# STATED PREFERENCE MODELING AND ANALYSIS OF MANAGED LANES

by

KAVEH FAROKHI SADABADI

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN CIVIL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2007

ACKNOWLEDGEMENTS

My deepest gratitude goes to my parents and my dear sister who gave me much needed emotional support during my studies at UTA. Also, I would like to express my humble appreciation of all great teachers in my country who contributed to my success and left in me the desire for learning as well as the dissemination of what I know.

My special acknowledgements are due to Dr. Stephen P. Mattingly, my advisor and mentor, who taught me the importance of patience and constructive critical thinking in research. The time and care spent by my other committee members Dr. Sia Ardekani and Dr. Jim Williams in revision of my work and their recommendations for the improvement of the final version of this thesis are sincerely appreciated.

The partial financial support provided by Graduate Research Assistant position offered by School of Urban and Public Affairs (SUPA) made it possible for me to pursue and complete my studies at UTA. Dr. Jianling Li's leadership was instrumental in the progress and success of TxDOT project 0-5286 which provided me with this assistantship. Her insight and devotion to the accuracy of analyses is acknowledged.

Finally, I wish to thank my friends; Auttawit, Saty, Chulsu, Sergio, Phong, Isaradatta, and Ali, at the Transportation Research Lab who maintained a friendly working environment.

November 13, 2006

ABSTRACT

STATED PREFERENCE MODELING AND ANALYSIS OF

MANAGED LANES

Publication No. _____

Kaveh Farokhi Sadabadi, M.S.

The University of Texas at Arlington, 2007

Supervising Professor: Dr. Stephen P. Mattingly

The thesis deals with the Managed Lanes demand estimation problem in the state of Texas. Managed Lanes are a generalized version of High Occupancy Toll roads, whose lane management and operational features may be dynamically adjusted.

A stated preference survey is designed in both English and Spanish to collect the required data. The survey was primarily conducted on two different internet websites for Dallas/Fort Worth metroplex and the Houston area during summer 2006. Each survey taker is presented with up to four scenarios. Scenarios are defined in terms of travel times in Managed Lanes versus General Purpose Lanes, as well as, the toll

charged for access to the Managed Lane. Respondents were asked to choose among three modes and two lane types. Out of 4634 collected responses, 2026 entries were from DFW and 2562 were from Houston.

Multinomial and Nested Logit models were estimated. Alternative Specific Multinomial Logit models were selected based on the overall fit and percent correct measures on the estimation and validation datasets.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

## 1.1 Problem Statement

During the 1970s carpooling was very much regarded as a promising solution to all of the perceived problems for urban transportation. However, it did not take long before all the hopes and enthusiasm in the practice declined during the 1980s and 90s. This decline happened even though specific policies were in place to encourage carpooling. The most notable example of this kind is the ISTEA, ratified in 1991, which favored high-occupancy vehicle (HOV) lane construction (Ferguson 1997; Jaskevich 2001; Poole and Balaker 2005).

The resulting divergence of the supply side efforts manifested in the increasing number of HOV lane miles built and the demand side's continuing decline in carpooling during the 1990s, led to a popular perception of HOV lane underutilization as well as calls by HOV opponents to open up the lanes to general traffic (Poole and Orski 1999; Economist 2000; Jaskevich 2001; Dahlgren 2002; Clements and Veldhuis 2003; Cooke 2005; Copeland 2005).

Coincidentally, historic budget constraints resulting in a backlog of needed transportation improvements provided transportation agencies with an incentive to consider applying road pricing principles to HOV lanes. The road pricing concept has long been considered to be capable of providing the optimal solution to the widespread

urban transportation problems, such as congestion and air pollution. Advances in electronic tolling technology, have also played a very important role in making managed lanes a reality in the last several years (Copeland 2005; Washington 2005).

Managed lanes are facilities where access eligibility is controlled by pricing policies and other considerations. Therefore, operation of managed lanes must include, among other things, setting desired objectives for the facility and detailed specification of user eligibility and pricing policies. Most managed lane facilities in the U.S. are currently run as high occupancy/toll (HOT) lanes, where single-occupancy-vehicles (SOVs) are charged a toll for using the facility, while providing the high-occupancy-vehicles (HOVs) with free access to travel on the facility.

However, results of recent research identified several inefficiencies associated with the HOT lane operation and called for eventually eliminating HOV preferential treatment. The research findings recommended that either HOV lanes should be opened up to all traffic, or they should be expanded to include all the existing lanes in the corridor by charging a toll on them, as well (Dahlgren, 2002; Varaiya, 2005; and Small, et al., 2006).

On the other hand, supporters of HOT lanes argue that the current policy promises a number of benefits, including promotion of non-SOV modes, especially carpools and transit modes, generating much needed transportation revenues, easing political opposition to roadway construction projects, and enhancing air quality (DeCorla-Souza et al. 2003; Hadley 2004; Regan 2004; Groat 2004; Connolly 2005; Eckenrode 2005; Poole and Balaker 2005; Poole and Soucie 2006).

The debate raises a series of questions about whether managed lanes are an appropriate alternative, whether HOVs should be given preferential treatment, and what are the potential impacts of different HOV preferential treatments.

## 1.2 Scope and Objectives

The answer to the questions raised in the previous section and, in general, a better understanding of the mode and lane choice behavior of road users when faced with a managed lane alternative may be obtained through development of a behavioral choice model. The model should have both descriptive and predictive capabilities and should include policy variables of interest to facilitate the objective evaluation of managed lanes.

Alternatively, the model can be used as a tool to design policy variables, which control the overall demand and operation of the corridor (i.e. managed and general purpose lanes). The output of the model may also be used in an impact study to evaluate traffic externalities, such as air pollution, caused by operation of managed lane in conjunction with general purpose lanes facilities. This can be performed by post-processing the choice model estimates.

The potential applications of this model suggest that it may play a central role in three different study aspects for a highway facility, namely planning, operation, and control. Thus, the insight provided by this choice model should be of importance to planners, operation managers and, in general, all decision makers at various levels who want to deal with a managed lane facility.

The thesis reports on the estimation of the choice model by examining responses to a stated preference survey of travelers in Dallas/Fort Worth (DFW) and Houston, the two largest metropolitan areas in the state of Texas, from summer 2006. Therefore, the model and any results related to it are limited to the State of Texas and more specifically to these two major population centers. Any generalization to other places and times should take into consideration that the model may not be an appropriate representation of the behavior for those geographies and or time periods.

## 1.3 Overview

The next chapter provides a literature review of the concept of HOV lanes and its evolution to managed lanes. Also, the second chapter contains a concise review of discrete choice and random utility theories.

Chapter three is specifically devoted to the methodology adopted for this study. The Logit model and its close offspring, nested Logit models, are reviewed. Estimation techniques, particularly the maximum likelihood technique, are discussed and measures for goodness of fit are introduced.

The specifics of survey design and implementation are presented in chapter four, which also demonstrates some findings from the preliminary statistical analyses performed on the collected data. In the latter sections, the weighting process and its application in the model estimation is discussed. A variety of calibrated models are presented and compared to select the final model among them and to draw any important conclusions.

Chapter five contains the examination of the final models and their implications to give better insights into the stated behavioral shift of road users in terms of mode choice and lane selection decisions when faced with various access restrictions and preferential treatment policies for managed lanes under various general circumstances. Furthermore, a number of recommendations for future studies in this area are presented.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 High Occupancy Vehicles and Managed Lanes

The definition provided by the Texas Department of Transportation (TxDOT) for managed lanes states that (Owens and Lewis, 2002):

*"A managed lane is a facility that increases freeway efficiency by packaging various operational and design actions. Lane management operations may be adjusted at any time to better match regional goals."*

The following points in this definition may deserve further elaboration. First, the stated goal of increasing *freeway efficiency* may be broadly translated into demand management, congestion mitigation, and reduction of environmental pollutants.

Second, the definition suggests that the method of achieving the above goals will include both *operational and design* features, which may encompass right-of-way separations, access/egress designs, access restrictions based on occupancy or vehicle type, time of the day and or toll charges.

Third, the operational features of this type of facility may be varied *at any time*. This is intended to bring about the type of flexibility that is seldom present in the operations of typical highway facilities.

Fourth, the importance of consistency in design and operation of this type of facility with the broader *regional goals* is emphasized. It should be noted that the tolling

potential of the managed lanes provides them with a unique ability to provide finances for construction and or maintenance of similar or different projects; thus, making them more appealing to decision makers.

Table 2-1 provides a summary of different goals and objectives, which may be pursued by consideration of a managed lane. The appropriate measures of effectiveness (MOEs) to help gauge the level of achievement of those objectives are also included.

Managed lanes are a natural generalization of the more familiar concept of HOV[1] lanes. Although two HOV lane variants; HOT[2] lanes and ETL[3] facilities, preceded the more sophisticated managed lane facilities, they are considered to be subsets of the latter. In addition to HOV facilities, truck-only lanes and limited-access express lanes are also included in the managed lane type facilities.

HOV lanes have been operational in North America for the past three decades. In 1969, Virginia's Shirley Highway (I-395) implemented an exclusive busway, but later a HOV facility emerged when four-or-more person carpools and vanpools were given access to the facility in 1973. HOV lanes, in essence, offer a reduction in travel time and cost to those who choose to carpool or vanpool. This, combined with high energy prices in the US and concerns over social and environmental problems regarding mobility, provide a set of incentives for people to leave their personal cars and carpool together.

[1] High Occupancy Vehicle
[2] High Occupancy / Toll
[3] Express Toll Lane

Table 2-1 Managed Lane Goals (Source: Burris et al., 2007)

| Goals | Objectives | Measures of Effectiveness |
|---|---|---|
| 1. Improve operational efficiency of the transportation system | Reduce congestion | Reduce average travel time |
| | | Compare average travel time of an HOV lane versus a ML |
| | | Percentage time GPL is LOS D or worse |
| | | Percentage of time the HOV or ML is operating in LOS D or worse |
| | | Average speeds |
| | | Travel time index (TTI) (a ratio of travel time in the peak period versus travel time in the off-peak period). |
| | Improve travel time reliability | Percentage of vehicles (ML and GPL) travel time less than 1.2 times Free flow travel time |
| | | Difference between 95th percentile travel time and 50th percentile travel time |
| | | Percentage of time vehicles achieve free flow speeds |
| | Maximize throughput and person carrying capacity | Number of vehicles per hour (ML and GPL) |
| | | Number of persons per hour (ML and GPL) |
| | | Increase in Average Vehicle Occupancy (AVO) and/or transit usage |
| 2. Provide more travel options to the users | Provide additional travel options | Count number of travel options (count number of vehicles/persons selecting new options) |
| | | Increase in AVO and/or transit usage |
| 3. Generate revenue | Generate net revenues | Calculate difference between revenue and costs for conversion to MLs |
| 4. Develop a sustainable transportation system | Reduce emissions | Calculate emissions (Nitrous oxides (NOx), VOC, Carbon monoxide (CO), Particulate matter) for both MLs and GPLs |
| | Reduce fuel usage | Calculate fuel usage for both MLs and GPLs |
| | Maximize use of existing infrastructure | Vehicle counts |
| | | Increase in average vehicle occupancy (AVO) and/or transit usage |
| | Pay for itself (operations and maintenance covered) | Revenue versus costs |
| | | Acceleration of construction |
| 5. Improve net societal benefits | Improve benefits to society | Calculate net societal benefits and costs. Costs include construction costs, operation and maintenance costs, and capital costs. Benefits include travel time savings, fuel savings, and emissions savings. |
| | | Acceleration of construction |
| 6. Enhance and support emergency management operations | Enhance and support emergency management operations | Reduced response time to emergencies |
| | | Additional evacuation route |

Since 1993, vehicle miles traveled have increased by 25 percent, while the percentage use and absolute number of carpools and vanpools for commute trips has declined to a thirty-year low –10,057,000 trips in 2003, down from 11,852,000 in 1993 (Bureau of Transportation Statistics, 2005). In the same 10-year timeframe, HOV lane miles have more than doubled, from approximately 1,300 lane miles in 1995 to over 2,500 in 2000, and are forecast to be over 3,200 by the end of 2006. The majority of these HOV lane miles are located in California (1,000), Georgia (400), and Texas (300) (Fuhs and Obenberger, 2005).

Access to HOV lanes is controlled by particular occupancy and use policies. Most freeway-based HOV lanes are open to vehicles with two or more passengers. The goal is to provide a level of demand that justifies the HOV lane without overloading it. The better traffic flow conditions on HOV lanes should give some SOV[4]s, an appropriate incentive to drive extra distance, if necessary, to pick-up a carpool partner and become eligible for HOV lane use. This suggests that there is a delicate trade-off between the time gained from using a HOV lane versus the extra time spent in forming a carpool and any other potential inconvenience involved in carpool formation.

In the past two decades, even though carpool rates have declined by thirty percent, carpooling inside HOV corridors has more than doubled (Stockton et al., 1999). The decline in the performance of HOV lanes due to higher demands has occasionally prompted increases in the level of exempt occupancies. This may contribute to an underutilization of HOV lanes, which combined with severe congested conditions in the

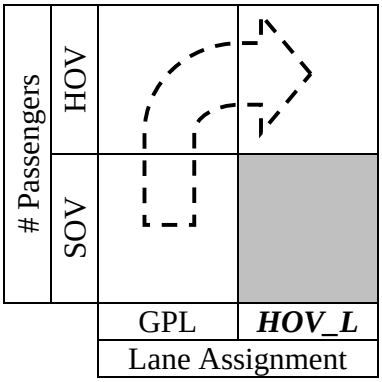
[4] Single Occupancy Vehicle

GPL[5]s, have fueled an animosity for underutilized HOV lanes because the lanes may provide more overall benefit by opening them to general traffic (Fuhs and Obenberger, 2005). In response, HOT lanes can be developed to sell the excess capacity of HOV lanes to SOVs who are willing to pay for the privilege of using them (Swisher et al., 2002).

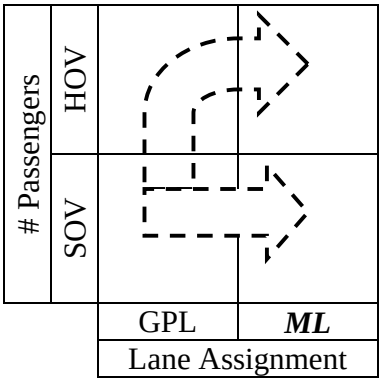
As stated earlier, managed lane facilities are generally perceived and implemented with a variety of operational and design features, which make them distinctive from each other. These features typically include system management techniques, such as time of day, vehicle type and or value pricing restrictions.

Figure 2-1 demonstrates a schematic of the decision space for road users in the presence of HOV lanes and ML. The user decision space in the case of HOV lanes suggests that access to these lanes is restricted to HOV modes, and for a SOV to gain access to the HOV lane, it must first become a HOV. However, in the ML case SOVs are not required to change their occupancies. Certainly, they have a choice between forming a carpool or paying the toll to use the ML facility.

