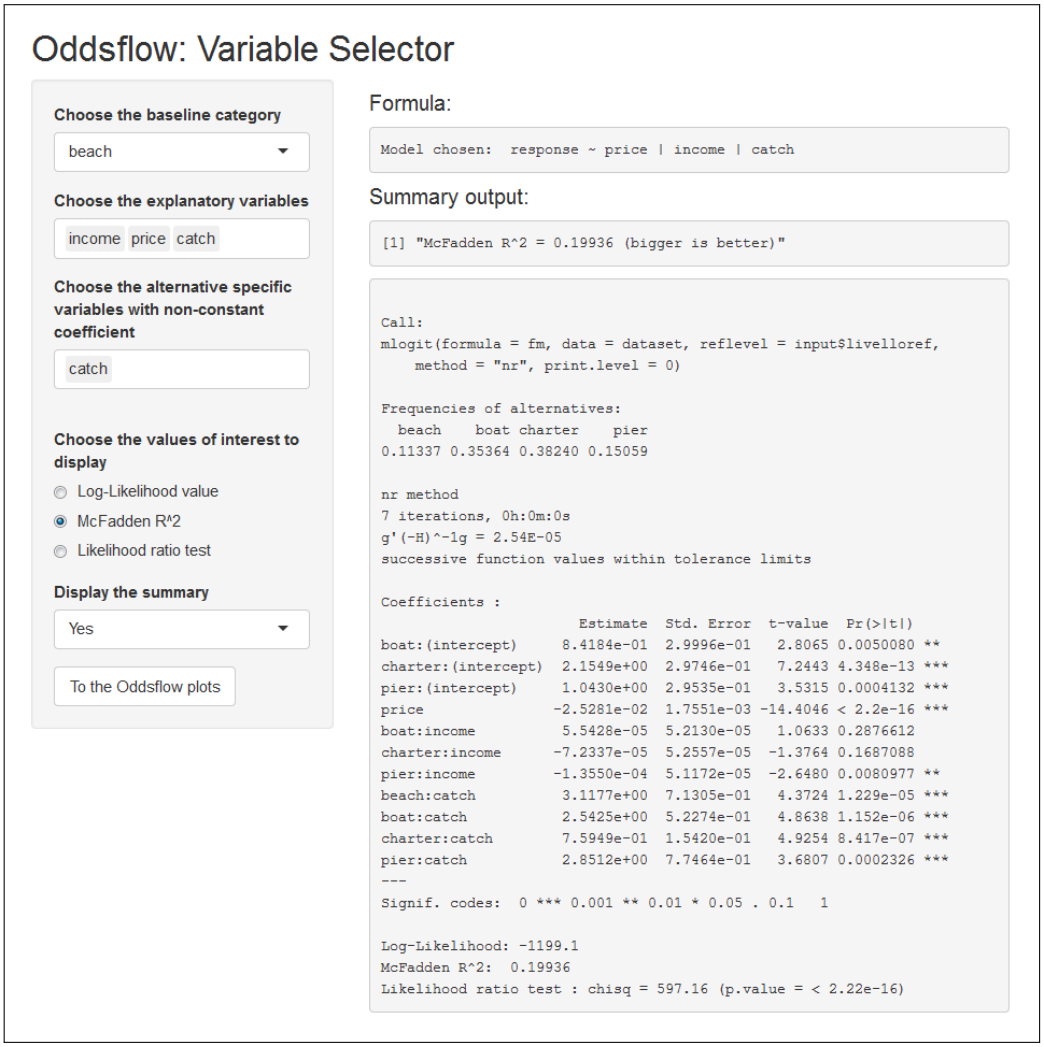


Figure 4.7: The result of a variable selection procedure with the `Oddsflow` app (case with alternative specific variables).

4.2 The Interactive Oddsflow Plots

The second step of the Oddsflow app is the panel of interactive plots. The Oddsflow graph plots seen in Chapter 3 are here loaded and changes depending on the explanatory variables' values that the user sets via the widgets in the left panel of the application. It is also possible to display multiple plots in the same room, and fix a P-value threshold to hide the arrows corresponding to those coefficients with P-value under that limit.

Interactive Oddsflow Plots with Individual Specific Variables

Figure 4.8 depicts the right away step of the software after Figure 4.4.

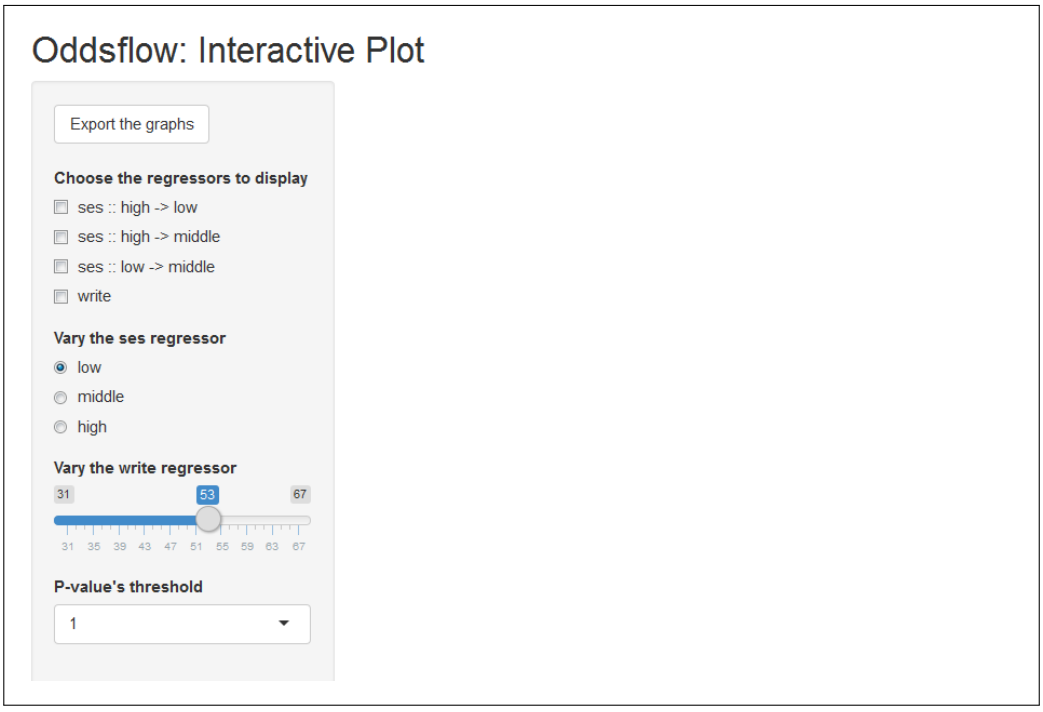


Figure 4.8: The panel with interactive Oddsflow plots at its start.

The first widget from the top in Figure 4.8 allows the user to decide which regressor's Oddsflow plot to display in the right panel of the application. Since it is a multi-selection tool, more than one graph at a time can be visualized.

Following the first widget are values selectors, one for each individual specific explanatory variable in the model, and J for each alternative specific

explanatory variable, where J is the number of alternatives in the response variable. The continuous sliders span the variable's range (the maximum and minimum values in the data set), and are set by default on their average values in the data set. The discrete selectors, on the other hand, have a button for each option in the categorical variables, and are set by default on the first of each of them in alphanumerical order. Each combination of these settings represents the choices of a decision maker which the estimated probabilities are calculated for.

Finally, the user can set a P-value threshold. Only contributions of regressors (represented by the arrows) with P-value lower than the level set are going to be represented. The possible thresholds are reported in Section 3.1.

Every click of the "Export the graphs" button summons the R plot device, from which is possible to save or import all the right panel contents in the formats permitted by the above-mentioned environment.

Figure 4.9 is an example with one only graph plotted. Figures 4.11 and 4.10 display how the estimated probabilities change varying both the continuous and the categorical regressors.

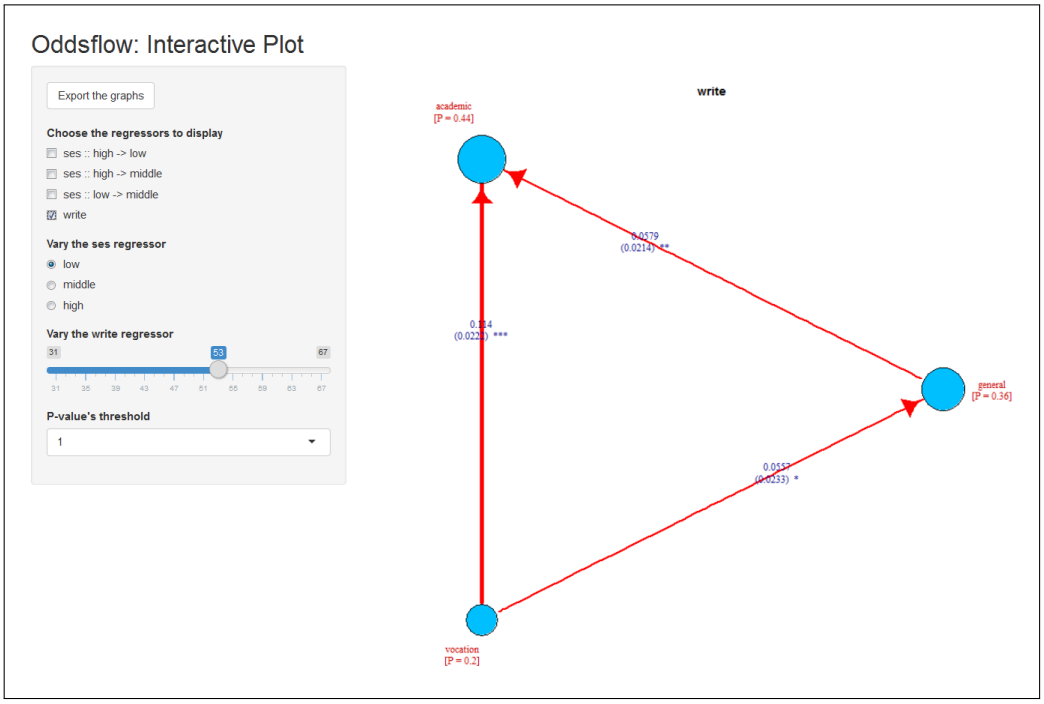


Figure 4.9: Interactive writing score's `Oddsflow` plot.

The differences between each `Oddsflow` plot are the direction of the arrows and the values written on them, because each graph refers to a specific

variable, with proper regression coefficients. On the other hand, the estimated probabilities are in common for each plot because are generated from the same set of variables.

Confronting Figures 4.11 and 4.10 the user may also notice that the estimated probabilities are now changed, while the arrows' aspect is not. It is due to the fact that the former are calculated for a different individual, with writing score = 34 and ses = high, and the latter depends only on the model, and not on the data submitted to make statistical prediction.

To make possible a comparison between coefficients from different continuous regressors, it's possible to plot more than one `Oddsflow` plot at a time in the right panel of the app. But variables can have different unities of measures; in this case the comparison of coefficients can be troublesome, since every one is scaled on its referring variable.

For this reason, when more than one `Oddsflow` plot is depicted, the arrows' shafts depend on coefficients extracted from the same model, but with the variables' standard score instead of the raw variables. This procedure leaves intact the signs, rescaling the values to allow a comparison between the coefficients and to avoid gigantic arrows that would compromise the plots' fruition.