[5] General Purpose Lane




(a)




(b)

Figure 2-1 Schematic Decision Space in the Presence of (a) HOV lanes, and (b) Managed Lanes

## 2.2 Discrete Choice Theory

A decision regarding the use of managed lanes and driving alone or forming a carpool can be studied using discrete choice theory. A choice can be viewed as the outcome of a sequential decision making process:

1. definition of the choice problem
2. generation of alternatives
3. evaluation of the alternatives' attributes
4. choice
5. implementation

In each choice process, certain entities and elements are present and influence the outcome. A specific theory of choice is a collection of procedures that defines the following elements:

1. decision maker
2. alternatives
3. attributes of alternatives
4. decision rule

### *2.2.1. Decision Maker*

A unit of decision making can be either an individual, a group of individuals (e.g. a family or a household), or a private or public organization. In the second case, the interactions of people involved may give rise to special processes in decision making, which may not be easily observed or understood.

*2.2.2. Alternatives*

Any single decision maker may have a number of alternatives for her decision problem which are not necessarily the same as the alternatives available to another individual. Each individual's alternatives are termed as her choice set, which is a subset of the universal set of alternatives available to all members of the population. It is rational to suggest that decision makers only consider alternatives as available to them that they know and consider to be feasible.

Swait (1984) discusses the role of environmental and personal constraints on the composition of the choice set. In general, there are two types of choice sets: continuous and discontinuous. The first type, basically, is comprised of the economically possible quantities of various commodities one considers for consumption. However, the discontinuous type choice sets are formed where the alternatives are by their own nature discontinuous. In other words, when the choice problem is about choosing one alternative and rejecting the rest of the alternatives then the decision maker is said to have a discontinuous choice set.

*2.2.3. Alternative Attributes*

Each alternative in a decision maker's choice set may have a vector of attributes; based on this vector, the attractiveness of that alternative may be evaluated. The attribute values may be measured on an ordinal or cardinal scale. For instance, the travel speed of travel modes may be defined on an ordinal scale, that is, for a passenger car, the average travel speed may be defined as fast, compared to that of a bus transit system, which may be considered as slow. However, on a cardinal scale the same

attribute may be explicitly specified as 60 and 35 miles per hour for the two modes, respectively.

The attributes' vector may be homogeneous, which means that all vectors have the same, or convertible, units. Alternatively, the vector may be heterogeneous, which implies the attributes are stated in completely different units.

Another notable distinction between attributes may stem from the fact that whether an attribute is a certain outcome of the alternative or it is uncertain, stated in terms of expectations or ranges of possible values.

*2.2.4. Decision Rule*

A decision rule describes the internal mechanism that the decision maker uses to process the information and arrive at a unique choice among all different alternatives.

Slovic et al. (1977) and Svenson (1979) give a comprehensive list of proposed decision rules. These rules can be categorized into five different groups; namely, dominance, satisfaction, lexicographic, elimination by aspects, utility. If an alternative is better for one attribute than another alternative and is not worse for the rest of the attributes, then the former is said to be dominant over the latter. The satisfaction rule states that each attribute should meet a minimum satisfactory level; otherwise, the alternative is not acceptable. Lexicographic rules take into consideration the importance of attributes, so it is assumed under this decision rule that a decision maker chooses the alternative that is the most attractive for the most important attribute. If there is a tie, then she looks at the next most important attribute, and so on. Elimination by aspects is, in effect, a combination of lexicographic and satisfaction rules. Under the utility rule, it

is assumed that different attributes are commensurable, in other words, the attractiveness of an alternative expressed by a vector of attributes is reducible to a scalar. Thus, the utility rule supports the notion of trade-offs among attributes, while the other four rules are non-compensatory. The utility rule suggests that a decision maker implicitly or explicitly is comparing different attributes of an alternative.

*2.2.5. Rational Behavior*

The key to the theory of choice lies in the beliefs of the decision maker as to what the outcome(s) of a certain choice should be. It is evident that different decision makers may not have a unique set of beliefs and their beliefs may be quite dissimilar. Therefore, rationality, which is generally defined as following the rules of common sense and logic, may not be very helpful in the analysis of individual choice behavior.

However, considering rationality in the sense of following a consistent and calculated decision making process is very important. In its broader definition, rationality contrasts with the impulsiveness arising from decision makers' state of mind and taste preferences at the time a decision is being made.

Simon (1957) has first suggested the concept of bounded rationality as a modification to the classical perfect rationality notion. Under the latter, a decision maker is capable of perfect retention and processing of large quantities of information to arrive at the most consistent decisions. Alternatively, bounded rationality aims to recognize the natural and realistic limitations of human beings as decision makers, in terms of data retention, analyses and implementation.

In order to avoid ambiguities in this regard, the general consensus about the rational behavior of a decision maker is limited to her manifestation of a consistent and transitive set of preferences. In this setting, consistency implies that the choice will be repeated under the identical situations, and transitivity suggests that once alternative one is preferred to alternative two, which in turn is preferred over alternative three, then alternative one is preferred to alternative three.

## 2.3 Random Utility Theory

The random utility concept as proposed by Manski (1977) suggests that inconsistencies in choice behavior should be attributed to the observational deficiencies of the analyst. It is always assumed that a decision maker selects the alternative with the highest utility. Utilities are not known to the analyst with certainty and therefore are treated by the analyst as random variables.

From this perspective, the choice probability of alternative $i$ is equal to the probability that the utility of alternative $i$ is greater than or equal to the utilities of all other alternatives in the choice set. This rule can be written in mathematical form, as:

$$P_n(i) = \Pr ob(U_{in} \geq U_{jn}, \forall j \in C_n) \qquad (2\text{-}1)$$

where,

$P_n(i)$ is the probability with which individual n chooses alternative i,

$C_n$ is the choice set belonging to individual n, and

$U_{in}$ is the utility of alternative i for individual n.

*2.3.1. Sources of Randomness*

As stated earlier, randomness in utilities may be the outcome of various shortcomings on the part of the analyst. Manski (1973) identifies four distinct sources of randomness in utility, namely:

1. unobserved attributes,
2. unobserved taste variations,
3. measurement errors and imperfect information, and
4. instrumental (or proxy) variables.

*2.3.2. Deterministic and Random Components of the Utility*

In order to make the random utility theory operational, three very basic steps may be taken. In fact, these steps are necessary for development of any choice model.

1. the separation of total utility into deterministic and random components,
2. the specification of the deterministic component, and
3. the specification of the random component.

Step one effectively is the core idea behind choice models, which are based on the random utility theory. It suggests that each utility may be divided into two additive parts, as follows:

$$U_{in} = V_{in} + \varepsilon_{in} \tag{2-2}$$

$$U_{jn} = V_{jn} + \varepsilon_{jn} \tag{2-3}$$

where,

$V_{in}, V_{jn}$ are the deterministic (systematic) components of the utilities for alternatives $i$ and $j$, respectively, for individual $n$, and

$\varepsilon_{in}, \varepsilon_{jn}$ are the random (disturbances) components of the utilities for alternatives $i$ and $j$, respectively, for individual $n$.

It should be noted that the deterministic part of the utility may assume any functional form (linear, logarithmic, exponential, etc.). Also, the attributes that are included in the specification of the deterministic part may be divided into two major groups. First, alternative attributes as expected and perceived by the decision maker. In the travel mode choice setting, they may include travel time, travel cost, comfort, convenience and safety of travel by a specific mode. Second, the group of attributes may be relevant to a decision maker's characteristics as they are also believed to play a role in the decision making process. For instance, attributes like income level, auto ownership, household size, age, occupation type and gender may be regarded as influential on many individuals' trip decisions. The distinction between modal and personal attributes may be expressed in the following mathematical form,

$$V_{in} = V(z_{in}, S_n) \tag{2-4}$$

where,

$z_{in}$ attributes of alternative $i$ for individual $n$, and

$S_n$ attributes of individual $n$.

The difficulty of specifying $V_{in}$ results from the definition of combinations of $z_{in}$ and $S_n$ that reflect a reasonable hypothesis about the effects of such variables on the utility and also the appropriate functional form for $V$.

In general, two criteria for selecting a functional from for $V$ exist. First, it should reflect any plausible theory about how various elements in $V$ influence utility. Second, it is desirable to use functions that have convenient computational properties making it easy to estimate their unknown parameters. Normally, a linear form in parameters is used to specify the function $V$; however, it does not necessarily mean that the function is also linear in attributes. Substituting relation 2-2 into equation 2-1 leads to,

$$P_n(i) = \Pr ob(V_{in} + \varepsilon_{in} \geq V_{jn} + \varepsilon_{jn}, \forall j \in C_n) \quad (2\text{-}5)$$

Or alternatively,

$$P_n(i) = \Pr ob(\varepsilon_{jn} - \varepsilon_{in} \leq V_{in} - V_{jn}, \forall j \in C_n) \quad (2\text{-}6)$$

The above formula suggests that the probability of individual $n$ choosing alternative $i$ is equal to the probability of differences between random terms in the utilities of alternative $i$ and any other alternative in the choice set being less than or equal to the differences in the deterministic terms of those utility pairs, respectively.

*2.3.3. IID and Different Discrete Choice Models*

Under the assumption of independent and identical distribution of error terms (iid) different choice models may be obtained. Assuming a uniform distribution for random terms leads to what is known as a linear probability model. This type of choice models suffers from the inevitable discontinuity present at the limits of the uniform distribution, which eventually produces choice probabilities equal to zero and one (completely deterministic) beyond those limits.

A normal distribution of random terms gives rise to probit models which have a much better theoretical and behavioral appeal. The Probit model has been widely used in diverse fields. One of the first transportation applications of Probit models analyzes the mode choice of commuters in Chicago (Lisco, 1967). Unfortunately, probit models do not have a closed form solution and therefore their calibration and estimation is difficult as the choice probability must be expressed as an integral.

This shortcoming of the Probit model has given rise to a search for models similar to probit, which have a closed form solution. Assumption of a Gumbel (Type I extreme value) distribution for the random term of the utility function has brought about a family of choice models, which are widely known as Logit models.

# CHAPTER 3

# METHODOLOGY

## 3.1 Logit Model

Reconsidering equation 2.2, it is more appropriate for the ease of presentation in this chapter to introduce a new notation for the vector of attributes of an alternative,

$$x_{in} = h(z_{in}, S_n) \tag{3-1}$$

where,

$h$ is a vector-valued function, and

$x_{in}$ is the vector of (transformed and/or combined) attributes of alternative $i$ for individual $n$.

Based on this notation and earlier discussion about the use of linear in parameter utility functions, the deterministic component of utility function may be written as,

$$V_{in} = \beta x_{in} = \beta_1 x_{in1} + \beta_2 x_{in2} + ... + \beta_K x_{inK} \tag{3-2}$$

where,

$\beta$ is the vector of $K$ unknown parameters $< \beta_1, \beta_2, ..., \beta_K >$, and

$x_{in1}, x_{in2}, ..., x_{inK}$ are the $K$ elements of the vector $x_{in}$.

As stated earlier, under the assumption of independent and identical Gumbel distribution of random components of utility functions the Logit model will be obtained,

$$P_n(i) = \frac{e^{\mu V_{in}}}{\sum_{j \in C_n} e^{\mu V_{jn}}} \tag{3-3}$$

or,

$$P_n(i) = \frac{e^{\mu \beta x_{in}}}{\sum_{j \in C_n} e^{\mu \beta x_{jn}}} = \frac{e^{\beta' x_{in}}}{\sum_{j \in C_n} e^{\beta' x_{jn}}} \tag{3-4}$$

In the case that the choice set is only comprised of two alternatives, the Logit model is called a binomial Logit model, which is a special case of the more general multinomial Logit model.

*3.1.1. Properties of Logit*

3.1.1.1 Independence from Irrelevant Alternatives

The IIA property holds that for a specific individual, the ratio of the choice probabilities of any two alternatives is not influenced by the systematic utilities of any other alternative(s). In other words, under IIA, the ratio of choice probabilities of two alternatives should solely be determined by their own attributes. In the case of a Logit model it can be shown, using (3-3), that IIA holds,

$$\frac{P_n(i)}{P_n(j)} = \frac{e^{V_{in}}}{e^{V_{jn}}} = e^{V_{in} - V_{jn}} \tag{3-5}$$

Although it seems to be a very trivial observation, the IIA property may have profound implications on the accuracy of the Logit model. Especially, if the alternatives are not really mutually independent form one another. Technically speaking, the IIA property is a direct product of the assumption of independent error terms in the utilities

of all alternatives. Therefore, any choice model built on the assumption of iid in the random components of utility functions will produce unrealistic predictions once the condition is not strongly satisfied. A famous case of the violation of iid is known as the red bus/ blue bus paradox.

3.1.1.2 Elasticities of Logit

Manheim (1979) has provided a thorough discussion of the concept of elasticity. Elasticity of the choice probability of an alternative with respect to any attribute of the same or another alternative is defined as the percent change in that probability as a result of a one percent change in the attribute in question. The knowledge of elasticities for a choice probability will serve in the sensitivity analysis of the choice model under investigation. So, elasticities can be very useful in gaining insight into the importance and impact of different attributes on the choice probabilities.

The following gives a point elasticity of the probability of person $n$ choosing alternative $i$ with respect to an attribute of the same alternative (direct elasticity),

$$E_{x_{ink}}^{P_n(i)} = \frac{\partial P_n(i)}{\partial x_{ink}} \cdot \frac{x_{ink}}{P_n(i)} = \frac{\partial \ln P_n(i)}{\partial \ln x_{ink}} \tag{3-6}$$

in the case of the Logit model, it can be shown that (3-6) will result in,

$$E_{x_{ink}}^{P_n(i)} = [1 - P_n(i)] x_{ink} \beta_k = [1 - P_n(i)] \beta_k \cdot \frac{\partial h^k}{\partial z_{ink}} . z_{ink} \tag{3-7}$$

where,

$$x_{ink} = h^k(z_{ink}) \tag{3-8}$$

However, for the Logit model, the point cross elasticity of a disaggregate choice probability can be obtained using the following equation,

$$E_{x_{jnk}}^{P_n(i)} = -P_n(j).\beta_k.\frac{\partial h^k}{\partial z_{jnk}}.z_{jnk}, \quad for\ j \neq i \tag{3-9}$$

*3.1.2. Estimation of Multinomial Logit*

The estimation problem refers to the problem of econometrically finding the utility parameters $\beta_1, \beta_2, \ldots, \beta_K$ from a sample of observations. For tackling this problem, the most widely used estimator is the maximum likelihood estimator (MLE). McFadden (1974) has led the work in the area of applying MLE to the estimation of multinomial Logit models. The likelihood function is defined as,

$$L^*(\beta) = \prod_{n=1}^{N} \prod_{i \in C_n} P_n(i)^{y_{in}} \tag{3-10}$$

where,

$$y_{in} = \begin{cases} 1 & , if\ \ person\ n\ chose\ alternative\ i \\ 0 & , otherwise \end{cases} \tag{3-11}$$

So, it can be trivially shown that,

$$\sum_{i \in C_n} y_{in} = 1 \quad , \quad y_{in} = 1\ or\ 0 \quad \forall i \in C_n \tag{3-12}$$

The likelihood function (3-10) represents a non-linear objective function for the estimation problem under consideration. Thus, it is generally accepted that the log-likelihood function should be utilized, instead, which facilitates the estimation process.

$$L(\beta) = \sum_{n=1}^{N} \sum_{i \in C_n} y_{in} \ln P_n(i) \qquad (3\text{-}13)$$

where, for a linear in parameter Logit model, equation (3-13) will be transformed into,

$$L(\beta) = \sum_{n=1}^{N} \sum_{i \in C_n} y_{in} (\beta x_{in} - \ln \sum_{j \in C_n} e^{\beta x_{jn}}) \qquad (3\text{-}14)$$

The MLE is shown to be a consistent, asymptotically normal and also asymptotically efficient estimator (Ben-Akiva and Lerman, 1985).

## 3.2 Nested Logit Model

In each choice set, there is a practical chance that a number of alternatives are less than independent. This is especially true, when a group of alternatives are by nature more similar than others. For instance, in a mode choice problem, it may be argued that public transit modes have more in common versus the private car. The previous section discusses that this will be the root cause of iid violation in MNL, which will lead to the unrealistic modeling of choice behavior.

The probit model has a unique capacity to address this problem using its general and arbitrary covariance matrix specification; however, due to computational problems it is not widely used. Alternatively, a structural variation in the MNL may, at least to some extent, resolve this concern. Putting similar alternatives inside a nest and then representing the nest by a composite alternative, which competes at a different level with the rest of individual's alternatives, can improve the results. This variation of MNL is called the nested Logit model.

Figure 3-1 shows a possible organization of the above mode choice example. The following nested structure may be adopted to form a public transit nest containing two modes, namely bus and metro. Then, the combined nest competes with the private car mode.

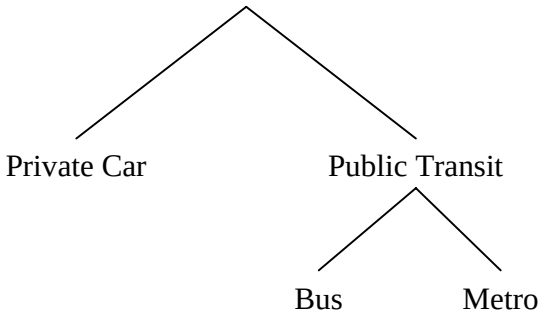



Figure 3-1 Structure of a Nested Model

Based on this nested structure, the probability of choosing the bus mode may be computed using the following formula,

$$P(Bus) = P(Transit) \times P(Bus \mid Transit) \tag{3-15}$$

Estimation of the nested Logit model is essentially the same as MNL estimation. However, in the nested case, each nest should be represented at the higher level by a composite utility of the form,

$$V_I = \varphi EMU + \alpha Z \tag{3-16}$$

where,

$V_I$ is the composite utility of the nest,

$Z$ is the set of common attributes between the alternatives in the nest,

$\varphi, \alpha$ are parameters to be estimated, and

$EMU$ is the expected maximum utility of the lower nest alternatives, computed as,

$$EMU = \ln \sum_{j \in I} e^{V_j} \tag{3-17}$$

The value of parameter $\varphi$ gives an indication of how appropriate the specification of a nested Logit model structure is. The value of $\varphi$ is always larger than zero and smaller than or equal to one. The closer this value is to zero, indicates more improvement gained by the nested structure and use of the nested Logit model. On the contrary, the closer the value of $\varphi$ is to one, implies that the ordinary MNL specification is not very different from the nested Logit model.

## 3.3 Goodness of Fit Measures

The significance of utility parameters estimated using MLE can be evaluated through an ordinary t-test. However, this is a test for single parameters and will help make decisions regarding the inclusion or rejection of a variable into the model.

In the absence of any information, it may be assumed that each alternative will be chosen with equal probability by each individual. Therefore, plugging in equal choice probabilities in equation 3-13, the log-likelihood of a naïve model without the use of any information may be obtained as follows,

$$L(0) = N \ln(\frac{1}{C}) \tag{3-18}$$

where, $C$ is the number of alternatives in the choice set.

The improvement in modeling brought about by using the available information regarding the choice behavior can be measured through following,

$$\rho^2 = 1 - \frac{L(\beta)}{L(0)} \tag{3-19}$$

this statistic will vary between 0 and 1. The closer $L(\beta)$ is to zero, the better the fitted model is and as a result $\rho^2$ will tend toward one. However, if the improvement in the model over the naïve model, gained through use of information, is not significant then $L(\beta)$ will not be very different from $L(0)$ and $\rho^2$ will tend towards zero.

The above measure is dependent on the mixture of observed choices present in the data. In other words, the market share of various alternatives may be very different from the balanced case assumed in the naïve model. Therefore, compared to the calculated $L(0)$, any model may suggest a large improvement over the naïve case and thus result in a convincingly large $\rho^2$ statistic. To account for this effect, the following corrected measure of goodness of overall fit is introduced:

$$\rho_c^2 = 1 - \frac{L(\beta)}{L(C)} \tag{3-20}$$

in which $L(C)$ is defined as,

$$L(C) = \sum_{i \in C_n} N_i \ln(\frac{N_i}{N}) \tag{3-21}$$

where,

$L(C)$, log-likelihood of the model assuming the probability of choice for each alternative by all individuals is equal to the alternative's market share, and

$N_i$, number of individuals choosing alternative $i$

# CHAPTER 4

# APPLICATION AND RESULTS

This chapter presents the stated preference part of the survey designed to draw responses from Dallas and Houston areas with regard to the choice of travel mode and lane when presented with a set of travel times and tolls. A summary of the implementation phase and the consequent outreach effort along with the results of the preliminary statistical analyses on the data is provided. The efforts to prepare the final dataset for model estimation are described. They include cleaning, coding, aggregation and weighting of the initial data. Finally, the estimated Logit and nested-Logit models for the DFW and Houston areas are presented and compared. The effects of trip purpose and previous experience in using toll roads on the stated preference for managed lanes are also investigated.

## 4.1 Survey Design and Data Collection

### *4.1.1 Survey Design*

Throughout the development of the survey the objective was to examine a traveler's potential use of MLs. To do that, the questions had to be related to the perceived benefits of MLs and other reasons to support implementation. Table 2-1 presented the measures of effectiveness (MOE) for assessing the success of MLs in meeting their goals. Survey development focused on acquiring public feedback on their perception of the resulting MOEs. As stated earlier, central to all of these evaluations

was the prediction of how the travelers would alter their behavior given specific GPL and ML options. Therefore, the survey was designed with this goal in mind.

In the absence of any real operational MLs in Houston or Dallas, the challenge of measuring the travelers' mode and route choice behavior given specific GPL and ML options, was addressed using both revealed preference and stated preference questions. Each respondent was presented with up to four stated preference questions. The questions were designed to measure the respondents' value of time by adjusting both the travel time and toll based on their answers to the previous questions. However, the specifics of each question were tied back to the revealed typical travel distance of the respondents on their most frequently traveled freeways.

Since the survey data includes responses from two metropolitan regions, the study can assess if the reaction to MLs differs for these two different areas. The hypothesis that current toll users are more likely to use MLs is the incentive for the inclusion of a specific question in the survey; namely whether the respondent pays a toll while traveling, and if they frequently travel on a toll road. Another question regarding the general support or opposition to the concept of Managed Lane is included. This information combined with socio-economic data and critical characteristics of the current typical respondent trips helps to increase understanding and to explain the respondents' stated preferences.

*4.1.2 Data Collection*

The major part of the data collection effort was conducted online using two websites (www.houstontravelsurvey.org and www.dallastravelsurvey.org). The survey

was made available in both English and Spanish. Out of 4634 filled out surveys only 39 were completed in Spanish. The online administration of the survey vastly facilitated customizing questions so that only relevant questions were asked from each respondent. For example, if the respondent indicated they never rode transit, then the only transit related question they received was one asking why they chose not to ride transit. The web survey would also remind each respondent of the values they had indicated earlier so that the chances of confusion regarding questions would be minimized. The biggest advantage was in the stated preference questions as the toll rate and VTTS could vary dynamically based on the options selected in the previous question.

The survey benefited in both content and exposure due to the help of several agencies which operate road facilities and have carried out work in this field. Meetings were carried out with the following agencies:

- Harris County Toll Road Authority (HCTRA),
- Houston-Galveston Area Council (HGAC),
- The Metropolitan Transit Authority of Harris County (METRO),
- Transportation Management Organization in Greenway Plaza and Uptown Houston (TREK),
- Texas Department of Transportation (TxDOT) Houston District,
- North Texas Tollway Authority (NTTA), and
- North Central Texas Council of Governments (NCTCOG).

Cooperation with the above agencies significantly helped to improve the survey by incorporating their views as well as help advertise the survey. The links to the pages

were put on various government body websites, such as NTTA (http://www.ntta.org), Dallas Area Rapid Transit (http://www.dartnet.org, https://www.dart.org), NCTCOG (http://nctcog.org), and City of Dallas (http://www.dallascityhall.com). Small "push cards" (three inch by five inch card stock pieces of paper) with an introduction to the survey and the survey link were distributed in both Houston and Dallas.

About 3600 respondents completed the survey online by the end of May. Initial analysis of the survey respondents indicated that the share of low-income respondents was not proportional to their share in the overall population. This was likely due to the unavailability of the internet to low income households. As a result and in order to correct for this effect, it was decided to target low income households. A few Department of Public Safety (DPS) offices in Houston and Dallas were selected for this purpose. The respondents were not all familiar with the use of laptops, which introduced the need for a combination of paper and laptop surveys.

As a result of this outreach program, which was conducted over a period of two weeks in late July, almost 1000 new entries were added to the dataset, which increased the total number to 4634, with 2,026 from DFW and 2,562 from Houston while 46 survey takers did not reveal their locations. Toycen et al. (2006) provided a complete description of the survey method and data collection process, as well as the outreach efforts involved in data collection.

Table 4-1 Variables and Their Characteristics and Definitions

| ***Category*** | ***Variable Name*** | ***Variable Description*** | ***Variable Type*** | ***Variable Definition and Values*** |
|---|---|---|---|---|
| *Constant* | **unit** | Constant utility term | Continuous | |
| *Alternative Characteristics* | **ttime** | Travel time on managed lane (minutes) | Continuous | |
| | **toll** | Amount of toll paid for using managed lane (dollars) | Continuous | |
| *Decision Maker Characteristics* | **age** | Respondent's age category | Discrete | (1) 16-24<br>(2) 25-34<br>(3) 35-44<br>(4) 45-54<br>(5) 55-64<br>(6) over 65 |
| | **gender** | Respondent's gender | Binary | (0) Female<br>(1) Male |
| | **ethnicity** | Respondent's ethnicity | Discrete | (1) Caucasian<br>(2) African-American<br>(3) Hispanic<br>(456) Asian/Native American and Others |
| | **hhtype** | Respondent's household type | Discrete | (1) Single<br>(2) Single Parent<br>(3) Unrelated<br>(4) Married with Children<br>(5) Married w/o Children<br>(6) Others |
| | **hhnum** | Respondent's household size | Discrete | 0 to 7 |
| | **motorveh** | Respondent's household motorized vehicle ownership | Discrete | 0 to 7 |
| | **occtype** | Respondent's occupation type | Discrete | (1) Professional<br>(2) Technical<br>(3) Sales<br>(4) Administrative<br>(5) Service<br>(6) Manufacturing<br>(7) Stay home<br>(8) Student<br>(9) Self employed<br>(10) Unemployed<br>(11) Retired<br>(12) Others |
| | **educ** | Respondent's education level | Discrete | (1) Less than high school<br>(2) High school graduate<br>(3) Vocational<br>(4) College graduate<br>(5) Post graduate |

Table 4-1 continued

| *Category* | *Variable Name* | *Variable Description* | *Variable Type* | *Variable Definition and Values* |
|---|---|---|---|---|
| *Decision Maker Characteristics* | **income** | Respondent's household annual income level | Discrete | (123) Less than 25,000<br>(4) 25,000-35,000<br>(5) 35,000-50,000<br>(6) 50,000-75,000<br>(7) 75,000-100,000<br>(8) 100,000-150,000<br>(910) More than 150,000 |
| | **purpose** | Respondent's typical trip purpose | Discrete | (1) Commute<br>(2) Work<br>(3) School<br>(4) Other |
| | **tollpay** | Indicating whether the respondent pays toll on her typical trips | Binary | (0) No<br>(1) Yes |
| | **mlbin** | Respondent's interest in using managed lane | Binary | (0) Not interested<br>(1) Interested |
| | **fampool** | Indicating whether the respondent typically carpools with her family members | Binary | (0) No<br>(1) Yes |
| *Interactions* | **pcomtoll** | See 4.2.3 | Continuous | |
| | **pwrktoll** | See 4.2.3 | Continuous | |
| | **pschtoll** | See 4.2.3 | Continuous | |
| | **prectoll** | See 4.2.3 | Continuous | |
| | **tollptoll** | See 4.2.3 | Continuous | |
| *Choice* | **decision** | Respondent's stated preference | Discrete | (1) SOV on ML<br>(2) SOV on GPL<br>(3) HOV2 on ML<br>(4) HOV2 on GPL<br>(5) HOV3+ on ML<br>(6) HOV3+ on GPL |

The set of variables utilized in the model estimation are presented in Table 4-1. These variables may be grouped into two major categories, that is, alternative characteristics, and decision maker characteristics, which include two and thirteen main variables, respectively. The choices are limited to the combinations of three different modes, SOV, HOV2, HOV3+ and two lane types, that is ML and GPLs. Therefore, in total, six choice combinations are considered.

## 4.2 Dataset Preparation

Before proceeding any further, it is necessary to prepare the dataset for any future analysis. The preparation process consists of cleaning and re-coding, data aggregations (if necessary), definition and inclusion of new dummy and interaction variables, and dataset weighting.

### *4.2.1 Dataset Cleaning and Re-Coding*

The cleaning stage consisted of looking for typos, out of range values and consideration of comments to fill out the unanswered or partially filled fields in the most reasonable way. Also, consistency checks were conducted to make sure that each individual's responses to related questions generally agreed with each other. In the case of inconsistencies, the fields with higher levels of entry effort would override. All in all, the objective in data cleaning was to make the best use of all available data and prevent data from being discarded. Only a few records, which were obviously entered due to repeated clicks on the submit button, were removed.

In order to prepare the dataset for use in the statistical analysis, some entries had to be recoded. This was especially true for string type discrete entries, such as gender, which had to be transformed into number codes, 0 or 1, to represent the same data for use in the estimation process.