Interactive `Oddsflow` Plots with Alternative Specific Variables

As assumed in Chapter 3, we consider only databases with continuous alternative specific variables, and one categorical variable as response. When it comes to compute estimated probabilities, the user has to submit data on the attributes of all the alternatives, and not just the attributes for the chosen alternatives.

In the `Fishing` database example (Cameron and Trivedi, 2005), we stated that the variable price's coefficient will be generic, and the coefficient associated with the catch variable will vary on the basis of the response's alternatives. The `Oddsflow` plot for the catch variable is the same described in Section 3.2, while there is not any graphical representation for the price coefficient's variable, since it is just a number, independent from individual and from the response's categories. It's still possible to set the values for the price variable because it is a regressor in the model, so is necessary to compute the estimated probabilities.

All the features for individual specific variables are identical to the case with only individual specific variables. An instance of the `Oddsflow` interactive plot is in Figure 4.12.

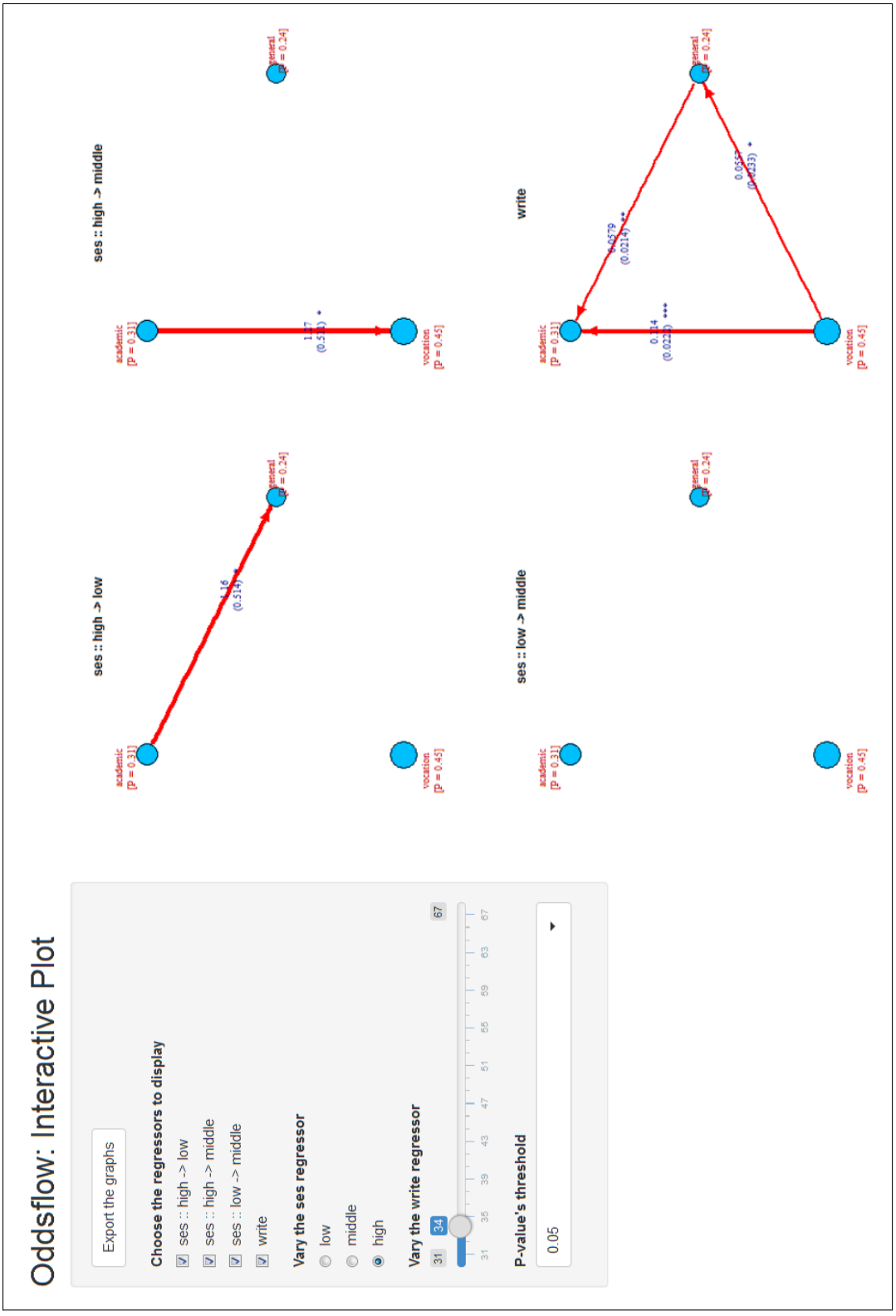


Figure 4.10: Interactive `Oddsflow` plots with with writing score = 34 and ses = high.

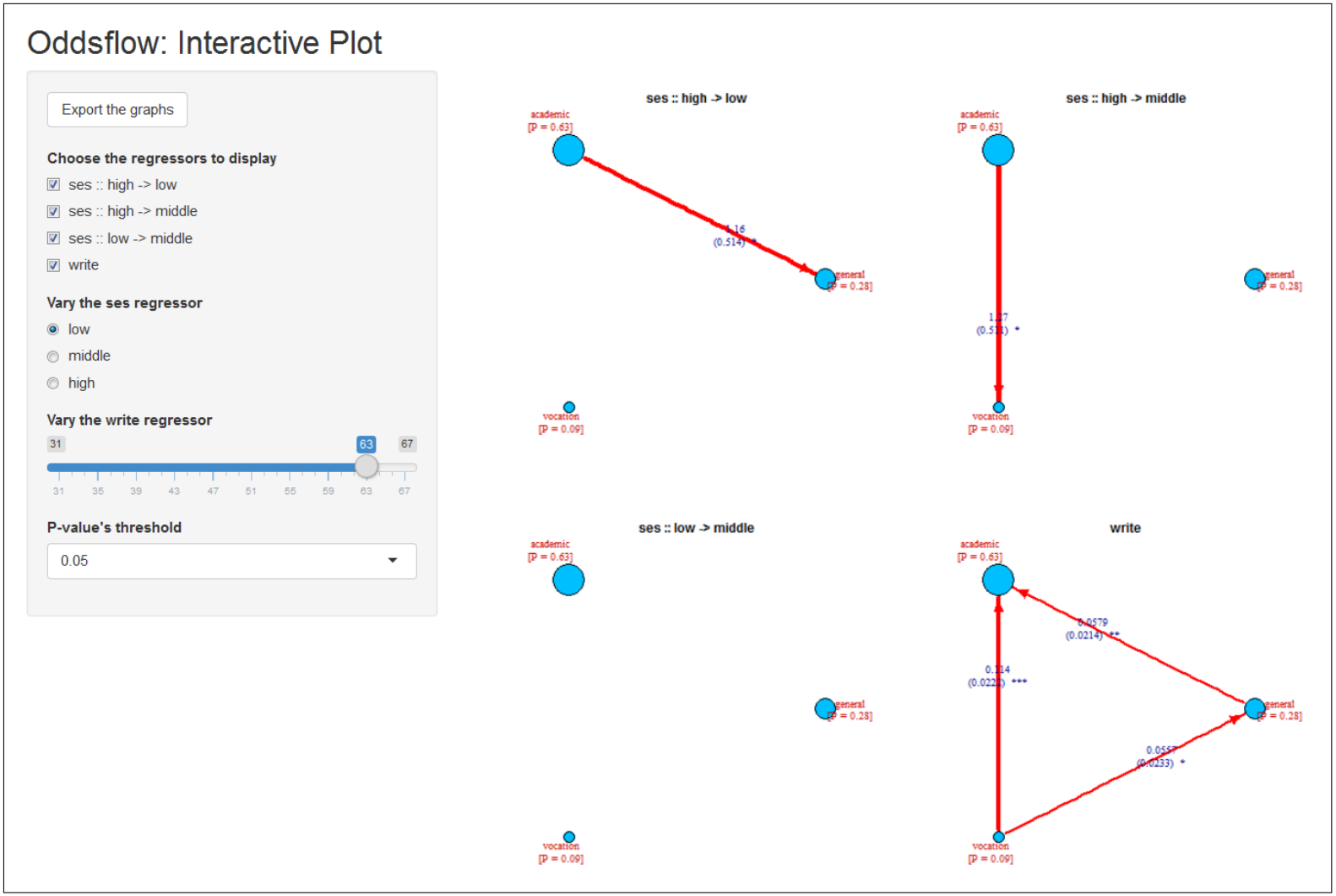


Figure 4.11: Interactive `Oddsflow` plots with writing score = 63 and ses = low.