### *4.2.2 Dataset Aggregation*

Based on the original categorical definition of some key trip and socio-demographic variables, the data collection effort did not seem to have gathered enough observations in all initial categories to provide for a statistically sound analysis. As a

result, it was decided to aggregate some categories as a means to increase the number of observations and reduce the bias in the final models.

Specifically, the recreational trip purpose was aggregated with the other trip purpose category. Also, the Asian and Native American ethnic groups were aggregated into the group of other ethnicities. For the case of the income variable, the three initial lowest income categories were combined to form the less than $25,000 annual income level, while the two highest income categories were also aggregated to form the $150,000 and more annual income group; thus, reducing the number of income levels from ten to seven.

*4.2.3 Variable Definitions*

In another effort to prepare the data for model estimation, the categorical variables were expanded into a set of dummy variables with binary values of either 0 or 1, each designated to one of the categories belonging to the original variable. It should be noted that each variable with $N$ categories could be represented by $N-1$ binary dummy variables. Letting all the dummies associated with the variable in question equal to zero would specify the final category without any representative dummy variable. In this study, the last category was selected to be the category without a corresponding dummy variable. In total, 37 new dummy variables were created using this process, which when combined with the initially existing four binary variables resulted in 41 binary variables in the final dataset.

In addition, five interaction variables were introduced to the dataset. The interactions of trip purposes and the toll amount and also previous experience with toll road use and the charged toll were represented using the following variables.

$$pcomtoll = \begin{cases} toll & , if\ \ trip\ \ purpose\ is\ to\ commute \\ 0 & , otherwise \end{cases}$$

$$pwrktoll = \begin{cases} toll & , if\ \ trip\ \ purpose\ is\ to\ do\ the\ work\ related\ \ jobs \\ 0 & , otherwise \end{cases}$$

$$pschtoll = \begin{cases} toll & , if\ \ trip\ \ purpose\ is\ to\ go\ to\ school \\ 0 & , otherwise \end{cases}$$

$$prectoll = \begin{cases} toll & , if\ \ trip\ \ purpose\ is\ recreation\ or\ other \\ 0 & , otherwise \end{cases}$$

$$tollptoll = \begin{cases} toll & , if\ \ respondent\ already\ \ pays\ toll\ \ for\ her\ trips \\ 0 & , otherwise \end{cases}$$

It is important to note that the following convention was adopted to name the variables. The convention helps identify the dummies associated with each base categorical variable as well as if the variable is generic or an alternative specific one.

$[Var\_Name]\_[d\#]\_[Alternative\ \#]$

where,

$[Var\_Name]$ is the variable name

$[d\#]$ indicates a dummy variable associated with category number # in the base variable, and

$[Alternative\ \#]$ indicates an alternative specific variable associated with alternative number #.

As an example, the variable $age\_d3\_2$ is the alternative specific dummy variable for the 35-44 age group specific to the second choice, that is SOV traveling on GPL.

*4.2.4 Dataset Weighting*

The relative distribution of annual household income versus ethnic groups, in the most recent census for DFW and Houston areas are shown in Tables 4-2 and 4-3, respectively.

Table 4-2 Census Distribution of Household Annual Incomes vs. Ethnicities in Dallas Fort-Worth Metroplex

| Household Annual Income | Caucasians | African American | Hispanic | Asian / Native American and others | Total: |
|---|---|---|---|---|---|
| Less than $24,999 | 12.03% | 5.14% | 4.69% | 1.17% | 23.03% |
| $25,000 to $34,999 | 7.46% | 2.11% | 2.56% | 0.53% | 12.66% |
| $35,000 to $49,999 | 10.71% | 2.25% | 2.90% | 0.80% | 16.66% |
| $50,000 to $74,999 | 14.27% | 2.32% | 2.68% | 1.12% | 20.38% |
| $75,000 to $99,999 | 9.05% | 0.99% | 1.07% | 0.60% | 11.70% |
| $100,000 to $149,999 | 7.87% | 0.61% | 0.61% | 0.53% | 9.62% |
| $150,000 or more | 5.27% | 0.24% | 0.26% | 0.27% | 6.04% |
| Total: | 66.66% | 13.65% | 14.77% | 5.01% | 100.08% |

Table 4-3 Census Distribution of Household Annual Incomes vs. Ethnicities in Houston Area

| Household Annual Income | Caucasians | African American | Hispanic | Asian / Native American and others | Total: |
|---|---|---|---|---|---|
| Less than $24,999 | 10.27% | 6.87% | 7.70% | 1.52% | 26.36% |
| $25,000 to $34,999 | 5.80% | 2.47% | 3.69% | 0.69% | 12.66% |
| $35,000 to $49,999 | 8.34% | 2.71% | 4.00% | 0.89% | 15.94% |
| $50,000 to $74,999 | 11.58% | 2.64% | 3.55% | 1.16% | 18.93% |
| $75,000 to $99,999 | 7.73% | 1.25% | 1.38% | 0.71% | 11.07% |
| $100,000 to $149,999 | 7.26% | 0.74% | 0.80% | 0.66% | 9.46% |
| $150,000 or more | 4.69% | 0.27% | 0.40% | 0.33% | 5.69% |
| Total: | 55.68% | 16.96% | 21.52% | 5.96% | 100.12% |

The totals are not exactly equal to 100%, which is due to round off errors caused by division calculations. The corresponding distributions, for the same regions, in the collected sample are presented in Tables 4-4 and 4-5, respectively.

Comparisons with the census data suggest that the distribution of socio-demographic characteristics, such as income and race, in the sample dataset is not representative of the target population out of which the sample is taken. Therefore, before any model estimation, the dataset is weighted to account for the observed discrepancies between population and the sample, in terms of the income and ethnicity distributions.

Two alternatives are available to accomplish this goal. Conceptually, the first method tries to reduce the number of observations in cells, by leaving out randomly chosen observations, so that finally the remaining ones will match the relative distribution in the census. It is obvious that in this method some data will never be used in the estimation process. However, the cut out data may be used in the validation of the models.

The second method, on the contrary, will increase the sample cell sizes by adding randomly chosen observations to each cell, from among the ones originally in the same cell, to make the relative cell sizes compatible with the census data. Although this method seemingly makes complete use of all collected data by preventing observations from being kept out of estimation process, the method will result in the need for putting aside some data for validation purposes from the very beginning. It means that both methods eventually leave out some data in the estimation procedure;

however, due to the control on the number of validation observations gained by using the second method it is adopted for application to weight the sample in this study.

Table 4-4 Sample Distribution of Household Annual Incomes vs. Ethnicities in Dallas Fort-Worth Metroplex

| Household Annual Income | Caucasians | African American | Hispanic | Asian / Native American and others | Total: |
|---|---|---|---|---|---|
| Less than $24,999 | 150<br>2.22% | 81<br>1.20% | 43<br>0.64% | 61<br>0.90% | 335<br>4.97% |
| $25,000 to $34,999 | 149<br>2.21% | 75<br>1.11% | 46<br>0.68% | 12<br>0.18% | 282<br>4.18% |
| $35,000 to $49,999 | 522<br>7.74% | 67<br>0.99% | 77<br>1.14% | 48<br>0.71% | 714<br>10.59% |
| $50,000 to $74,999 | 967<br>14.34% | 82<br>1.22% | 59<br>0.88% | 57<br>0.85% | 1165<br>17.28% |
| $75,000 to $99,999 | 1119<br>16.60% | 75<br>1.11% | 44<br>0.65% | 96<br>1.42% | 1334<br>19.79% |
| $100,000 to $149,999 | 1527<br>22.65% | 56<br>0.83% | 84<br>1.25% | 124<br>1.84% | 1791<br>26.56% |
| $150,000 or more | 971<br>14.40% | 34<br>0.50% | 32<br>0.47% | 84<br>1.25% | 1121<br>16.63% |
| Total: | 5405<br>80.17% | 470<br>6.97% | 385<br>5.71% | 482<br>7.15% | 6742<br>100.00% |

A 80-20 random division of the sample dataset is considered to form the estimation and validation observations, respectively. Tables 4-6 and 4-7 show the number of original observations used in the estimation and validation of the models for the DFW area. While the corresponding data for the Houston observations are given in Tables 4-8 and 4-9. Also the total number of observations used in the estimation and validation of overall models for both DFW and Houston are presented in Tables 4-10 and 4-11, respectively.

The outcome of the application of the described weighting method on the estimation observations of DFW, Houston and combined dataset, in terms of the

number of observations in each cell, is given in Tables 4-12, 4-13, and 4-14. This implies that on average DFW, Houston and combined estimation data are repeated 4.6, 2.7, and 3.5 times, respectively to form the associated weighted datasets. These values are in the tolerable range and do not pose any problem in using the weighted datasets.

Table 4-5 Sample Distribution of Household Annual Incomes vs. Ethnicities in Houston Area

| Household Annual Income | Caucasians | African American | Hispanic | Asian / Native American and others | Total: |
|---|---|---|---|---|---|
| Less than $24,999 | 212<br>2.36% | 119<br>1.33% | 179<br>2.00% | 39<br>0.43% | 549<br>6.12% |
| $25,000 to $34,999 | 260<br>2.90% | 65<br>0.72% | 119<br>1.33% | 27<br>0.30% | 471<br>5.25% |
| $35,000 to $49,999 | 656<br>7.31% | 116<br>1.29% | 143<br>1.59% | 44<br>0.49% | 959<br>10.69% |
| $50,000 to $74,999 | 1301<br>14.51% | 128<br>1.43% | 172<br>1.92% | 143<br>1.59% | 1744<br>19.44% |
| $75,000 to $99,999 | 1370<br>15.27% | 97<br>1.08% | 128<br>1.43% | 132<br>1.47% | 1727<br>19.26% |
| $100,000 to $149,999 | 1795<br>20.01% | 104<br>1.16% | 133<br>1.48% | 164<br>1.83% | 2196<br>24.48% |
| $150,000 or more | 1153<br>12.86% | 36<br>0.40% | 62<br>0.69% | 72<br>0.80% | 1323<br>14.75% |
| Total: | 6747<br>75.23% | 665<br>7.41% | 936<br>10.44% | 621<br>6.92% | 8969<br>100.00% |

Table 4-6 Un-weighted Sample Distribution of Household Annual Income vs. Ethnicity in Dallas Fort-Worth Metroplex Used for Estimation

| Household Annual Income | Caucasians | African American | Hispanic | Asian / Native American and others | Total: |
|---|---|---|---|---|---|
| Less than $24,999 | 124 | 64 | 34 | 51 | 273 |
| $25,000 to $34,999 | 125 | 54 | 37 | 10 | 226 |
| $35,000 to $49,999 | 417 | 47 | 60 | 37 | 561 |
| $50,000 to $74,999 | 775 | 68 | 48 | 43 | 934 |
| $75,000 to $99,999 | 888 | 61 | 33 | 79 | 1061 |
| $100,000 to $149,999 | 1222 | 43 | 70 | 94 | 1429 |
| $150,000 or more | 789 | 28 | 29 | 66 | 912 |
| Total: | 4340 | 365 | 311 | 380 | 5396 |

Table 4-7 Sample Distribution of Household Annual Income vs. Ethnicity in Dallas Fort-Worth Metroplex Used for Validation

| Household Annual Income | Caucasians | African American | Hispanic | Asian / Native American and others | Total: |
|---|---|---|---|---|---|
| Less than $24,999 | 26 | 17 | 9 | 10 | 62 |
| $25,000 to $34,999 | 24 | 21 | 9 | 2 | 56 |
| $35,000 to $49,999 | 105 | 20 | 17 | 11 | 153 |
| $50,000 to $74,999 | 192 | 14 | 11 | 14 | 231 |
| $75,000 to $99,999 | 231 | 14 | 11 | 17 | 273 |
| $100,000 to $149,999 | 305 | 13 | 14 | 30 | 362 |
| $150,000 or more | 182 | 6 | 3 | 18 | 209 |
| Total: | 1065 | 105 | 74 | 102 | 1346 |

Table 4-8 Un-weighted Sample Distribution of Household Annual Income vs. Ethnicity in Houston Area Used for Estimation

| Household Annual Income | Caucasians | African American | Hispanic | Asian / Native American and others | Total: |
|---|---|---|---|---|---|
| Less than $24,999 | 173 | 95 | 145 | 32 | 445 |
| $25,000 to $34,999 | 206 | 50 | 100 | 23 | 379 |
| $35,000 to $49,999 | 542 | 94 | 110 | 33 | 779 |
| $50,000 to $74,999 | 1046 | 107 | 149 | 119 | 1421 |
| $75,000 to $99,999 | 1087 | 79 | 101 | 108 | 1375 |
| $100,000 to $149,999 | 1408 | 77 | 111 | 125 | 1721 |
| $150,000 or more | 919 | 32 | 53 | 60 | 1064 |
| Total: | 5381 | 534 | 769 | 500 | 7184 |

Table 4-9 Sample Distribution of Household Annual Income vs. Ethnicity in Houston Area Used for Validation

| Household Annual Income | Caucasians | African American | Hispanic | Asian / Native American and others | Total: |
|---|---|---|---|---|---|
| Less than $24,999 | 39 | 24 | 34 | 7 | 104 |
| $25,000 to $34,999 | 54 | 15 | 19 | 4 | 92 |
| $35,000 to $49,999 | 114 | 22 | 33 | 11 | 180 |
| $50,000 to $74,999 | 255 | 21 | 23 | 24 | 323 |
| $75,000 to $99,999 | 283 | 18 | 27 | 24 | 352 |
| $100,000 to $149,999 | 387 | 27 | 22 | 39 | 475 |
| $150,000 or more | 234 | 4 | 9 | 12 | 259 |
| Total: | 1366 | 131 | 167 | 121 | 1785 |

Table 4-10 Un-weighted Sample Distribution of Household Annual Income vs. Ethnicity in Both Dallas Fort-Worth Metroplex and Houston Area Used for Estimation

| Household Annual Income | Caucasians | African American | Hispanic | Asian / Native American and others | Total: |
|---|---|---|---|---|---|
| Less than $24,999 | 297 | 159 | 179 | 83 | 718 |
| $25,000 to $34,999 | 331 | 104 | 137 | 33 | 605 |
| $35,000 to $49,999 | 959 | 141 | 170 | 70 | 1340 |
| $50,000 to $74,999 | 1821 | 175 | 197 | 162 | 2355 |
| $75,000 to $99,999 | 1975 | 140 | 134 | 187 | 2436 |
| $100,000 to $149,999 | 2630 | 120 | 181 | 219 | 3150 |
| $150,000 or more | 1708 | 60 | 82 | 126 | 1976 |
| Total: | 9721 | 899 | 1080 | 880 | 12580 |