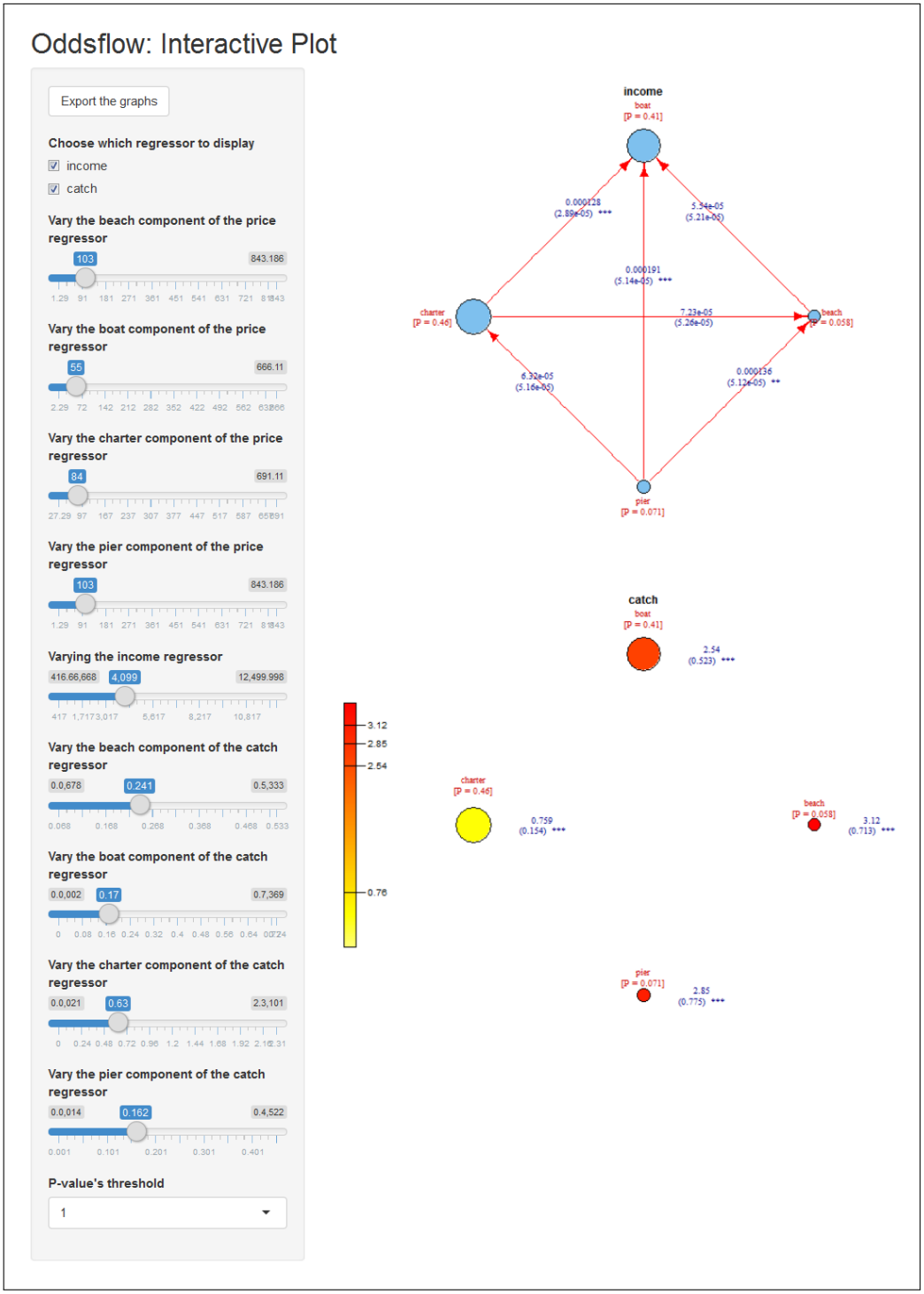


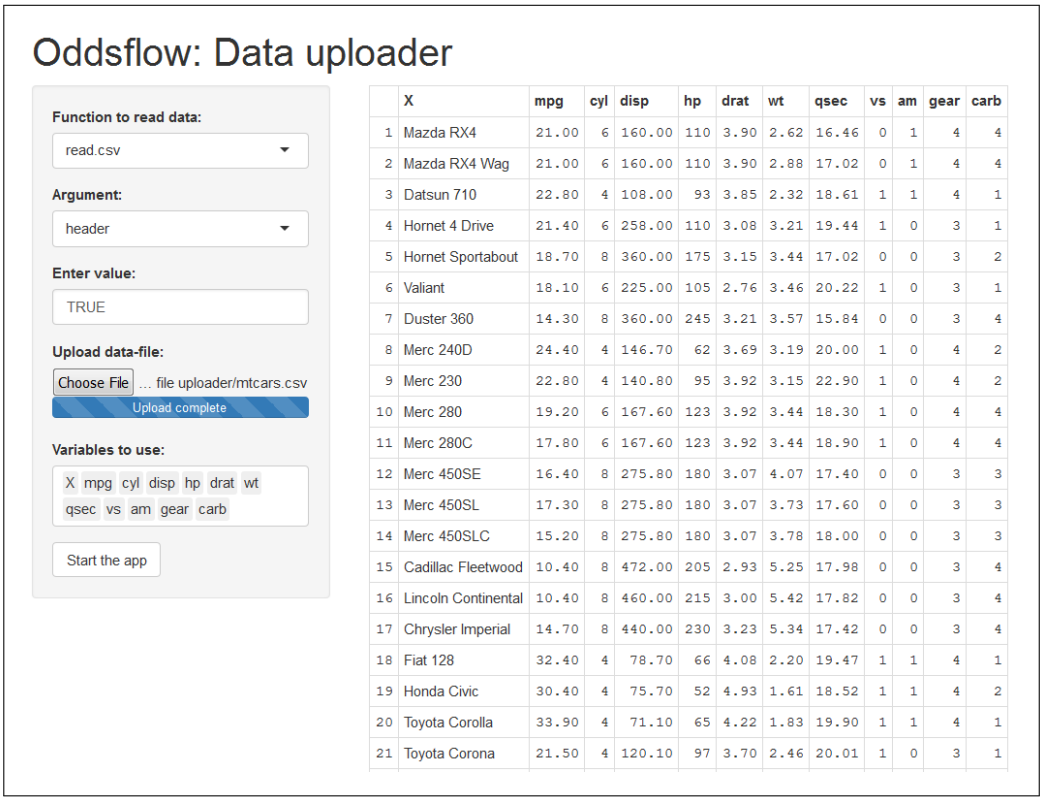
Figure 4.12: The interactive `Oddsflow` plots for the `Fishing` database.

4.3 Data Uploader

In the software is also included a data uploader to make the web application the only interface. In this way the passage of loading the database via the R console is completely hidden, and the user is not requested to give any lines of code to start the `Oddsflow` app.

Figure 4.13 shows how it works, testing the tool on a cars database. The first widget, "Function to read data", details which function should be used to read the submitted database. Then, for each argument of the chosen function, the user can choose to leave the default setting, or to change the value in the "Enter value" box. In the right panel the database appears as if it was read by R, so every flaw in the data reading can be corrected on the fly.

It is even possible to select which variables exclude from the database, just by deleting the unwanted ones in the "Variables to use" box.



	X	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
1	Mazda RX4	21.00	6	160.00	110	3.90	2.62	16.46	0	1	4	4
2	Mazda RX4 Wag	21.00	6	160.00	110	3.90	2.88	17.02	0	1	4	4
3	Datsun 710	22.80	4	108.00	93	3.85	2.32	18.61	1	1	4	1
4	Hornet 4 Drive	21.40	6	258.00	110	3.08	3.21	19.44	1	0	3	1
5	Hornet Sportabout	18.70	8	360.00	175	3.15	3.44	17.02	0	0	3	2
6	Valiant	18.10	6	225.00	105	2.76	3.46	20.22	1	0	3	1
7	Duster 360	14.30	8	360.00	245	3.21	3.57	15.84	0	0	3	4
8	Merc 240D	24.40	4	146.70	62	3.69	3.19	20.00	1	0	4	2
9	Merc 230	22.80	4	140.80	95	3.92	3.15	22.90	1	0	4	2
10	Merc 280	19.20	6	167.60	123	3.92	3.44	18.30	1	0	4	4
11	Merc 280C	17.80	6	167.60	123	3.92	3.44	18.90	1	0	4	4
12	Merc 450SE	16.40	8	275.80	180	3.07	4.07	17.40	0	0	3	3
13	Merc 450SL	17.30	8	275.80	180	3.07	3.73	17.60	0	0	3	3
14	Merc 450SLC	15.20	8	275.80	180	3.07	3.78	18.00	0	0	3	3
15	Cadillac Fleetwood	10.40	8	472.00	205	2.93	5.25	17.98	0	0	3	4
16	Lincoln Continental	10.40	8	460.00	215	3.00	5.42	17.82	0	0	3	4
17	Chrysler Imperial	14.70	8	440.00	230	3.23	5.34	17.42	0	0	3	4
18	Fiat 128	32.40	4	78.70	66	4.08	2.20	19.47	1	1	4	1
19	Honda Civic	30.40	4	75.70	52	4.93	1.61	18.52	1	1	4	2
20	Toyota Corolla	33.90	4	71.10	65	4.22	1.83	19.90	1	1	4	1
21	Toyota Corona	21.50	4	120.10	97	3.70	2.46	20.01	1	0	3	1

Figure 4.13: The `Oddsflow` data uploader interface.

Chapter 5

Examples

In this section some examples to test the `Oddsflow` app's features are proposed. The first two examples do not actually describe the choice of a decision maker, but the discrete choice models framework includes also the model of multinomial logistic regression, which is a perfect tool to solve some problems that may emerge from these collection of data. Furthermore, these examples show two different approaches to a problem that a user can engage with the `Oddsflow` app.

On the other hand, the third example displays how the software works when a data set with alternative specific variables is submitted, but does not enhance its analysis.

5.1 The `iris` Database

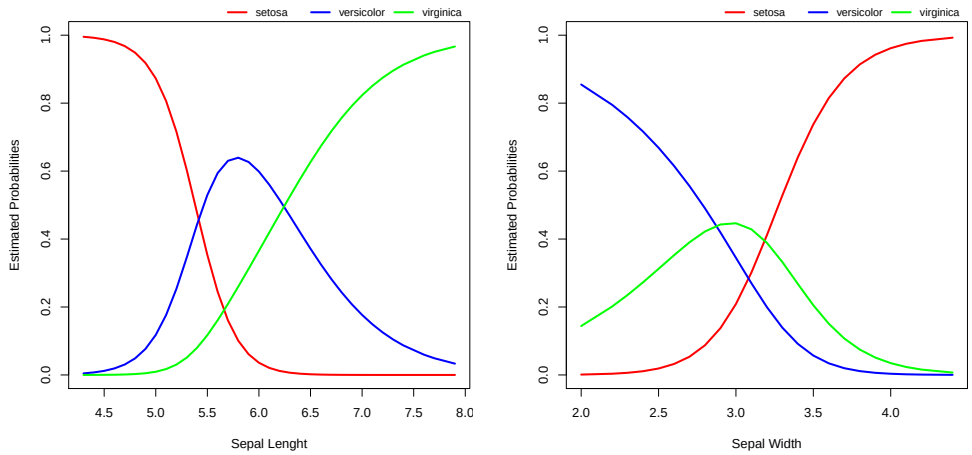
The `iris` database is a benchmark database, often proposed as example to learn the difference between supervised and unsupervised techniques in data mining. Since we expect it is well-known by the reader, it seems the appropriate subject of our demonstration.

The database has been introduced by Fisher (1936) and collected to quantify the morphologic variation of *Iris* flowers of three related species. The data consists of 50 samples from each of three species of *Iris* (*Iris setosa*, *Iris versicolor* and *Iris virginica*). Four features were measured from each sample: the length and the width of the sepals and petals.

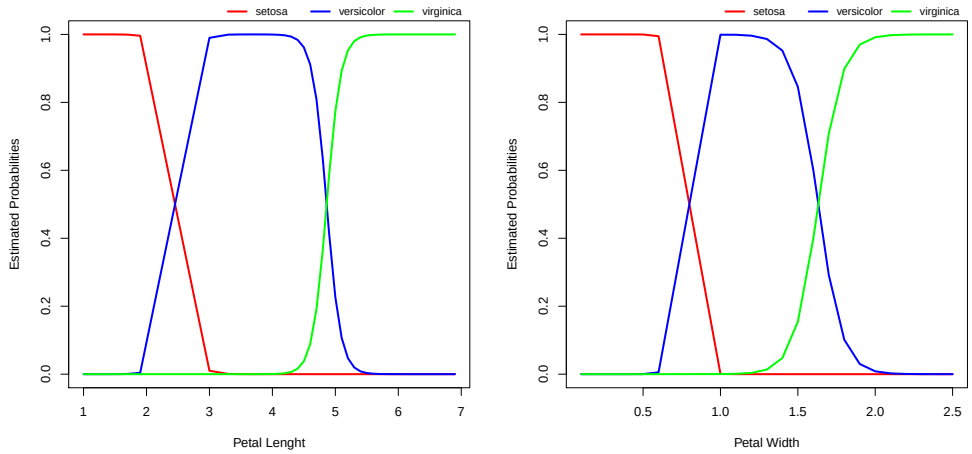
Instead of a multivariate regression model with all the continuous explanatory variables, four univariate models were chosen to first describe the trend of estimated probabilities. In fact, for each regressor of a multivariate model, there would be an infinite number of possible plots, given by all the possible values that the other regressors may take. Focusing only on

one of them is often pointless, and the loss of efficiency estimating the logistic regression coefficients with separate binary models is, for low number of explanatory variables, acceptable (Begg and Gray, 1984).

Figure 5.1 depicts the estimated probability plots for all explanatory variables in the dataset.



(a) Sepal Length (b) Sepal Width



(c) Petal Length (d) Petal Width

Figure 5.1: Estimated Probabilities for the `iris` database.

According to the petals' variables, the plots show that for some range of values a new flower has probability close to 1 to fall in one of the three classes. Furthermore, there are not values for which more than one estimated probability is non negative, so it is rather uncommon that a decision maker

(or a software) would ever choose among all the three alternatives. Two separate binary models (e.g. *Iris setosa* vs. not-*setosa* and *Iris versicolor* vs. not-*versicolor*) would be probably enough to estimate parameters. On the other side, the probabilities related to the sepals' features have usual shaped curves, for which a multinomial regression model seems adequate.

We proceed with model selection summoning the `Oddsflow` app on `iris` database (Figure 5.2).

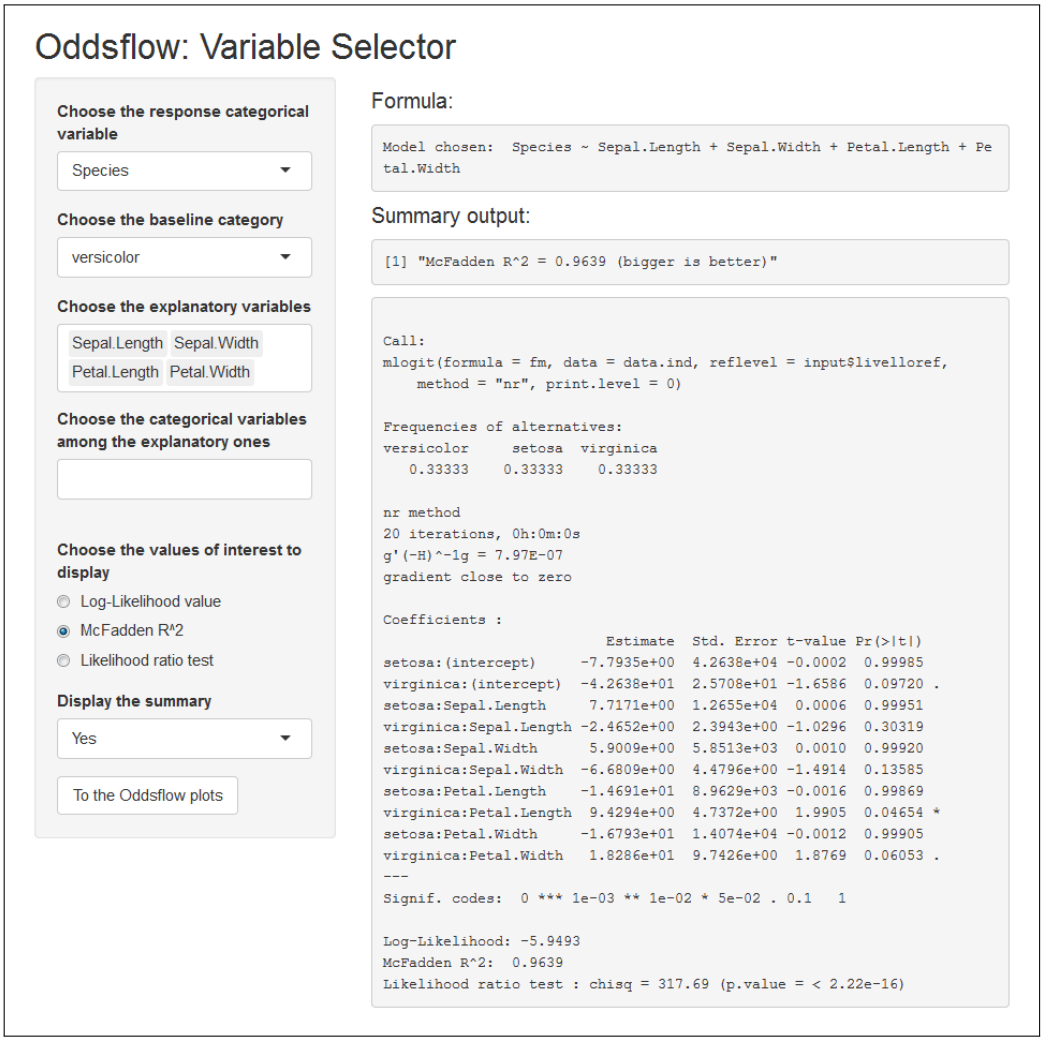


Figure 5.2: The first step of a variable selection procedure with `Oddsflow` app on the `iris` database.

The statistically relevant terms of regression appear to be the ones related to the alternatives *versicolor* and *virginica*, while the log odd between

versicolor and *setosa* seems to be equal to 1 (both probabilities P_{setosa} and $P_{versicolor}$ equal to 0.5, neglecting the *virginica* contribute due to IIA). This model is not able to discern among the first alternative and the other two, so it has to be changed to consider all the categories.

Being aware of the unusual behavior of the petals' features, a candidate model is the one including just sepals' features. The R summary, that can be also obtained with `Oddsflow` Variable Selector just by deleting the petals' variables in the "Choose the explanatory variables" widget, is the following:

```
R> fm  <- mFormula(Species ~ 0 | Sepal.Length + Sepal.Width | 0)
R> fit <- mlogit(fm, iris, reflevel = "versicolor")
R> summary(fit)

Call:
mlogit(formula = fm, data = iris, reflevel = "versicolor",
    method = "nr", print.level = 0)

Frequencies of alternatives:
versicolor     setosa  virginica
   0.33333    0.33333    0.33333

nr method
22 iterations, 0h:0m:0s
g'(-H)^-1g = 8.27E-07
gradient close to zero

Coefficients :
                          Estimate  Std. Error t-value  Pr(>|t|)
setosa:(intercept)       364.98917 44976.86104  0.0081  0.993525
virginica:(intercept)    -13.04603     3.09739 -4.2119 2.532e-05 ***
setosa:Sepal.Length     -136.76778 15708.53100 -0.0087  0.993053
virginica:Sepal.Length     1.90238     0.51692  3.6802  0.000233 ***
setosa:Sepal.Width       115.87073 15103.50484  0.0077  0.993879
virginica:Sepal.Width      0.40466     0.86283  0.4690  0.639078
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-Likelihood: -55.163
McFadden R^2:  0.66526
Likelihood ratio test : chisq = 219.26 (p.value = < 2.22e-16)
```

The Likelihood ratio test's P-value is close to 0, which means that the model fits properly the data, even if the McFadden R^2 is lower than the one corresponding to the complete model. But the model already reports a high predictive power[1], so a smaller R^2 will not compromise the above mentioned predictive power. On the basis of its P-values, the contribute of *Sepal.Width* is indubitably redundant. Excluding the variable gives the following model:

```
R> fm  <- mFormula(Species ~ 0 | Sepal.Length | 0)
R> fit <- mlogit(fm, iris, reflevel = "versicolor")
R> summary(fit)

Call:
mlogit(formula = fm, data = iris, reflevel = "versicolor",
    method = "nr", print.level = 0)

Frequencies of alternatives:
versicolor     setosa  virginica
   0.33333    0.33333    0.33333

nr method
7 iterations, 0h:0m:0s
g'(-H)^-1g = 0.000179
successive function values within tolerance limits

Coefficients :
                         Estimate Std. Error t-value  Pr(>|t|)
setosa:(intercept)       26.08176    4.88927  5.3345 9.582e-08 ***
virginica:(intercept)   -12.67706    2.90634 -4.3619 1.290e-05 ***
setosa:Sepal.Length      -4.81566    0.90684 -5.3104 1.094e-07 ***
virginica:Sepal.Length    2.03071    0.46567  4.3608 1.296e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-Likelihood: -91.034
McFadden R^2:  0.44758
Likelihood ratio test : chisq = 147.52 (p.value = < 2.22e-16)
```

The model is now fitting the data quite well (p.value close to 0), and has

[1] "values of 0.2 to 0.4 for rho-squared represent EXCELLENT fit", McFadden (1973, p.306).

a satisfying predictive power ($R^2 = 0.44758$). The analysis ends with the `Oddsflow` interactive plot in Figure 5.3.

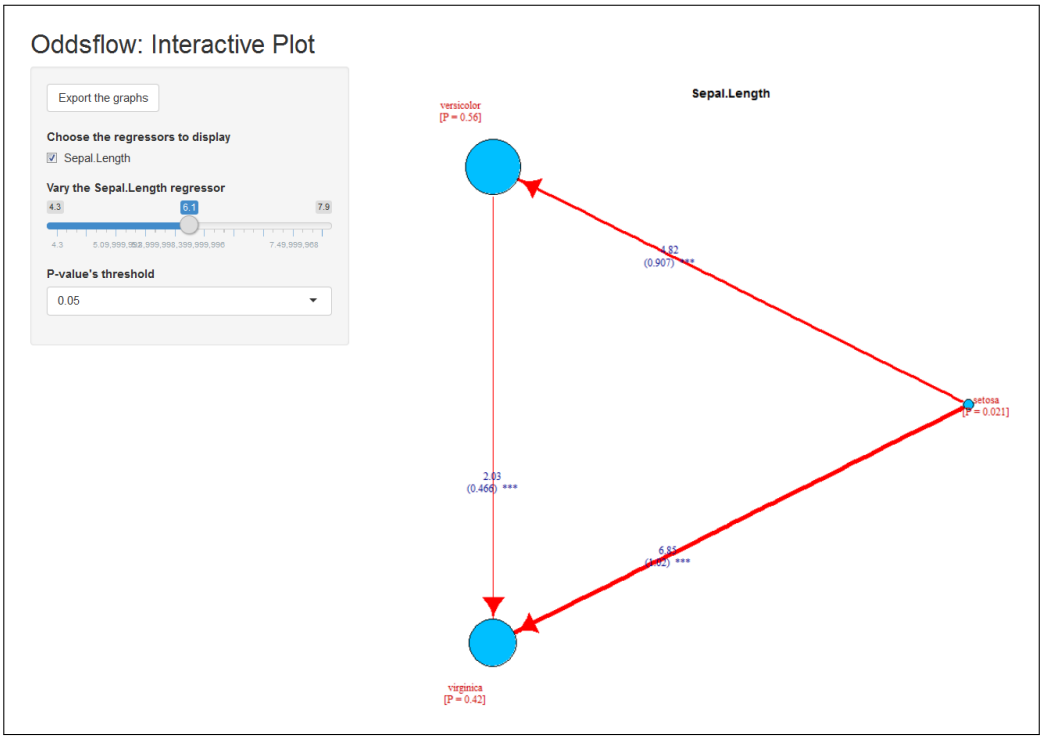


Figure 5.3: The `Oddsflow` interactive plot on the `iris` database.

As expected, the probability of a flower to belong the *setosa* species decreases when the flower's sepal length increases, while the probability of belonging the *virginica* species increases. Notice that in the graph the regression coefficient which multiplies *Sepal.Length* in the log odds between species *virginica* and *versicolor* is also represented, hidden in the previous summaries due to the choice of *versicolor* as baseline category.