Table 4-11 Sample Distribution of Household Annual Income vs. Ethnicity in Both Dallas Fort-Worth Metroplex and Houston Area Used for Validation

| Household Annual Income | Caucasians | African American | Hispanic | Asian / Native American and others | Total: |
|---|---|---|---|---|---|
| Less than $24,999 | 65 | 41 | 43 | 17 | 166 |
| $25,000 to $34,999 | 78 | 36 | 28 | 6 | 148 |
| $35,000 to $49,999 | 219 | 42 | 50 | 22 | 333 |
| $50,000 to $74,999 | 447 | 35 | 34 | 38 | 554 |
| $75,000 to $99,999 | 514 | 32 | 38 | 41 | 625 |
| $100,000 to $149,999 | 692 | 40 | 36 | 69 | 837 |
| $150,000 or more | 416 | 10 | 12 | 30 | 468 |
| Total: | 2431 | 236 | 241 | 223 | 3131 |

Table 4-12 Weighted Sample Distribution of Household Annual Income vs. Ethnicity in Dallas Fort-Worth Metroplex Used for Estimation

| Household Annual Income | Caucasians | African American | Hispanic | Asian / Native American and others | Total: |
|---|---|---|---|---|---|
| Less than $24,999 | 2993 | 1277 | 1166 | 291 | 5726 |
| $25,000 to $34,999 | 1855 | 524 | 636 | 131 | 3147 |
| $35,000 to $49,999 | 2663 | 559 | 722 | 200 | 4143 |
| $50,000 to $74,999 | 3547 | 576 | 665 | 278 | 5067 |
| $75,000 to $99,999 | 2249 | 247 | 266 | 148 | 2910 |
| $100,000 to $149,999 | 1956 | 152 | 152 | 132 | 2392 |
| $150,000 or more | 1311 | 59 | 65 | 66 | 1502 |
| Total: | 16575 | 3394 | 3672 | 1246 | 24888 |

Table 4-13 Weighted Sample Distribution of Household Annual Income vs. Ethnicity in Houston Area Used for Estimation

| Household Annual Income | Caucasians | African American | Hispanic | Asian / Native American and others | Total: |
|---|---|---|---|---|---|
| Less than $24,999 | 2013 | 1347 | 1508 | 298 | 5165 |
| $25,000 to $34,999 | 1137 | 484 | 723 | 136 | 2481 |
| $35,000 to $49,999 | 1633 | 531 | 783 | 175 | 3123 |
| $50,000 to $74,999 | 2269 | 517 | 696 | 227 | 3709 |
| $75,000 to $99,999 | 1515 | 245 | 271 | 139 | 2169 |
| $100,000 to $149,999 | 1423 | 146 | 157 | 129 | 1854 |
| $150,000 or more | 919 | 54 | 78 | 65 | 1115 |
| Total: | 10910 | 3323 | 4216 | 1167 | 19615 |

Table 4-14 Weighted Sample Distribution of Household Annual Income vs. Ethnicity in Both Dallas Fort-Worth Metroplex and Houston Area Used for Estimation

| Household Annual Income | Caucasians | African American | Hispanic | Asian / Native American and others | Total: |
|---|---|---|---|---|---|
| Less than $24,999 | 5005 | 2624 | 2674 | 588 | 10891 |
| $25,000 to $34,999 | 2993 | 1008 | 1360 | 267 | 5628 |
| $35,000 to $49,999 | 4296 | 1090 | 1505 | 375 | 7266 |
| $50,000 to $74,999 | 5817 | 1093 | 1361 | 505 | 8776 |
| $75,000 to $99,999 | 3765 | 491 | 537 | 287 | 5079 |
| $100,000 to $149,999 | 3379 | 298 | 309 | 261 | 4246 |
| $150,000 or more | 2230 | 113 | 143 | 131 | 2617 |
| Total: | 27485 | 6718 | 7888 | 2413 | 44504 |

## 4.3 Model Estimation

The model estimation is performed using the SAS software version 9.1.3. It provides an integrated environment for database management and analysis. The features of SAS makes it a very powerful tool in preparing extra large datasets for a variety of statistical analyses, including calibration and validation of a large number of multinomial and nested Logit models with different utility functions and structures. To this end, the MDC procedure is mainly used (SAS, 2006).

The same procedure can also be used to estimate more sophisticated models of the Logit family and is expected to be updated regularly in order to further expand its available features and options. The code written in the SAS environment for the preparation of the dataset and eventual model estimation and validation may be found in Appendix C.

Based on the nature of the choice alternatives in this study, the nested Logit models were estimated using two decision structures as shown in Figure 4-1. The structures suggest that mode and route choices are generally being made at two different levels and are being considered separately by individuals.

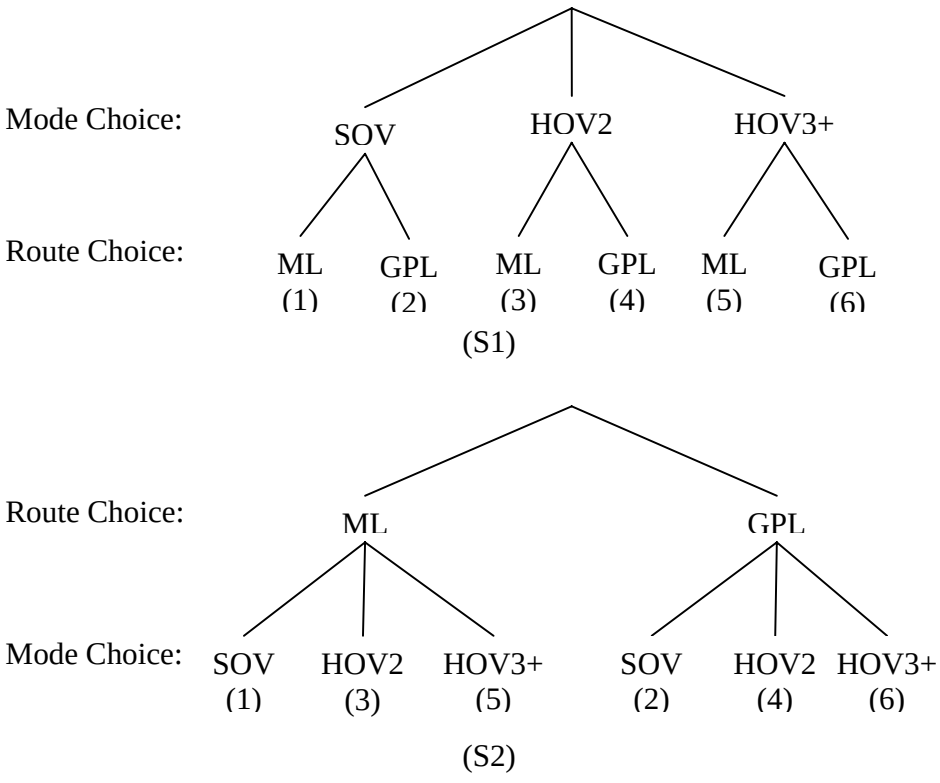



Figure 4-1 Nested Logit Structures (S1) Mode then Route Choice (S2) Route then Mode Choice

The author expects that the first proposed nested structure will result in better model estimation, because generally mode choice is considered to be a stronger choice compared to lane or route choice. The mode choice may be influenced by long term socio-economic considerations and modal preferences, while route choice may be a product of the dynamic conditions of the traffic, recurrent or nonrecurrent congestion, and past experiences.

Therefore, one may expect that the stated preference for MLs, as an independent choice from mode, should be strongly influenced by alternative as well as decision maker characteristics.

*4.3.1 Modeling the Stated Preference in Dallas/Fort-Worth*

Table 4-15 summarizes the results obtained from estimating various models using the DFW dataset. The estimated generic multinomial Logit model appears to be a poor model based on the resulting negative corrected overall measure for goodness of fit, as defined earlier. The percent correct for this model are also very low at 28.4 and 28.8 percent for estimation and validation data, respectively. The same model with alternative specific parameters had much better performance with $\rho^2$ equal to 0.4430 and corrected measure of 0.2337. The sharp decline in the corrected goodness of fit statistic is an indication of disproportionate frequencies of selected alternatives in the estimation dataset.

The percent correct of the alternative specific model is much better than the generic one at 60.3 and 55.2 percent for estimation and validation data, respectively. However, these values are not indicative of a model with strong predictive abilities.

Therefore, the consideration of choice models between pairs and or triplets of alternatives may be beneficial.

Table 4-15 Summary of Models Estimated for Dallas/Fort Worth Metroplex.

| | | MNL | | Nested Logit | | | |
|---|---|---|---|---|---|---|---|
| | | Generic | Alternative Specific | S1 | | S2 | |
| | | | | Generic | Alternative Specific | Generic | Alternative Specific |
| # Observations | | 21522 | 21522 | 21522 | | 23764 | 21522 |
| # Cases | | 129132 | 129132 | 129132 | | 142584 | 129132 |
| # Iterations* | | 17 | 418 | 184 | | 39 | 410 |
| $L(0)$ | | -38562 | -38562 | -38562 | | -42579 | -38562 |
| $L(C)$ | | -28030 | -28030 | -28030 | | -31711 | -28030 |
| $L(\beta)$ | | -37862 | -21480 | -26633 | | -34056 | -21496 |
| $\rho^2$ | | 0.0182 | 0.4430 | 0.3094 | | 0.2002 | 0.4426 |
| $\rho_c^2$ | | -0.3508 | 0.2337 | 0.0498 | | -0.0739 | 0.2331 |
| $\varphi_1$ | | | | 2.6444 | | 1.05E-08 | 0.3282 |
| $\varphi_2$ | | | | 2.3921 | | 1.05E-08 | 0.505 |
| $\varphi_3$ | | | | 2.1327 | | | |
| Percent Correct | Estimation | 28.4 | 60.3 | | | | 60.1 |
| | Validation | 28.8 | 55.2 | | | | 55.9 |

(*) Models are estimated using the Dual Quasi-Newton algorithm.

The nested Logit model with the S1 structure and generic parameters have nest coefficients larger than one, which indicates an inappropriate structure has been adopted. Thus, the alternative specific version of the model is not estimated.

On the other hand, the model based on the S2 structure with generic parameters is tested. The model estimation again results in a negative $\rho_c^2$, while the nest coefficients are significantly very small and close to zero. This indicates that while the corresponding decision structure may be very appropriate for the problem under consideration, the imposition of a requirement for generic parameters has led to the reduction of the descriptive capabilities of the estimated model.

The S2 model with alternative specific parameters produces a much better fit with $\rho^2$ and $\rho_c^2$ equal to 0.4426 and 0.2331, respectively. The nest coefficients of the model are safely less than one and percent correct of the model for estimation and validation data are respectively calculated as 60.1 and 55.9 percent.

Based on the results presented in Table 4-15, the MNL model with alternative specific parameters along with the nested logit model built using decision structure S2 with alternative specific parameters have manifested virtually the same level of both descriptive and predictive performance. It is generally expected that an alternative specific model will have a better fit on the data than its generic counterpart, because the former utilizes a specific set of parameters to be estimated for each alternative, while in the latter all the variables in various alternatives' utility functions get the same parameter.

However, due to the simple theory behind the MNL model and its independence from any particular decision making structure and comparatively easier implementation for future applications, the alternative specific MNL model is selected as the best estimated model for describing the combined mode and lane choice behavior in the DFW metroplex. It is known that MNL models are easier to update and interpret compared to the Nested Logit models. For instance, the sensitivity analysis and variable aggregations may be performed in a straight-forward manner for the MNL models. The estimated parameters of the model are shown in Table 4-16. The complete results of the parameter estimation along with their corresponding standard error and t-statistic are presented in Appendix A.

The parameters suggest that the travel time has a significant and negative effect on the choice of all alternatives, except HOV3+ using the ML (A5). However, the toll effect should be considered with the effects of the interaction variables included. For instance, in the case of a SOV choosing to use ML (A1) on a commute trip (pcomtoll=toll) which already is using toll roads in its typical trips (tollptoll=toll), the toll effect can be calculated as the sum of the corresponding separate effects, that is:

(-0.4145) + (0.3702) + (-0.0329) = -0.0772 [utility/dollar]

which, in turn, can be used to calculate the value of time (VOT) for the specific type of user and trip described above,

(-0.0333) [utility/min] ÷ (-0.0772) [utility/dollar] × (60) [min/hr] = 25.88 [dollar/hr],

which is significantly larger than average VOTs obtained from previous studies in DFW. This reinforces the common hypothesis that SOV users of the ML will have higher VOTs than the average driver.

According to the model, males choose all the alternatives with a higher probability compared to their female counterparts, except for the HOV3+ on GPL (A6). Also, the higher the household size, the smaller the chance of using both the SOV and HOV2 modes. However, the size of the household does not exert any significant impact on the choice of HOV3+ on ML (A5). The greater the number of motor vehicles that are owned by a household, the probability of using SOV and HOV2 modes will be higher and the probability of driving in HOV3+ carpools are significantly lower. As expected, support for the ML concept will significantly increase the utilities of the ML related options (A1, A3, A5), as it reduces those of the GPL related options (A2, A4, A6).