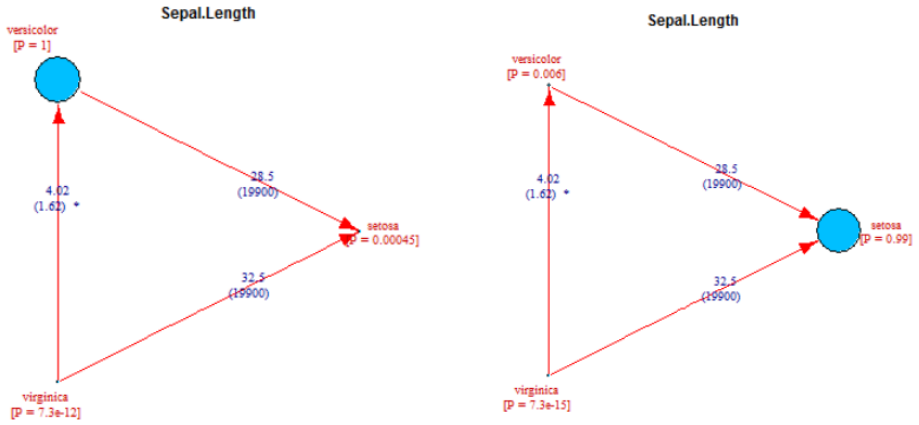
We now consider the model with also *Petal.Length* features and the relative `Oddsflow` interactive plot. Despite one of the two variables would be sufficient for fitting the model and make prevision, the McFadden R^2 rises meaningfully to 0.92763. The most interesting aspect to notice is how the petal variable's value influences the estimated the response probabilities when it falls in the ranges of certainty belonging to a class.

Figure 5.5 depicts how *Petal.Lenght* conditions estimated response probabilities in ranges identified in Figure 5.1c. Each *Petal.Length*'s value taken in these ranges corresponds to an estimated response probability close to 1

for a single alternative, and to 0 for the others. The *Sepal.Length*'s values considered will be the ones maximizing the expected probabilities of choosing one of the latter alternatives. As one can see from the plot, estimated probabilities in aforementioned ranges depend only by *Petal.Length*'s value.

Focusing on ranges where *Petal.Length*'s estimated probabilities are not fixed to 1, some differences with separate binary models' estimated probabilities emerge.

In Figure 5.4, *Petal.Width* takes value 2.5. According to Figures 5.1a and 5.1c, a *Sepal.Length*'s increment should favour species *versicolor* over *setosa*. Instead, as represented by the arrow directed from *versicolor* to *setosa* in Figure 5.4a, the probability flow switches in the opposite direction, and remains in species *setosa* for higher values of *Sepal.Length*.



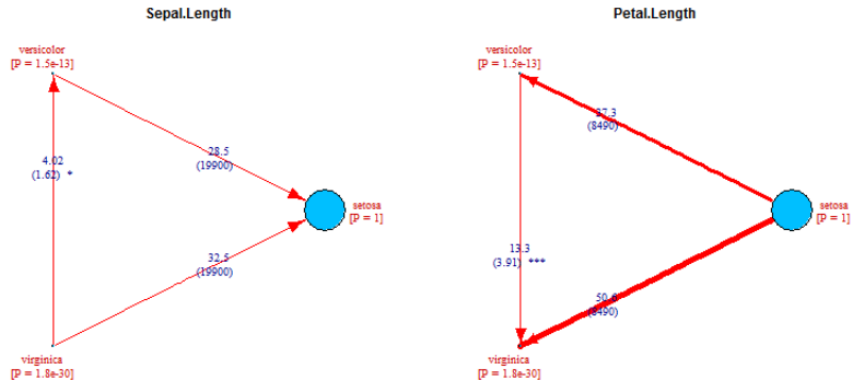
(a) *Sepal.Length* = 4.75, *Petal.Length* = 2.5

(b) *Sepal.Length* = 5.2, *Petal.Length* = 2.5

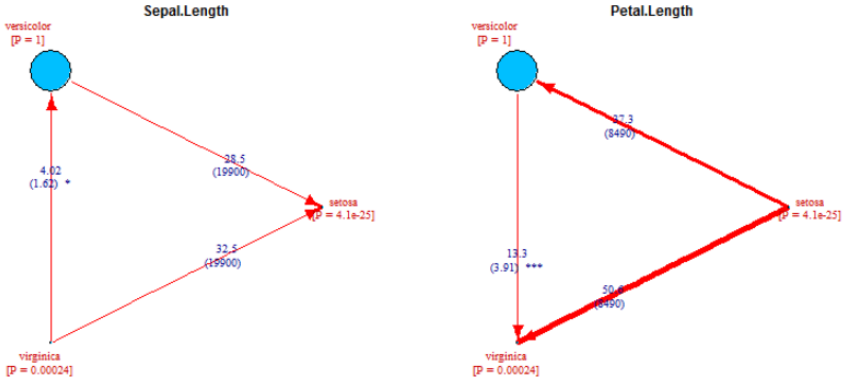
Figure 5.4: `Oddsflow` plots for interesting values of *Sepal.Length* and *Petal.Length* (2).

The case with *Petal.Length* taking values in interval $[4.2, 5.5]$ will not be depicted. As expected from both Figure 5.1 and `Oddsflow` plots, the probability flow courses from species *versicolor* and *setosa* to *virginica*, but slower than the previous example.

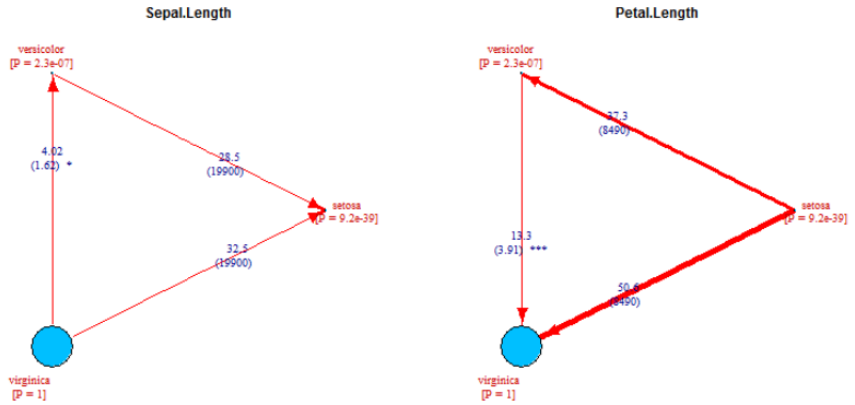
It is worth to mention that all these evaluations have been made without computing a single line of code, or plotting any static graph. All considerations and values comparisons have been done just switching the selectors of the `Oddsflow` app's widgets. Despite our analysis just scratched the problem's surface, at least was conducted with the least possible effort and in an intuitive and efficient way.



(a) *Sepal.Length* = 4.75, *Petal.Length* = 1.5



(b) *Sepal.Length* = 4.75, *Petal.Length* = 3.8



(c) *Sepal.Length* = 7.65, *Petal.Length* = 6.4

Figure 5.5: `Oddsflow` plots for interesting values of *Sepal.Length* and *Petal.Length* (1).

5.2 The Alligators' Food Choice Problem

The alligators' food choice problem, presented in Agresti (2002), consists in a study of factors influencing the primary food choice of alligators. It used 219 alligators captured in four Florida lakes. The response variable is the primary food type found in an alligator's stomach. Five types of food are considered to have been possible:

- fish
- invertebrate
- reptile
- bird
- other.

The choice is described by three explanatory variables:

1. *gender*: the gender of the captured alligator (categorical, takes value "male" and "female")
2. *size*: the size of the captured alligator (categorical, takes values "≤ 2.3" and "> 2.3")
3. *lake*: the lake where the alligator was captured: (categorical, takes values "Hancock", "Oklawaha", "Trafford" and "George").

Since all explanatory variables are categorical, the data set can be summarized by a contingency table, such as the one in Table 5.1.

The next step is the start of the `Oddsflow` app, which first interactive tool is the variable selector. The user is asked to choose among categorical variable the response, and among all other variables the covariates of the model that will be fitted. We will select the Food variable as categorical response due to the problem's request.

Our first model will be the one including all the disposable explanatory variables (Lake, Gender and Size). The model's summary output, same as the one displayed in the `Oddsflow` variable selector, is reported in Figure 5.6.

The P-value threshold above that we will consider a regressor's contribute statistically irrelevant is set at 0.05. The only non necessary covariate appears to be the gender. Comparing the previous model with the one featuring just lake and size (Figure 5.7), we notice a minor decrease in the $R^2_{McFadden}$ and still a very low likelihood ratio test's P-value. The new model fits quite

satisfying the data, and has the same predictive power as the previous (it is also less complex since a variable was dropped). It can be also noticed that all regressors seem to add meaningful information to the model. At least one estimated coefficient for each variable has Wald test's P-value lower than 0.05, confirming its statistical relevance in the log odds formulas.

The model exploited for the rest of the analysis is then the one describing the food content of alligator's stomach according to lake of capture and animal's size.

Table 5.1: Contingency table for the `alligator` data set.

Lake	Gender	Size (m)	Primary Food Choice				
			Bird	Fish	Invertebrate	Other	Reptile
George	Female	≤ 2.3	0	8	1	1	0
		> 2.3	0	3	9	1	1
	Male	≤ 2.3	1	9	0	2	0
		> 2.3	2	13	10	2	0
Hancock	Female	≤ 2.3	2	3	0	3	1
		> 2.3	2	16	3	3	2
	Male	≤ 2.3	1	4	0	2	0
		> 2.3	0	7	1	5	0
Oklawaha	Female	≤ 2.3	1	0	1	0	0
		> 2.3	0	3	9	2	1
	Male	≤ 2.3	0	13	7	0	6
		> 2.3	0	2	2	1	0
Trafford	Female	≤ 2.3	0	0	1	0	0
		> 2.3	1	2	4	4	1
	Male	≤ 2.3	3	8	6	5	6
		> 2.3	0	3	7	1	1

```
Coefficients :
                      Estimate Std. Error t-value Pr(>|t|)
invert:(intercept)    0.169024   0.378755  0.4463 0.655408
rep:(intercept)      -3.416038   1.085132 -3.1480 0.001644 **
bird:(intercept)     -2.432112   0.770664 -3.1559 0.001600 **
other:(intercept)    -1.430732   0.538094 -2.6589 0.007840 **
invert:size>2.3      -1.336261   0.411193 -3.2497 0.001155 **
rep:size>2.3          0.557036   0.646608  0.8615 0.388977
bird:size>2.3         0.730239   0.652280  1.1195 0.262919
other:size>2.3       -0.290583   0.459926 -0.6318 0.527515
invert:lakehancock   -1.780512   0.623211 -2.8570 0.004277 **
rep:lakehancock       1.129459   1.192800  0.9469 0.343691
bird:lakehancock      0.575266   0.795217  0.7234 0.469429
other:lakehancock     0.766575   0.568551  1.3483 0.177563
invert:lakeoklawaha   0.913182   0.476117  1.9180 0.055114 .
rep:lakeoklawaha      2.530256   1.122117  2.2549 0.024140 *
bird:lakeoklawaha    -0.550350   1.209867 -0.4549 0.649192
other:lakeoklawaha    0.026058   0.777771  0.0335 0.973273
invert:laketrafford   1.155822   0.492786  2.3455 0.019002 *
rep:laketrafford      3.061046   1.129729  2.7095 0.006738 **
bird:laketrafford     1.236990   0.866099  1.4282 0.153225
other:laketrafford    1.557763   0.625674  2.4897 0.012784 *
invert:genderm       -0.462963   0.395523 -1.1705 0.241796
rep:genderm          -0.627559   0.685276 -0.9158 0.359785
bird:genderm         -0.606429   0.688848 -0.8804 0.378668
other:genderm        -0.252569   0.466347 -0.5416 0.588100
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log-Likelihood: -268.93
McFadden R^2:  0.11003
Likelihood ratio test : chisq = 66.497 (p.value = 6.7234e-07)
```

Figure 5.6: Summary output for the complete regression model.

```
Coefficients :
                      Estimate Std. Error t-value  Pr(>|t|)
invert:(intercept)  -0.0908140  0.3080387 -0.2948 0.7681363
rep:(intercept)     -3.6657931  1.0589728 -3.4616 0.0005369 ***
bird:(intercept)    -2.7237365  0.7103953 -3.8341 0.0001260 ***
other:(intercept)   -1.5727214  0.4748222 -3.3122 0.0009255 ***
invert:size>2.3     -1.4582046  0.3959441 -3.6829 0.0002306 ***
rep:size>2.3         0.3512628  0.5800316  0.6056 0.5447853
bird:size>2.3        0.6306597  0.6424797  0.9816 0.3262957
other:size>2.3      -0.3315503  0.4482520 -0.7397 0.4595115
invert:lakehancock  -1.6583586  0.6128772 -2.7059 0.0068128 **
rep:lakehancock      1.2427742  1.1854319  1.0484 0.2944670
bird:lakehancock     0.6951176  0.7812635  0.8897 0.3736081
other:lakehancock    0.8261962  0.5575405  1.4819 0.1383779
invert:lakeoklawaha  0.9372193  0.4719043  1.9860 0.0470292 *
rep:lakeoklawaha     2.4588695  1.1181263  2.1991 0.0278709 *
bird:lakeoklawaha   -0.6532062  1.2020891 -0.5434 0.5868596
other:lakeoklawaha   0.0056531  0.7765743  0.0073 0.9941919
invert:laketrafford  1.1219848  0.4905126  2.2874 0.0221741 *
rep:laketrafford     2.9352509  1.1164090  2.6292 0.0085589 **
bird:laketrafford    1.0877668  0.8416688  1.2924 0.1962211
other:laketrafford   1.5163687  0.6214347  2.4401 0.0146828 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Log-Likelihood: -270.04
McFadden R^2:  0.10636
Likelihood ratio test : chisq = 64.283 (p.value = 9.7842e-08)
```

Figure 5.7: Summary output for the model with Lake and Size.

Once chosen the model to fit, the `Oddsflow` app can be started to compare interactive graph plots among all variables selected. In this model they are all categorical, so the interface will allow the user to display simultaneously six graphs for Lake regressor and one graph for Size. The Lake covariate has four alternatives, and all the possible switches from an option to another have to be compared. According to the Equation 3.2, there will be six plots describing all possible switches among them. Figure 5.8 depicts the `Oddsflow` app with all variables' graphs displayed and a 0.05 P-value threshold.

The first two graph plots show an interesting behavior. Switching from a small size to a bigger one, the probability for an alligator to feed with invertebrates decreases, independently from the lake in which it has been captured. The same states when we look for the probability flow of log odds between lake Hancock and George: the fact that all the edges leave the invertebrate vertex means that switching from the former lake to the latter, the probability that the primary food found in the alligator's stomach is mainly invertebrate decreases. Figures 5.9 and 5.10 shows the passage from the two options.

We may also focus our attention on those contributes with Wald test P-value < 0.001. Figure 5.11 displays only `Oddsflow` plots with at least a significative coefficient's regressor, and only coefficients that satisfy the above-mentioned bound. Comes to light that all the statistically relevant terms involve invertebrate, fish and other options. Furthermore, the only plots regarding the Lake variable with a necessary coefficient describe the switch between lake Hancock and another lake. There are not any other relevant probability flow among pair of lakes in which one of those is not lake Hancock.

All these considerations could lead an user focused on the simplification of the model to collapse the set of food choice to just invertebrates, fish and other kind of food, and even collapse the range of lakes in a binary set containing lake Hancock and a representative variable "Other lakes" that gather all other lakes in the database.

As in the `iris` example, this quick study was conducted just evaluating the output of `Oddsflow` plots and changing the P-value's threshold of coefficient visualized.

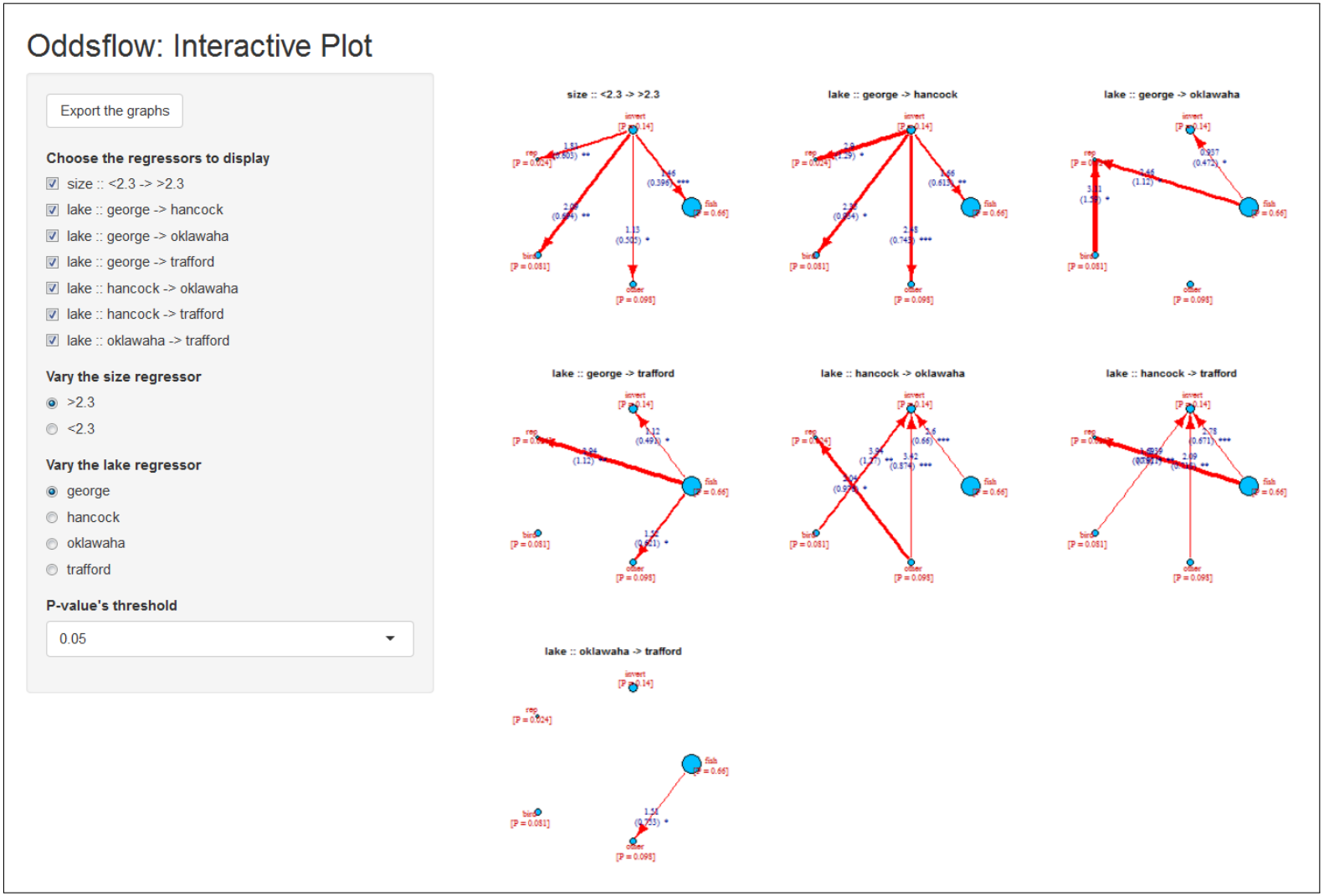
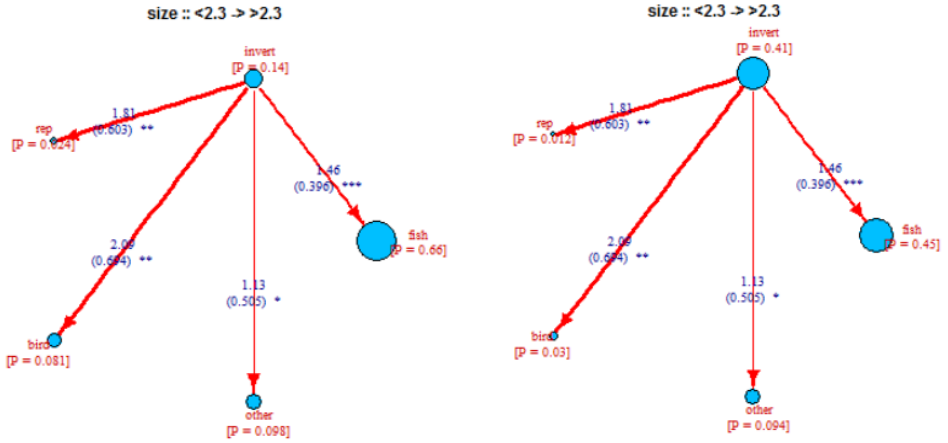
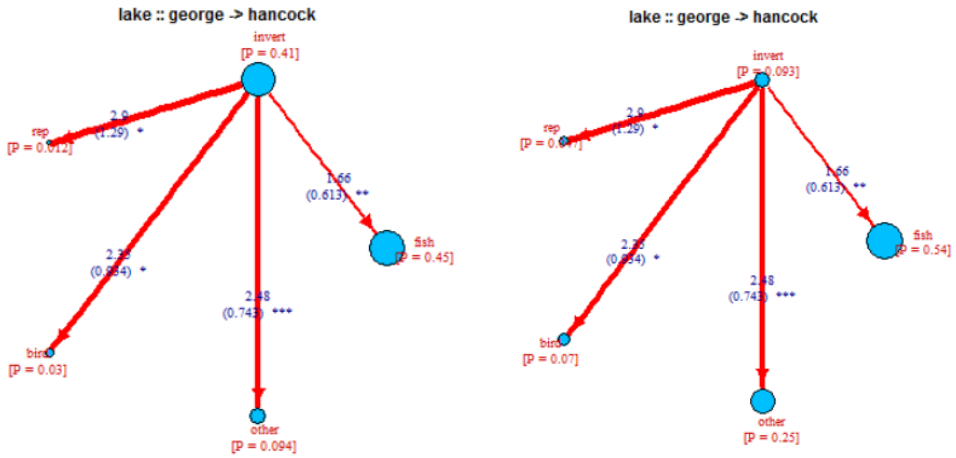


Figure 5.8: `Oddsflow` for the Alligator's Food Choice Problem.



(a) Size ≤ 2.3 (b) Size > 2.3

Figure 5.9: `Oddsflow` plots for Lake George and varying Size.