Table 4-16 Alternative Specific MNL Model Parameters for DFW Metroplex

| Variable | Alternatives | | | | | |
|---|---|---|---|---|---|---|
| | A1 | A2 | A3 | A4 | A5 | A6 |
| **unit** | 5.6392 | 8.5467 | 5.5533 | 6.7286 | -17.3395 | -8.0787 |
| **ttime** | -0.0333 | -0.0212 | -0.0118 | -0.0307 | 0.00593 | -0.00989 |
| **toll** | -0.4145 | | -19.0565 | | 16.3993 | |
| **gender** | 0.3354 | 0.2046 | 0.334 | 0.2723 | 0.4271 | -1.5733 |
| **hhnum** | -0.0983 | -0.0396 | -0.2344 | -0.5061 | 0.0636 | 0.8135 |
| **motorveh** | 0.3008 | 0.2232 | 0.1914 | -0.1039 | -0.1705 | -0.4318 |
| **mlbin** | 0.9636 | -0.3019 | 0.8829 | -1.3341 | 0.4491 | -0.6629 |
| **fampool** | -1.3264 | -1.5474 | 0.3386 | 1.7889 | 0.7046 | 0.0243 |
| **tollpay** | 0.5112 | 0.145 | -0.0223 | -0.121 | 0.0561 | -0.5362 |
| **age_d1** | -2.4275 | -3.8979 | -2.5822 | -5.0391 | 17.6377 | -3.7135 |
| **age_d2** | -2.5809 | -3.7213 | -2.0474 | -3.6142 | 19.0095 | -7.0681 |
| **age_d3** | -2.7542 | -3.8566 | -2.8784 | -3.5535 | 18.0525 | -5.03 |
| **age_d4** | -3.0449 | -3.8178 | -2.9374 | -3.5597 | 17.6122 | -4.2758 |
| **age_d5** | -2.7415 | -3.5653 | -2.3948 | -2.5944 | 16.4693 | -5.1905 |
| **ethni_d1** | 0.3509 | 0.2526 | 0.5817 | -0.6797 | -0.3143 | -0.1894 |
| **ethni_d2** | -0.1637 | -0.4954 | 0.3418 | 0.6329 | -0.2872 | -0.0257 |
| **ethni_d3** | -0.2056 | -0.1485 | 0.0915 | -1.2826 | 0.4488 | 1.1028 |
| **hhtyp_d1** | -1.2238 | -1.4749 | -2.9383 | -2.77 | -3.8865 | 12.3032 |
| **hhtyp_d2** | -1.5465 | -1.8949 | -3.1179 | -3.6644 | -1.8334 | 12.0623 |
| **hhtyp_d3** | -2.0294 | -2.9252 | -2.3644 | -2.2263 | -3.1054 | 12.6665 |
| **hhtyp_d4** | -2.1185 | -2.5847 | -2.675 | -1.9547 | -2.3757 | 11.7129 |
| **hhtyp_d5** | -2.4123 | -2.6077 | -3.0267 | -1.8409 | -3.0017 | 12.8984 |
| **occtyp_d1** | 0.3769 | 0.167 | 0.6471 | 0.991 | 0.3805 | -2.558 |
| **occtyp_d2** | 0.4878 | 0.2862 | 0.4112 | 0.6139 | -0.311 | -1.4829 |
| **occtyp_d3** | 0.4451 | 0.3972 | 0.5399 | 0.6014 | -0.3082 | -1.6707 |
| **occtyp_d4** | 0.5404 | 0.5651 | 0.7444 | 1.9135 | -0.00772 | -3.7561 |
| **occtyp_d5** | 4.1267 | 9.1155 | -8.1107 | -2.5819 | 8.71 | -11.2596 |
| **occtyp_d6** | 5.2685 | 4.8476 | 5.5373 | 6.4578 | -11.5656 | -10.5487 |
| **occtyp_d7** | 6.0755 | 5.11 | 6.3637 | -6.5253 | 6.1875 | -17.2286 |
| **occtyp_d8** | -0.6103 | -0.287 | -1.3446 | 3.1653 | 0.1155 | -1.019 |
| **occtyp_d9** | 0.5459 | 0.1259 | 0.2453 | 0.5583 | 0.3451 | -1.8317 |
| **occtyp_d10** | 6.4143 | 6.7383 | 5.779 | -6.7188 | -17.8195 | 5.6438 |
| **occtyp_d11** | -0.7399 | -1.0316 | 0.4117 | 1.3807 | 1.9059 | -1.9553 |
| **educ_d1** | -1.3065 | -1.7247 | -2.4058 | 0.9621 | 0.6552 | 3.8009 |
| **educ_d2** | -1.2132 | -0.9541 | -0.8834 | 0.4152 | 0.9143 | 1.7166 |
| **educ_d3** | -0.5069 | -0.7368 | -0.4093 | -0.2627 | 0.535 | 1.3852 |
| **educ_d4** | -0.5456 | -0.5174 | -0.7912 | -0.4893 | 0.9746 | 1.3665 |
| **income_d123** | -0.1956 | -0.3141 | 1.1902 | 0.357 | 1.4131 | -2.4592 |
| **income_d4** | -0.7002 | 0.0187 | -0.0202 | 0.3414 | 0.2729 | 0.0977 |
| **income_d5** | 0.0425 | 0.7486 | 1.2224 | 0.9278 | 1.0347 | -3.9821 |
| **income_d6** | -0.3716 | 0.0139 | -0.0173 | 0.0851 | 0.174 | 0.1104 |
| **income_d7** | -0.51 | -0.129 | -0.0251 | 0.2118 | 0.2102 | 0.2347 |
| **income_d8** | -0.3687 | 0.0136 | -0.0296 | 0.1781 | -0.3571 | 0.5573 |
| **purpose_d1** | 0.2145 | 0.6261 | -0.4783 | 0.8857 | -0.7945 | -0.5071 |
| **purpose_d2** | 0.134 | 0.4581 | -0.7814 | 0.469 | 0.3032 | -0.6653 |
| **purpose_d3** | 0.6084 | 0.1219 | -0.3995 | -1.5419 | 0.0835 | 1.0989 |
| **pcomtoll** | 0.3702 | | 18.8432 | | -16.2426 | |
| **pwrktoll** | 0.4013 | | 19.0604 | | -20.2579 | |
| **pschtoll** | -0.0447 | | 19.0842 | | -15.754 | |
| **prectoll** | 0.2786 | | 18.7918 | | -16.5432 | |
| **tollptoll** | -0.0329 | | -0.0705 | | -0.2684 | |

- Dark cells indicate variables which are not significant at 0.05 level.

Forming carpools with other family members as a typical means of traveling around will significantly decrease the utility of SOV related modes (A1, A2), while at the same time it will increase the utilities of HOV modes. However, it does not have a significant effect on the utility of HOV3+ on GPL (A6). In general, being a regular toll road user increases the utility of driving a SOV, in particular on ML (A1) compared to on GPL (A2). On the other hand, it will significantly reduce the utility of HOV3+ on GPL (A6).

Ages less than 65 have significantly smaller utilities in both SOV and HOV2 alternatives (A1, A2, A3, A4). However, being younger than 65 has a mixed effect on the HOV3+ modes. It will increase the utility associated with HOV3+ on ML (A5) while decreasing that of HOV3+ on GPL (A6).

The estimated parameters for the ethnicity variable suggest that Caucasians have a higher utility in making a SOV choice (A1, A2) than African Americans or Hispanics. Also, Hispanics are more likely to choose HOV3+ modes (A5, A6) than any other ethnic group considered in the study.

The household type has a significant role in the model. Obviously, singles have more tendency to choose SOV modes than other types. Students and retirees have smaller utilities in SOV use, while employees in service and manufacturing, along with the unemployed and stay home people have a higher utility in using SOV alternatives.

Education level has a significant role in the utility of various alternatives. It should be noted that the coefficients associated with the education dummy variables are relative to the post graduate level with a coefficient of zero. Thus, decreasing negative

coefficients as the level of education increases means more utility is attributed to the SOV mode related alternatives by higher educated people. This trend typically reverses for the HOV2 and HOV3+ related alternatives.

Income levels, overall, are not having much significance on the choice of HOV alternatives. However, in the case of the SOV alternatives, it is generally expected that as the income level increases, the utility of choosing the ML also increases. The results show a few exceptions in this trend. This issue may be alleviated through some further aggregation of income levels, which are currently examined separately in the model.

Work related trips do not have a significant impact on the utility of SOV for the ML choice (A1), but for commute and school trip purpose the utility of this choice increases. In the SOV on GPL alternative (A2), all trip purposes are significant, but commute and work related trips have a larger impact on the utility. School trips are not significantly affecting the utility associated with HOV2 using the ML alternative (A3). Finally, HOV3+ using the ML alternative (A5), work related and school trips are shown not to be significant in changing the utility compared to the other trip purposes.

*4.3.2 Modeling the Stated Preference in Houston*

For the Houston dataset, a summary of the estimated models is presented in Table 4-17. The MNL model with generic parameters once again has a poor fit, which is indicated by a low $\rho^2$ and a negative $\rho_c^2$. On the contrary, the alternative specific MNL model produces a much better fit to the data, and it also performed satisfactorily when considering the percent correct for the estimation and validation datasets with 53.5 and 48.1 percent, respectively.

Table 4-17 Summary of Models Estimated for Houston

| | | MNL | | Nested Logit | | | |
|---|---|---|---|---|---|---|---|
| | | Generic | Alternative Specific | S1 | | S2 | |
| | | | | Generic | Alternative Specific | Generic | Alternative Specific |
| # Observations | | 17514 | 17514 | 17514 | | 17514 | 17514 |
| # Cases | | 105084 | 105084 | 105084 | | 105084 | 105084 |
| # Iterations* | | 20 | 261 | 251 | | 94 | 369 |
| $L(0)$ | | -31381 | -31381 | -31381 | | -31381 | -31381 |
| $L(C)$ | | -24905 | -24905 | -24905 | | -24905 | -24905 |
| $L(\beta)$ | | -30726 | -20358 | -24072 | | -27870 | -20342 |
| $\rho^2$ | | 0.0209 | 0.3513 | 0.2329 | | 0.1119 | 0.3518 |
| $\rho_c^2$ | | -0.2337 | 0.1826 | 0.0334 | | -0.1191 | 0.1832 |
| $\varphi_1$ | | | | 1.1378 | | 0.0822 | 1.6717 |
| $\varphi_2$ | | | | 0.9348 | | 0.1006 | 0.6739 |
| $\varphi_3$ | | | | 0.8304 | | | |
| Percent Correct | Estimation | | 53.5 | | | 43.2 | |
| | Validation | | 48.1 | | | 43.3 | |

(*) Models are estimated using the Dual Quasi-Newton algorithm.

The Nested Logit model with the S1 decision structure and generic parameters results in large nest coefficients with actually one of them, associated with the SOV nest, being larger than one making it an inappropriate structure for modeling the choice behavior. Considering the S2 structure; however, gives rise to a generic model with appropriate nest structure, but a weak overall data fit. Its corresponding alternative specific model, on the contrary, has a better overall fit but suffers from a larger than one nest coefficient for the ML nest. Therefore, in this case, the MNL model with alternative specific parameters is selected to represent the choice behavior for the Houston model.

A summary of the parameters estimated for the selected model is presented in Table 4-18. The full version of the estimation results along with the standard error and t-statistics of the parameters is given in Appendix B.

The results indicate that travel times have a significantly negative effect on the utility of all alternatives, as expected. On the contrary, the amount of toll charged for ML use is only significant in the SOV case (A1). However, considering the interaction variables, the one associated with the school trip purpose is the only significant one in the utility of ML choice for SOV and HOV3+ (A1, A5). Being a toll road user also affects the amount of toll charged to use ML with SOV as well as HOV2 (A1, A3). Based on the results, the value of time for SOV commuters choosing to use the ML (A1), who currently are using toll roads, is given in the following calculations,

(-0.1365) + (0.0425) = -0.094 [utility/dollar], and

(-0.0417) [utility/min] ÷ (-0.094) [utility/dollar] × (60) [min/hr] = 26.62 [dollar/hr].

Gender is only influential for the SOV and HOV2 choices. Household size is significant in all SOV (A1, A2) decisions as well as HOV decisions when they consider MLs (A3, A5). A household's motor vehicle ownership is not significant in HOV2 decisions (A3, A4) and HOV3+ on GPL decisions (A6). Support for ML facilities is significant in the selection of all alternatives, except for the use of ML by HOV3+ mode (A5). Its associated signs are also compatible with the hypothesis that the support for ML will increase the probability of its use, as the ML related alternatives (A1, A3, A5) all have positive parameters, while the rest are negative.

Table 4-18 Alternative Specific MNL Model Parameters for Houston

| Variables | Alternatives | | | | | |
|---|---|---|---|---|---|---|
| | A1 | A2 | A3 | A4 | A5 | A6 |
| unit | -0.6636 | 2.6034 | -1.6927 | 3.9648 | -1.0959 | -2.033 |
| ttime | -0.0417 | -0.0312 | -0.0185 | -0.0285 | -0.0189 | -0.0239 |
| toll | -0.1365 | | -2.2549 | | -1.6374 | |
| gender | 0.3146 | 0.1624 | 0.2879 | -0.4847 | -0.00601 | -0.2757 |
| hhnum | -0.1288 | -0.2318 | 0.1564 | -0.00703 | 0.2443 | -0.0335 |
| motorveh | 0.0977 | 0.1724 | -0.0495 | -0.0715 | -0.154 | 0.006878 |
| mlbin | 1.5331 | -0.4384 | 0.4827 | -1.1514 | 0.1304 | -0.5299 |
| fampool | -1.6273 | -1.5653 | 0.4021 | 1.2599 | 0.7177 | 0.8004 |
| tollpay | -0.1415 | -0.1476 | -0.1567 | 0.2375 | 0.0706 | 0.1114 |
| age_d1 | 0.7639 | 0.0698 | 1.5481 | 0.0118 | -0.8481 | -1.5633 |
| age_d2 | 0.5775 | -0.1295 | 0.773 | -0.6977 | -1.0465 | 0.5074 |
| age_d3 | 0.4241 | -0.0469 | 0.9164 | -0.4318 | -0.6214 | -0.2557 |
| age_d4 | 0.4035 | 0.1373 | 1.0924 | -0.4562 | -0.9463 | -0.2377 |
| age_d5 | -0.0123 | -0.2577 | 0.5451 | -0.3848 | -0.4921 | 0.581 |
| ethni_d1 | 0.5574 | -0.2727 | -0.1479 | -1.382 | -0.4128 | 1.6759 |
| ethni_d2 | 0.1819 | -0.4046 | -0.3195 | -1.876 | 0.1538 | 2.2728 |
| ethni_d3 | 0.4493 | -0.0117 | 0.1134 | -1.9673 | -0.1651 | 1.5994 |
| hhtyp_d1 | 0.2163 | 0.1646 | 0.403 | -0.7404 | 0.6349 | -0.6852 |
| hhtyp_d2 | 0.3801 | 0.1627 | 0.4478 | -1.1667 | 0.4932 | -0.3116 |
| hhtyp_d3 | 0.0161 | 0.024 | 0.6093 | 0.0225 | 1.7199 | -2.4119 |
| hhtyp_d4 | 0.2155 | 0.224 | -0.0176 | -0.0307 | 0.5285 | -0.9227 |
| hhtyp_d5 | -0.00059 | -0.0221 | 0.4725 | 0.1256 | 0.2919 | -0.8762 |
| occtyp_d1 | 0.34 | 0.4792 | 0.2902 | -1.1826 | 0.3486 | -0.2635 |
| occtyp_d2 | 0.337 | 0.6479 | -0.1963 | -1.181 | 0.5636 | -0.1574 |
| occtyp_d3 | 0.3752 | 0.9515 | 0.1006 | 0.7297 | -1.0113 | -1.1233 |
| occtyp_d4 | 0.2063 | 0.1399 | 0.2778 | -0.4612 | 0.6109 | -0.7689 |
| occtyp_d5 | 1.8859 | 2.5149 | 1.9159 | 1.0036 | -4.0835 | -3.2334 |
| occtyp_d6 | 0.3436 | 0.1396 | 0.1718 | -1.0568 | 1.258 | -0.8627 |
| occtyp_d7 | 0.4002 | 0.8805 | 0.6428 | -1.6773 | 1.3582 | -1.5863 |
| occtyp_d8 | 0.5797 | 0.7145 | 0.7302 | -1.274 | 1.1971 | -1.8735 |
| occtyp_d9 | 0.6306 | 0.2785 | -0.2032 | -1.5398 | 0.8854 | -0.0272 |
| occtyp_d10 | -0.2923 | -0.1339 | 0.8378 | -1.068 | 0.3998 | 0.2466 |
| occtyp_d11 | 0.8889 | 1.433 | 1.3615 | -0.5421 | -2.1302 | -0.9835 |
| educ_d1 | -0.3486 | 0.0876 | -1.7918 | 0.5549 | 0.4925 | 1.0241 |
| educ_d2 | -0.3654 | -0.7248 | -0.3268 | -0.0226 | 0.2712 | 1.1583 |
| educ_d3 | -0.6177 | -0.3319 | -0.154 | -0.3283 | -0.0689 | 1.4952 |
| educ_d4 | -0.248 | -0.0219 | 0.1919 | -0.2826 | -0.0349 | 0.3977 |
| income_d123 | -0.4227 | -0.00693 | 0.587 | -0.4339 | 0.31 | -0.0374 |
| income_d4 | 0.5107 | 0.8047 | 0.8449 | 0.1802 | -0.0744 | -2.2526 |
| income_d5 | -0.2355 | 0.3851 | 0.127 | -0.1246 | 0.2643 | -0.4065 |
| income_d6 | -0.1447 | -0.0174 | 0.4711 | -0.5527 | 0.366 | -0.1225 |
| income_d7 | -0.2252 | -0.1113 | 0.4127 | 0.2457 | -0.00091 | -0.3161 |
| income_d8 | -0.2245 | -0.0701 | 0.0969 | -0.1308 | 0.5411 | -0.2125 |
| purpose_d1 | 0.2472 | 0.2355 | -0.1203 | -0.1757 | -0.0978 | -0.0949 |
| purpose_d2 | 0.29 | 0.3854 | -0.6088 | 0.4241 | 0.1328 | -0.6669 |
| purpose_d3 | 0.5405 | -0.6432 | -1.7532 | -0.2347 | -0.2292 | 2.3117 |
| pcomtoll | 0.0233 | | 2.0105 | | 1.491 | |
| pwrktoll | 0.0493 | | 2.2153 | | 1.5084 | |
| pschtoll | -0.1375 | | 1.827 | | -4.439 | |
| prectoll | 0.0503 | | 1.9312 | | 0.9796 | |
| tollptoll | 0.0425 | | 0.0875 | | 0.0271 | |

- Dark cells indicate variables which are not significant at 0.05 level.

The results also indicate that carpooling with family members will decrease the utility of the SOV mode using either lane type; however, it actually increases the probability of choosing other carpooling modes. The current experience with toll road use and actual toll payment is significant in reducing the utility of the SOV mode related alternatives (A1, A2). That is the case also with the ML choice for HOV2 (A3), while it increases the probability of using GPL for HOV2 (A4). Being a regular toll road user does not have any significant effect in the utility of lane choices for the HOV3+ mode (A5, A6).

In general, age is a significant factor in the utility of alternatives associated with SOV, and also in the ML choices of other modes. Overall, younger people seem to have a higher propensity to choose the ML.

The results suggest that the ethnic background of individuals also is partially significant in their decisions. This is especially evident in the SOV related lane choices (A1, A2) and the choice of GPL by other modes (A4, A6). In the SOV case, white Caucasians and Hispanics, by far, have shown more interest in using ML, while Asians and others are more likely to use GPL. African Americans have less interest, than other ethnic groups, in driving on ML as HOV2 (A3). The same behavior is shown to be the case with Caucasians in the case of HOV3+ mode using ML (A5).

Single parents are more interested in using ML as SOVs (A1), but married households with children (household size larger than 3) are more likely, compared to other household types, to use GPL as SOV. The latter, also, is less likely to use ML as HOV2. Understandably, single parents and single households are less likely to use GPL

than ML as HOV2 and HOV3+, which implies that they may be more likely to form carpools with the objective of accessing the ML.

Service type occupations have a higher tendency to use either ML or GPL as SOV and or HOV2 modes. Education level, income and trip purpose have been less significant in the utility of the alternatives associated with the HOV modes. As a general trend, an increase in the educational level and the income level increases the utility of SOV on ML (A1) and decreases the utility of SOV on GPL (A2). Compared to other trip purposes, school trips are shown to have a higher tendency to use SOV on ML (A1), as well as HOV3+ on GPL (A6). But, they have significantly less utility in adopting ML with HOV2 (A3).

*4.3.3 Comparison between Dallas/Fort-Worth and Houston Stated Preferences*

In the DFW model, tolls, per se, are significant in the utilities of all ML related alternatives (A1, A3, A5), while in the Houston model this is only true for the SOV mode using ML (A1). Taking into consideration all the toll related interaction variables and their associated parameters; the toll amount in the DFW model plays a more significant role in the decisions of the potential ML users than in the Houston model, although, Houston currently has an extensive network of toll roads as compared to the DFW.

Furthermore, the current toll road users in DFW have a significantly higher utility in using either ML or GPL as SOV (A1, A2). On the contrary, the current toll road users in Houston have a significantly smaller utility for SOV related alternatives. Effectively, this implies that toll road users in DFW are looking forward to the

introduction of ML facilities as an additional incentive for them to maintain their SOV status, while in Houston they are reacting by showing their disinterest in being SOV and instead their desire to be HOV2 using GPL (A4).

## 4.4 Proposals for Future Research

### *4.4.1 Modeling the Stated Preference for Binary Choices within Modes*

An alternative to the modeling effort is to decompose the datasets based on the modal stated preferences and then estimate a lane choice model for each mode, separately. This is expected to be helpful given the fact that in this study scenarios are designed based on a small number of variables, that is travel time and toll, which are not very significant in the mode choice decisions. In general, mode (occupancy) choice is more a function of socio-demographic attributes of the respondent and her typical trip characteristics and habits. For each mode, the current modal use of respondents should also be taken into account. It is suspected that, to a large extent, respondents stick to their current modes in their stated preferences.

### *4.4.2 Modeling the Combined Stated Preference in Dallas/Fort-Worth and Houston*

It is desirable to estimate a model for the combined datasets of DFW and Houston. The estimated model can be used as the Texas model.

### *4.4.3 Modeling the Stated Preference for Mode Choices*

The datasets can alternatively be used to estimate models for the respondents' mode choice behavior. In fact, there are two possibilities for doing the mode choice modeling. First, one can consider the respondents' current mode choice as the dependent variable to be modeled. The resulting revealed preference model may have

better qualities as opposed to the stated preference models, which may be obtained using the modal stated preferences in the sample.

*4.4.4 Modeling the Stated Preference Using Probit Model*

In this study, Logit and Nested Logit models were used to estimate the models. As stated earlier, it is known that other models, such as Probit, may have better performance in modeling the discrete choice behavior as compared to the models of Logit family. Therefore, it is proposed that Probit models be estimated for the stated preferences and their performance be examined and compared to the Logit models. SAS is capable of estimating Probit models as well.

# APPENDIX A

## DALLAS-FORT WORTH ALTERNATIVE SPECIFIC MULTINOMIAL LOGIT MODEL PARAMETER ESTIMATES

| Parameter | DF | Estimate | Standard Error | t Value | Approx Pr > \|t\| |
|---|---|---|---|---|---|
| unit_1 | 1 | 5.6392 | 0.2417 | 23.33 | <.0001 |
| unit_2 | 0 | 8.5467 | . | . | . |
| unit_3 | 1 | 5.5533 | 0.3002 | 18.5 | <.0001 |
| unit_4 | 1 | 6.7286 | 0.515 | 13.06 | <.0001 |
| unit_5 | 0 | -17.3395 | . | . | . |
| unit_6 | 0 | -8.0787 | . | . | . |
| ttime_1 | 1 | -0.0333 | 0.00367 | -9.07 | <.0001 |
| ttime_2 | 1 | -0.0212 | 0.001409 | -15.03 | <.0001 |
| ttime_3 | 1 | -0.0118 | 0.003971 | -2.98 | 0.0029 |
| ttime_4 | 1 | -0.0307 | 0.00256 | -12 | <.0001 |
| ttime_5 | 1 | 0.00593 | 0.004522 | 1.31 | 0.1898 |
| ttime_6 | 1 | -0.00989 | 0.004932 | -2.01 | 0.045 |
| toll_1 | 1 | -0.4145 | 0.085 | -4.87 | <.0001 |
| toll_3 | 0 | -19.0565 | . | . | . |
| toll_5 | 0 | 16.3993 | . | . | . |
| gender_1 | 1 | 0.3354 | 0.0409 | 8.19 | <.0001 |
| gender_2 | 0 | 0.2046 | . | . | . |
| gender_3 | 1 | 0.334 | 0.0554 | 6.03 | <.0001 |
| gender_4 | 1 | 0.2723 | 0.0996 | 2.73 | 0.0062 |
| gender_5 | 1 | 0.4271 | 0.0979 | 4.36 | <.0001 |
| gender_6 | 1 | -1.5733 | 0.2002 | -7.86 | <.0001 |
| hhnum_1 | 1 | -0.0983 | 0.0218 | -4.51 | <.0001 |
| hhnum_2 | 0 | -0.0396 | . | . | . |
| hhnum_3 | 1 | -0.2344 | 0.0318 | -7.37 | <.0001 |
| hhnum_4 | 1 | -0.5061 | 0.0652 | -7.76 | <.0001 |
| hhnum_5 | 1 | 0.0636 | 0.044 | 1.45 | 0.1476 |
| hhnum_6 | 1 | 0.8135 | 0.0811 | 10.03 | <.0001 |
| motorveh_1 | 0 | 0.3008 | . | . | . |
| motorveh_2 | 1 | 0.2232 | 0.021 | 10.64 | <.0001 |
| motorveh_3 | 1 | 0.1914 | 0.0326 | 5.86 | <.0001 |
| motorveh_4 | 1 | -0.1039 | 0.0666 | -1.56 | 0.1186 |
| motorveh_5 | 1 | -0.1705 | 0.0602 | -2.83 | 0.0047 |
| motorveh_6 | 1 | -0.4318 | 0.1231 | -3.51 | 0.0005 |
| mlbin_1 | 0 | 0.9636 | . | . | . |
| mlbin_2 | 1 | -0.3019 | 0.0499 | -6.05 | <.0001 |
| mlbin_3 | 1 | 0.8829 | 0.0745 | 11.85 | <.0001 |
| mlbin_4 | 1 | -1.3341 | 0.0971 | -13.74 | <.0001 |
| mlbin_5 | 1 | 0.4491 | 0.1136 | 3.95 | <.0001 |
| mlbin_6 | 1 | -0.6629 | 0.1897 | -3.49 | 0.0005 |
| fampool_1 | 1 | -1.3264 | 0.0888 | -14.93 | <.0001 |
| fampool_2 | 1 | -1.5474 | 0.0797 | -19.42 | <.0001 |
| fampool_3 | 0 | 0.3386 | . | . | . |

| Parameter | DF | Estimate | Standard Error | t Value | Approx Pr > \|t\| |
|---|---|---|---|---|---|
| fampool_4 | 1 | 1.7889 | 0.121 | 14.78 | <.0001 |
| fampool_5 | 1 | 0.7046 | 0.1161 | 6.07 | <.0001 |
| fampool_6 | 1 | 0.0243 | 0.2341 | 0.1 | 0.9174 |
| tollpay_1 | 1 | 0.5112 | 0.0486 | 10.52 | <.0001 |
| tollpay_2 | 0 | 0.145 | . | . | . |
| tollpay_3 | 1 | -0.0223 | 0.0617 | -0.36 | 0.7171 |
| tollpay_4 | 1 | -0.121 | 0.0958 | -1.26 | 0.2065 |
| tollpay_5 | 1 | 0.0561 | 0.092 | 0.61 | 0.5419 |
| tollpay_6 | 1 | -0.5362 | 0.2007 | -2.67 | 0.0075 |
| age_d1_1 | 0 | -2.4275 | . | . | . |
| age_d2_1 | 1 | -2.5809 | 0.16 | -16.14 | <.0001 |
| age_d3_1 | 1 | -2.7542 | 0.1603 | -17.18 | <.0001 |
| age_d4_1 | 1 | -3.0449 | 0.1592 | -19.13 | <.0001 |
| age_d5_1 | 0 | -2.7415 | . | . | . |
| age_d1_2 | 1 | -3.8979 | 0.1706 | -22.85 | <.0001 |
| age_d2_2 | 0 | -3.7213 | . | . | . |
| age_d3_2 | 0 | -3.8566 | . | . | . |
| age_d4_2 | 0 | -3.8178 | . | . | . |
| age_d5_2 | 1 | -3.5653 | 0.1641 | -21.73 | <.0001 |
| age_d1_3 | 1 | -2.5822 | 0.2251 | -11.47 | <.0001 |
| age_d2_3 | 1 | -2.0474 | 0.178 | -11.5 | <.0001 |
| age_d3_3 | 1 | -2.8784 | 0.1811 | -15.9 | <.0001 |
| age_d4_3 | 1 | -2.9374 | 0.1763 | -16.66 | <.0001 |
| age_d5_3 | 1 | -2.3948 | 0.2131 | -11.24 | <.0001 |
| age_d1_4 | 1 | -5.0391 | 0.3444 | -14.63 | <.0001 |
| age_d2_4 | 1 | -3.6142 | 0.2515 | -14.37 | <.0001 |
| age_d3_4 | 1 | -3.5535 | 0.2534 | -14.02 | <.0001 |
| age_d4_4 | 1 | -3.5597 | 0.2432 | -14.63 | <.0001 |
| age_d5_4 | 1 | -2.5944 | 0.2674 | -9.7 | <.0001 |
| age_d1_5 | 1 | 17.6377 | 0.4804 | 36.72 | <.0001 |
| age_d2_5 | 1 | 19.0095 | 0.4231 | 44.93 | <.0001 |
| age_d3_5 | 1 | 18.0525 | 0.4406 | 40.98 | <.0001 |
| age_d4_5 | 1 | 17.6122 | 0.4397 | 40.05 | <.0001 |
| age_d5_5 | 1 | 16.4693 | 0.5171 | 31.85 | <.0001 |
| age_d1_6 | 1 | -3.7135 | 0.5608 | -6.62 | <.0001 |
| age_d2_6 | 1 | -7.0681 | 0.557 | -12.69 | <.0001 |
| age_d3_6 | 1 | -5.03 | 0.4724 | -10.65 | <.0001 |
| age_d4_6 | 1 | -4.2758 | 0.4409 | -9.7 | <.0001 |
| age_d5_6 | 1 | -5.1905 | 0.5394 | -9.62 | <.0001 |
| ethni_d1_1 | 1 | 0.3509 | 0.0884 | 3.97 | <.0001 |
| ethni_d2_1 | 0 | -0.1637 | . | . | . |
| ethni_d3_1 | 1 | -0.2056 | 0.1044 | -1.97 | 0.0488 |
| ethni_d1_2 | 0 | 0.2526 | . | . | . |