(a) Lake George (b) Lake Hancock

Figure 5.10: `Oddsflow` plots for varying Lake and Size ≤ 2.3.

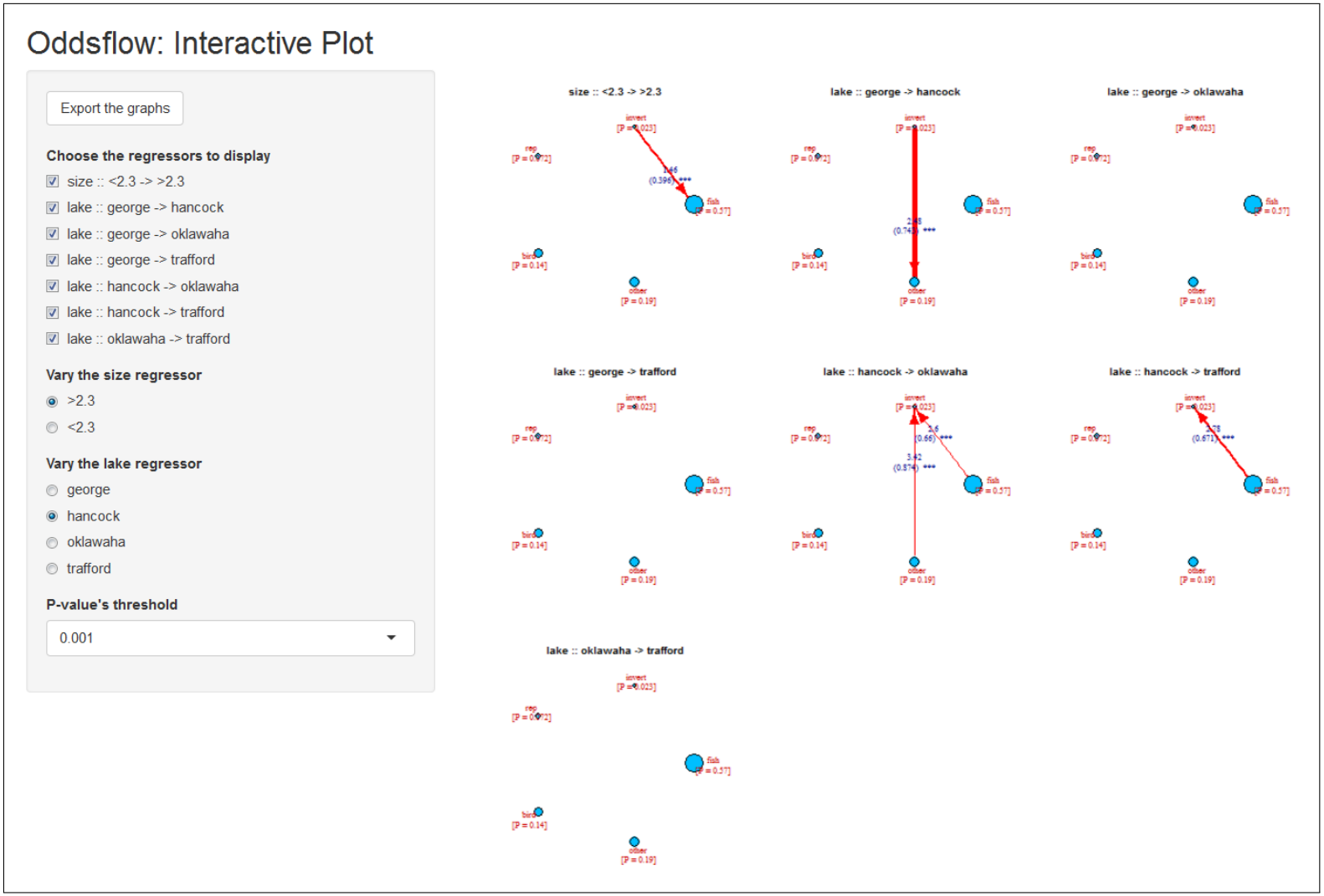


Figure 5.11: `Oddsflow` plots with coefficients with Wald test's P-value ≤ 0.001.

5.3 The TravelMode database

The TravelMode database (Greene, 2003) is a data collection for choices of individuals for a transport mode for inter-urban trips in Australia. The data set contains 7 variables for 840 observations. Four types of travel modes are considered to have been possible:

- car
- air
- train
- bus

For this example 6 variables will be considered:

1. *mode*: the 4 possible travel modes
2. *vcost*: vehicle cost measure
3. *travel*: travel time in the vehicle
4. *gcost*: generalized cost measure
5. *income*: household income
6. *size*: party size

The variables vcost, travel and gcost are alternative specific variables, so there are five values of them for each entry of the data set (one for each of the four travel modes). The remaining variables (income and size) depend only on the individual and have one valued shared among each alternative for each individual, then are individual specific. Figure 5.12 depicts the Oddsflow variable selector for the database and the summary output for the model:

$$mode \sim vcost + gcost \mid income + size \mid travel. \tag{5.1}$$

Equation (5.1) describes a model in which the travel mode is estimated with vcost and gcost as alternative specific variables with generic coefficients; income and size as individual specific variables with alternative specific coefficients; travel as alternative specific variable with alternative specific coefficients.

Variables vcost and gcost are both costs, then a constant regression coefficient is the most appropriate choice for them. Money spent on one alternative are in fact not qualitatively different from money spent on all the others.

Furthermore, the influence of these variables on the estimated probabilities is not depicted in the `Oddsflow` app, because they share a common constant coefficients with all the response's alternatives and their variations affects equally all the alternatives.

Conversely, travel time can be spent differently if the decision maker chooses a trip on board of car (that they will probably drive), or on a train or on a bus (that they will not drive). The specification ensured by the alternative specific coefficients is then necessary for this variable.

Figure 5.13 and 5.14 depicts the `Oddsflow` interactive plots for the variables income, size and travel. Some information can be extracted from these plots for the individual specific variables income and size. Passengers with a high income are more prone to choose the airplane to join their destinations, and a decrease in the income value raises the probability of taking the bus or the train. On the other side, numerous parties will choose with a higher probability the car or the train, leaving the airplane and the bus for smaller companies.

As expected, the alternative specific coefficients of the travel variable are negative. An increase in the travel time choosing any possible mode will decrease the probability that the relative transport mode is chosen by the decision maker.

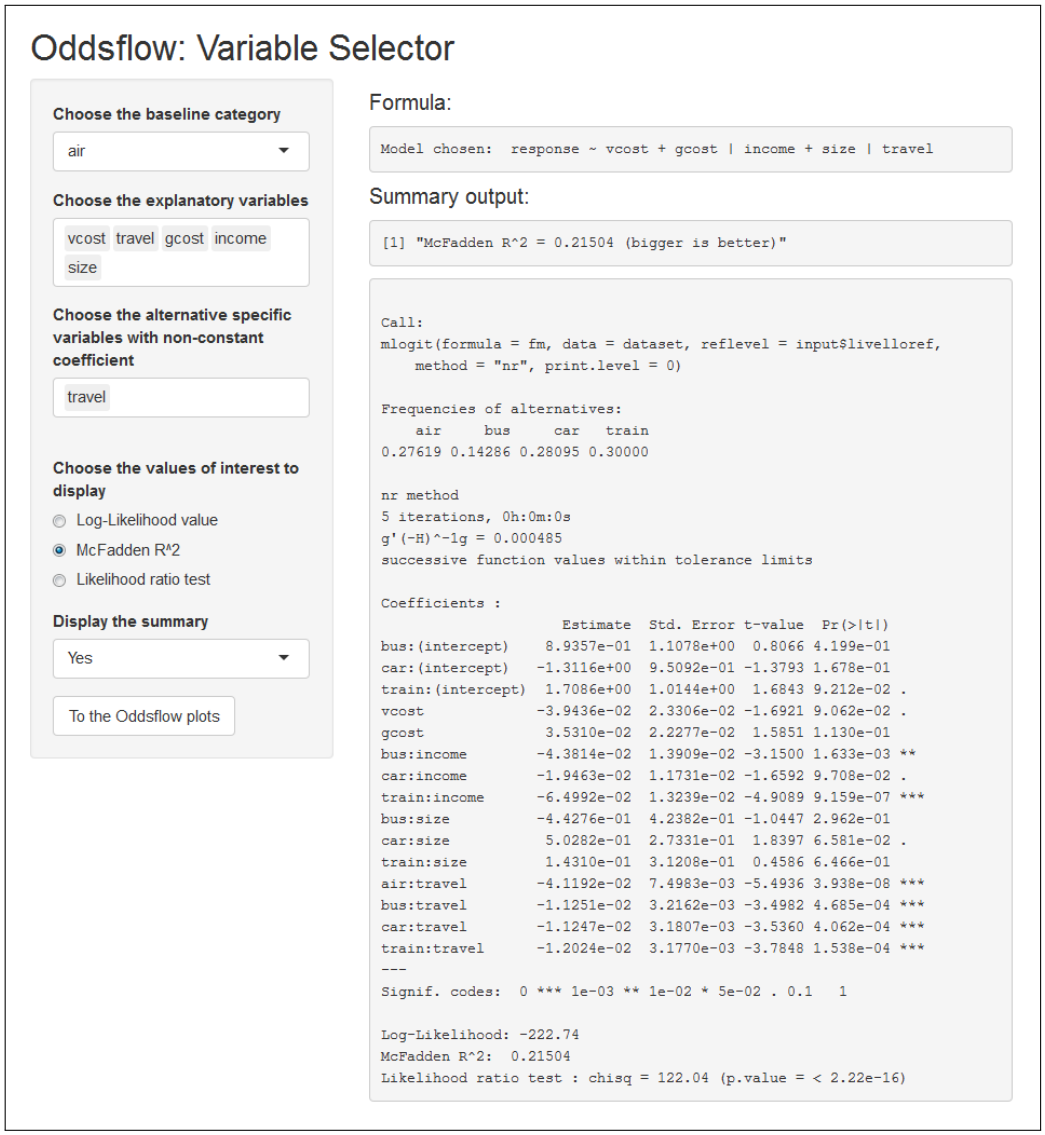
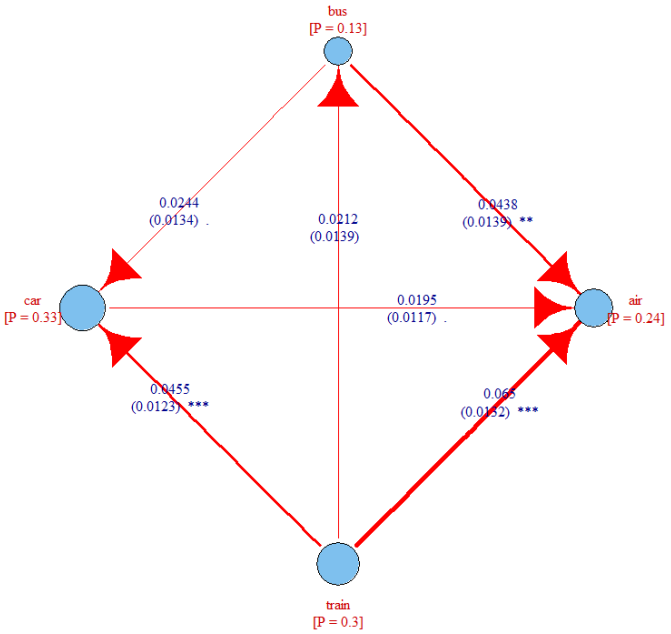
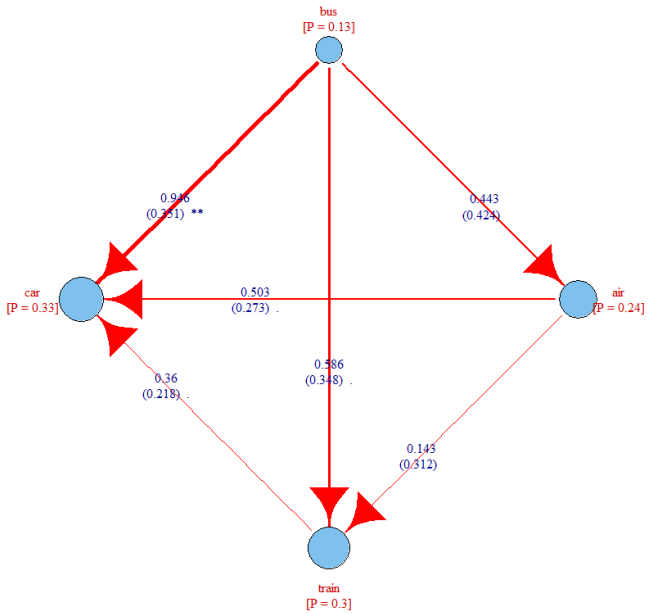


Figure 5.12: The `Oddsflow` variable selector for the `TravelMode` database.



(a) income



(b) size

Figure 5.13: The inteactive `Oddsflow` plot for the individual specific variables in the `TravelMode` database.

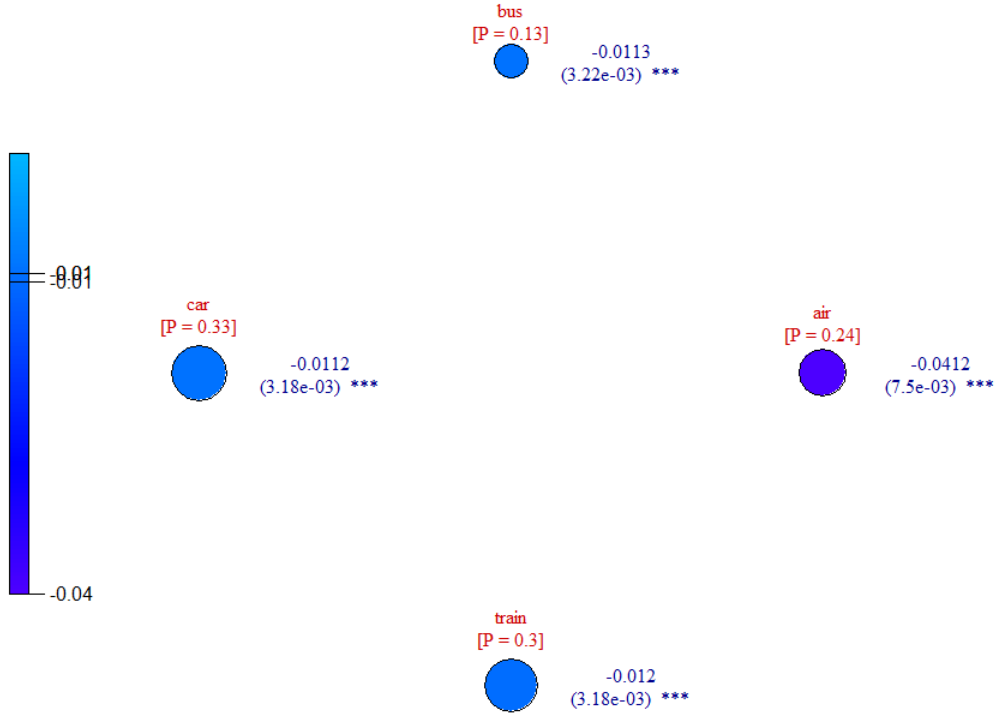


Figure 5.14: The inteactive `Oddsflow` plot for the train variable with alternative specific coefficients in the `TravelMode` database.

Chapter 6

Discussion

What makes the `Oddsflow` app unique are the various ideas which is born and designed for. The first impulse came from the growing interest in decision choice methods developed in the last decades. These days, information is massively collected with cutting edge technologies and devices, and these big sets of data need proper techniques to be studied. Among the vastness of available methods, discrete choice models are always more frequently appreciated for their inherent ability to both handle the so-called *big data*, and to efficiently describe a decision maker's process of choosing.

In addition to that, we encounter a shortage in graphical representation regarding multinomial logistic regression, a well-known method expressible with the discrete choice's equations. The available tools to represent the results of an aforementioned regression are few and based on binary logistic regression. It can be exploited for multinomial problems with small error, but sometimes these discrepancies cannot just be acceptable. Furthermore, an high number of alternatives constrains the generation of an exponential number of plots, making all the comparisons and evaluations burdensome.

The idea that we decided to represent is a probability flow that courses between the choice's alternatives. An excellent way to describe a network of this kind, which bounds all the options together, is the graph plot. For this reason, `Oddsflow` plots are designed as graph plots depicting the influence of each variable on the computation of estimated probability.

Finally, the `Oddsflow` app's interactivity allows to avoid a lot of lines of code and plots: variables in models can be selected and deselected just by clicking on them in the dedicated widget, and the same variables change with a cursor's scrolling. Estimated probability are computed and displayed instantaneously on top of nodes which areas increase and decrease.

All these features are meant to be appreciated by a wider group of persons than the usual target of a specific statistical software. This means that

everyone who is interested in or commissioned to approach a choice problem, with the `Oddsflow` app can perform his or her task with a single tool and without all the knowledge necessary to read and understand a programming language. The user's prerequisites are just the theory basis needed to face the problem, and not necessarily how to explain a computer to solve it, nor to interpret its output.

But if `Oddsflow` were just useful to compute estimated probability, the graph shape would not be needed. A pie chart would be probably enough. Its inner strength resides in the utility of the various log odds represented by edges in the `Oddsflow` plots and the possibility of plotting and varying the covariates in the same figure.

Few examples in Chapter 5 illustrate how information can be gathered with the `Oddsflow` app. Consider the `iris` database's problem of which species belongs a flower on the basis of its petal and sepal's features. For this problem emerged that for some models the binomial logistic representation's procedure leads to some errors in the estimated probability; we found that by confronting pairs of `Oddsflow` plots and the relative estimated probability plots derived by binary logistic regressions.

In the case of the alligators' food choice problem, instead, we represented all he model variables' `Oddsflow` plots in one figure. It helped us to discover that the model could be simplified grouping some response's alternatives in one option, and collapse the range of lakes in which the alligators were captured in a binary set containing just one lake (Hancock) and a representative element for all the others.

The directions towards point to carry forward our work are boundless. An useful improvement might regard the `Oddsflow` Variable Selector. It can be automatized giving the user an option to choose a stepwise variable selection, in addiction to the comparison of different models by generating and inspecting their summary output.

Another feature that can be improved is the `Oddsflow` plot for alternative specific variables. We decided to keep a sort of graph structure to avoid a sudden and confusing change when the user moves from an `Oddsflow` plot of an individual specific variable to one of an alternative specific. Conserving the nodes' positions in the plot might also help the comparison between different `Oddsflow` plots. Nonetheless, the network organization does not seem proper to display estimated probability proportional to the variable's option value. There are probably better shapes to accomplish the task, and should be evaluated even if they involve a continuity breaking.

Eventually, the graphical render of the graph plots should be enhanced. Unfortunately that does not depend on our software, but it is imputable to the `igraph` package. This is the most versatile tool to represent networks

in R, but still has some alignment and label arrangement problems. One solution might be using other programming languages (like C++ or Phyton) to write the plotting and interactive part of the software, and summoning R to estimate parameters of the models. That would mean dropping the Shiny package, a tool that during our work demonstrated its flexibility and reliability, for a greater design freedom at the price of increasing difficulty in the software programming.

Bibliography

Agresti, A. *Categorical Data Analysis*. New York: John Wiley & Sons, second edition, 2002.

Begg, C. B. and Gray, R. Calculation of polychotomous logistic regression parameters using individualized regressions. *Biometrika*, 71(1):11–18, 1984. URL `http://biomet.oxfordjournals.org/content/71/1/11.full.pdf`.

Biggs, N., Lloyd, E., and Wilson, R. *Graph Theory*. Oxford University Press, 1986.

Cameron, A. C. and Trivedi, P. K. *Microeconometrics : methods and applications*. Cambridge, 2005.

Croissant, Y. Estimation of multinomial logit models in r: The mlogit packages, 2013. URL `https://cran.r-project.org/web/packages/mlogit/mlogit.pdf`.

Csardi, G. and Nepusz, T. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006. URL `http://igraph.org`.

Fisher, R. A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936. URL `http://rcs.chemometrics.ru/Tutorials/classification/Fisher.pdf`.

Greene, W. *Econometric analysis, 5th edition*. Prentice Hall, Upper Saddle River, NJ, 2003.

Hoffman, S. D. and Duncan, G. J. Multinomial and conditional logit discrete-choice models in demography. *Demography*, 25(3):pp. 415–427, 1988. URL `http://www.jstor.org/stable/2061541`.

Kohavi, R. Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on*

Knowledge Discovery and Data Mining, pages 202–207. AAAI Press, 1996. URL https://www.aaai.org/Papers/KDD/1996/KDD96-033.pdf.

Luce, R. D. *Individual Choice Behavior*. New York: Wiley, 1959.

McFadden, D. Conditionial logit analysis of qualitative choice behavior. *Frontiers in Econometrics*, pages 105–135, 1973. URL http://eml.berkeley.edu/reprints/mcfadden/zarembka.pdf. New York: Wiley.

Pearson, K. Mathematical contributions to the theory of evolution XIII; on the theory of contingency and its relation to association and normal correlation. *Draper's Co. Research Memoirs, Biometric Series*, no. 1, 1904. URL https://archive.org/details/cu31924003064833.

RStudio Team. *Shiny: Easy web applications in R*. RStudio, Inc., 2014. URL http://shiny.rstudio.com.

RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, Inc., Boston, MA, 2015. URL http://www.rstudio.com/.

UCLA: Statistical Consulting Group. R data analysis examples: Multinomial logistic regression. URL http://www.ats.ucla.edu/stat/r/dae/mlogit.htm. (Accessed: August 22, 2015).

Wald, A. Tests of statistical hypotheses concerning several parameters when the number of observations is large. *Transactions of the American Mathematical Society*, 54(3):pp. 426–482, 1943. URL http://www.jstor.org/stable/1990256.

Wilks, S. S. The likelihood test of independence in contingency tables. *Ann. Math. Statist.*, 6(4):190–196, 12 1935. URL http://dx.doi.org/10.1214/aoms/1177732564.

Zeileis, A. and Croissant, Y. Extended model formulas in R: Multiple parts and multiple responses. *Journal of Statistical Software*, 34(1):1–13, 2010. URL http://www.jstatsoft.org/v34/i01/.