| Parameter | DF | Estimate | Standard Error | t Value | Approx Pr > \|t\| |
|---|---|---|---|---|---|
| ethni_d2_2 | 1 | -0.4954 | 0.1039 | -4.77 | <.0001 |
| ethni_d3_2 | 0 | -0.1485 | . | . | . |
| ethni_d1_3 | 1 | 0.5817 | 0.1276 | 4.56 | <.0001 |
| ethni_d2_3 | 1 | 0.3418 | 0.1535 | 2.23 | 0.026 |
| ethni_d3_3 | 1 | 0.0915 | 0.1449 | 0.63 | 0.5279 |
| ethni_d1_4 | 1 | -0.6797 | 0.1633 | -4.16 | <.0001 |
| ethni_d2_4 | 1 | 0.6329 | 0.1983 | 3.19 | 0.0014 |
| ethni_d3_4 | 1 | -1.2826 | 0.2378 | -5.39 | <.0001 |
| ethni_d1_5 | 1 | -0.3143 | 0.1783 | -1.76 | 0.078 |
| ethni_d2_5 | 1 | -0.2872 | 0.2036 | -1.41 | 0.1584 |
| ethni_d3_5 | 1 | 0.4488 | 0.187 | 2.4 | 0.0164 |
| ethni_d1_6 | 1 | -0.1894 | 0.4688 | -0.4 | 0.6863 |
| ethni_d2_6 | 1 | -0.0257 | 0.5195 | -0.05 | 0.9605 |
| ethni_d3_6 | 1 | 1.1028 | 0.5014 | 2.2 | 0.0278 |
| hhtyp_d1_1 | 1 | -1.2238 | 0.1101 | -11.11 | <.0001 |
| hhtyp_d2_1 | 1 | -1.5465 | 0.1274 | -12.14 | <.0001 |
| hhtyp_d3_1 | 0 | -2.0294 | . | . | . |
| hhtyp_d4_1 | 1 | -2.1185 | 0.1173 | -18.06 | <.0001 |
| hhtyp_d5_1 | 1 | -2.4123 | 0.1153 | -20.93 | <.0001 |
| hhtyp_d1_2 | 0 | -1.4749 | . | . | . |
| hhtyp_d2_2 | 0 | -1.8949 | . | . | . |
| hhtyp_d3_2 | 1 | -2.9252 | 0.1215 | -24.08 | <.0001 |
| hhtyp_d4_2 | 0 | -2.5847 | . | . | . |
| hhtyp_d5_2 | 0 | -2.6077 | . | . | . |
| hhtyp_d1_3 | 1 | -2.9383 | 0.1218 | -24.12 | <.0001 |
| hhtyp_d2_3 | 1 | -3.1179 | 0.157 | -19.86 | <.0001 |
| hhtyp_d3_3 | 1 | -2.3644 | 0.1456 | -16.24 | <.0001 |
| hhtyp_d4_3 | 1 | -2.675 | 0.1311 | -20.41 | <.0001 |
| hhtyp_d5_3 | 1 | -3.0267 | 0.1258 | -24.07 | <.0001 |
| hhtyp_d1_4 | 1 | -2.77 | 0.2598 | -10.66 | <.0001 |
| hhtyp_d2_4 | 1 | -3.6644 | 0.356 | -10.29 | <.0001 |
| hhtyp_d3_4 | 1 | -2.2263 | 0.282 | -7.89 | <.0001 |
| hhtyp_d4_4 | 1 | -1.9547 | 0.2604 | -7.51 | <.0001 |
| hhtyp_d5_4 | 1 | -1.8409 | 0.2459 | -7.49 | <.0001 |
| hhtyp_d1_5 | 1 | -3.8865 | 0.2444 | -15.9 | <.0001 |
| hhtyp_d2_5 | 1 | -1.8334 | 0.2375 | -7.72 | <.0001 |
| hhtyp_d3_5 | 1 | -3.1054 | 0.2569 | -12.09 | <.0001 |
| hhtyp_d4_5 | 1 | -2.3757 | 0.2334 | -10.18 | <.0001 |
| hhtyp_d5_5 | 1 | -3.0017 | 0.2394 | -12.54 | <.0001 |
| hhtyp_d1_6 | 1 | 12.3032 | 0.9548 | 12.89 | <.0001 |
| hhtyp_d2_6 | 1 | 12.0623 | 1.0009 | 12.05 | <.0001 |
| hhtyp_d3_6 | 1 | 12.6665 | 1.0024 | 12.64 | <.0001 |
| hhtyp_d4_6 | 1 | 11.7129 | 0.9623 | 12.17 | <.0001 |

| Parameter | DF | Estimate | Standard Error | t Value | Approx Pr > \|t\| |
|---|---|---|---|---|---|
| hhtyp_d5_6 | 1 | 12.8984 | 0.9266 | 13.92 | <.0001 |
| occtyp_d1_1 | 1 | 0.3769 | 0.0937 | 4.02 | <.0001 |
| occtyp_d2_1 | 1 | 0.4878 | 0.1027 | 4.75 | <.0001 |
| occtyp_d3_1 | 1 | 0.4451 | 0.1162 | 3.83 | 0.0001 |
| occtyp_d4_1 | 1 | 0.5404 | 0.1042 | 5.19 | <.0001 |
| occtyp_d5_1 | 1 | 4.1267 | 1.0203 | 4.04 | <.0001 |
| occtyp_d6_1 | 1 | 5.2685 | 0.2246 | 23.46 | <.0001 |
| occtyp_d7_1 | 1 | 6.0755 | 0.3129 | 19.42 | <.0001 |
| occtyp_d8_1 | 1 | -0.6103 | 0.1333 | -4.58 | <.0001 |
| occtyp_d9_1 | 1 | 0.5459 | 0.121 | 4.51 | <.0001 |
| occtyp_d10_1 | 1 | 6.4143 | 0.1645 | 38.99 | <.0001 |
| occtyp_d11_1 | 1 | -0.7399 | 0.2418 | -3.06 | 0.0022 |
| occtyp_d1_2 | 0 | 0.167 | . | . | . |
| occtyp_d2_2 | 0 | 0.2862 | . | . | . |
| occtyp_d3_2 | 0 | 0.3972 | . | . | . |
| occtyp_d4_2 | 0 | 0.5651 | . | . | . |
| occtyp_d5_2 | 0 | 9.1155 | . | . | . |
| occtyp_d6_2 | 0 | 4.8476 | . | . | . |
| occtyp_d7_2 | 1 | 5.11 | 0.3077 | 16.6 | <.0001 |
| occtyp_d8_2 | 0 | -0.287 | . | . | . |
| occtyp_d9_2 | 0 | 0.1259 | . | . | . |
| occtyp_d10_2 | 0 | 6.7383 | . | . | . |
| occtyp_d11_2 | 1 | -1.0316 | 0.2059 | -5.01 | <.0001 |
| occtyp_d1_3 | 1 | 0.6471 | 0.1185 | 5.46 | <.0001 |
| occtyp_d2_3 | 1 | 0.4112 | 0.1319 | 3.12 | 0.0018 |
| occtyp_d3_3 | 1 | 0.5399 | 0.1544 | 3.5 | 0.0005 |
| occtyp_d4_3 | 1 | 0.7444 | 0.1337 | 5.57 | <.0001 |
| occtyp_d5_3 | 0 | -8.1107 | . | . | . |
| occtyp_d6_3 | 1 | 5.5373 | 0.2944 | 18.81 | <.0001 |
| occtyp_d7_3 | 0 | 6.3637 | . | . | . |
| occtyp_d8_3 | 1 | -1.3446 | 0.1746 | -7.7 | <.0001 |
| occtyp_d9_3 | 1 | 0.2453 | 0.1543 | 1.59 | 0.1117 |
| occtyp_d10_3 | 1 | 5.779 | 0.2121 | 27.25 | <.0001 |
| occtyp_d11_3 | 0 | 0.4117 | . | . | . |
| occtyp_d1_4 | 1 | 0.991 | 0.2728 | 3.63 | 0.0003 |
| occtyp_d2_4 | 1 | 0.6139 | 0.301 | 2.04 | 0.0414 |
| occtyp_d3_4 | 1 | 0.6014 | 0.3434 | 1.75 | 0.0799 |
| occtyp_d4_4 | 1 | 1.9135 | 0.2854 | 6.7 | <.0001 |
| occtyp_d5_4 | 0 | -2.5819 | . | . | . |
| occtyp_d6_4 | 1 | 6.4578 | 0.3781 | 17.08 | <.0001 |
| occtyp_d7_4 | 0 | -6.5253 | . | . | . |
| occtyp_d8_4 | 1 | 3.1653 | 0.3203 | 9.88 | <.0001 |
| occtyp_d9_4 | 1 | 0.5583 | 0.3275 | 1.7 | 0.0882 |

| Parameter | DF | Estimate | Standard Error | t Value | Approx Pr > \|t\| |
|---|---|---|---|---|---|
| occtyp_d10_4 | 0 | -6.7188 | . | . | . |
| occtyp_d11_4 | 1 | 1.3807 | 0.3613 | 3.82 | 0.0001 |
| occtyp_d1_5 | 1 | 0.3805 | 0.1814 | 2.1 | 0.036 |
| occtyp_d2_5 | 1 | -0.311 | 0.2112 | -1.47 | 0.1408 |
| occtyp_d3_5 | 1 | -0.3082 | 0.2462 | -1.25 | 0.2107 |
| occtyp_d4_5 | 1 | -0.00772 | 0.1995 | -0.04 | 0.9691 |
| occtyp_d5_5 | 1 | 8.71 | 0.4499 | 19.36 | <.0001 |
| occtyp_d6_5 | 0 | -11.5656 | . | . | . |
| occtyp_d7_5 | 1 | 6.1875 | 0.358 | 17.28 | <.0001 |
| occtyp_d8_5 | 1 | 0.1155 | 0.2587 | 0.45 | 0.6552 |
| occtyp_d9_5 | 1 | 0.3451 | 0.2234 | 1.54 | 0.1224 |
| occtyp_d10_5 | 0 | -17.8195 | . | . | . |
| occtyp_d11_5 | 1 | 1.9059 | 0.3833 | 4.97 | <.0001 |
| occtyp_d1_6 | 1 | -2.558 | 0.2894 | -8.84 | <.0001 |
| occtyp_d2_6 | 1 | -1.4829 | 0.3397 | -4.37 | <.0001 |
| occtyp_d3_6 | 1 | -1.6707 | 0.3997 | -4.18 | <.0001 |
| occtyp_d4_6 | 1 | -3.7561 | 0.4917 | -7.64 | <.0001 |
| occtyp_d5_6 | 0 | -11.2596 | . | . | . |
| occtyp_d6_6 | 0 | -10.5487 | . | . | . |
| occtyp_d7_6 | 0 | -17.2286 | . | . | . |
| occtyp_d8_6 | 1 | -1.019 | 0.5096 | -2 | 0.0456 |
| occtyp_d9_6 | 1 | -1.8317 | 0.468 | -3.91 | <.0001 |
| occtyp_d10_6 | 1 | 5.6438 | 0.6784 | 8.32 | <.0001 |
| occtyp_d11_6 | 1 | -1.9553 | 0.4882 | -4 | <.0001 |
| educ_d1_1 | 1 | -1.3065 | 0.2348 | -5.57 | <.0001 |
| educ_d2_1 | 1 | -1.2132 | 0.0912 | -13.3 | <.0001 |
| educ_d3_1 | 0 | -0.5069 | . | . | . |
| educ_d4_1 | 1 | -0.5456 | 0.05 | -10.91 | <.0001 |
| educ_d1_2 | 0 | -1.7247 | . | . | . |
| educ_d2_2 | 0 | -0.9541 | . | . | . |
| educ_d3_2 | 1 | -0.7368 | 0.0571 | -12.9 | <.0001 |
| educ_d4_2 | 0 | -0.5174 | . | . | . |
| educ_d1_3 | 1 | -2.4058 | 0.3616 | -6.65 | <.0001 |
| educ_d2_3 | 1 | -0.8834 | 0.1126 | -7.85 | <.0001 |
| educ_d3_3 | 1 | -0.4093 | 0.0796 | -5.14 | <.0001 |
| educ_d4_3 | 1 | -0.7912 | 0.0687 | -11.51 | <.0001 |
| educ_d1_4 | 1 | 0.9621 | 0.362 | 2.66 | 0.0079 |
| educ_d2_4 | 1 | 0.4152 | 0.1615 | 2.57 | 0.0101 |
| educ_d3_4 | 1 | -0.2627 | 0.1381 | -1.9 | 0.0571 |
| educ_d4_4 | 1 | -0.4893 | 0.1245 | -3.93 | <.0001 |
| educ_d1_5 | 1 | 0.6552 | 0.4102 | 1.6 | 0.1102 |
| educ_d2_5 | 1 | 0.9143 | 0.21 | 4.35 | <.0001 |
| educ_d3_5 | 1 | 0.535 | 0.1806 | 2.96 | 0.003 |

| Parameter | DF | Estimate | Standard Error | t Value | Approx Pr > \|t\| |
|---|---|---|---|---|---|
| educ_d4_5 | 1 | 0.9746 | 0.1598 | 6.1 | <.0001 |
| educ_d1_6 | 1 | 3.8009 | 0.6925 | 5.49 | <.0001 |
| educ_d2_6 | 1 | 1.7166 | 0.5354 | 3.21 | 0.0013 |
| educ_d3_6 | 1 | 1.3852 | 0.4855 | 2.85 | 0.0043 |
| educ_d4_6 | 1 | 1.3665 | 0.4773 | 2.86 | 0.0042 |
| income_d123_1 | 0 | -0.1956 | . | . | . |
| income_d4_1 | 1 | -0.7002 | 0.0939 | -7.46 | <.0001 |
| income_d5_1 | 1 | 0.0425 | 0.0836 | 0.51 | 0.6111 |
| income_d6_1 | 1 | -0.3716 | 0.0771 | -4.82 | <.0001 |
| income_d7_1 | 1 | -0.51 | 0.0807 | -6.32 | <.0001 |
| income_d8_1 | 1 | -0.3687 | 0.0823 | -4.48 | <.0001 |
| income_d123_2 | 1 | -0.3141 | 0.0912 | -3.44 | 0.0006 |
| income_d4_2 | 0 | 0.0187 | . | . | . |
| income_d5_2 | 0 | 0.7486 | . | . | . |
| income_d6_2 | 0 | 0.0139 | . | . | . |
| income_d7_2 | 0 | -0.129 | . | . | . |
| income_d8_2 | 0 | 0.0136 | . | . | . |
| income_d123_3 | 1 | 1.1902 | 0.1389 | 8.57 | <.0001 |
| income_d4_3 | 1 | -0.0202 | 0.1366 | -0.15 | 0.8822 |
| income_d5_3 | 1 | 1.2224 | 0.1229 | 9.95 | <.0001 |
| income_d6_3 | 1 | -0.0173 | 0.1222 | -0.14 | 0.8875 |
| income_d7_3 | 1 | -0.0251 | 0.1275 | -0.2 | 0.8438 |
| income_d8_3 | 1 | -0.0296 | 0.131 | -0.23 | 0.8215 |
| income_d123_4 | 1 | 0.357 | 0.2635 | 1.35 | 0.1755 |
| income_d4_4 | 1 | 0.3414 | 0.2564 | 1.33 | 0.183 |
| income_d5_4 | 1 | 0.9278 | 0.2407 | 3.85 | 0.0001 |
| income_d6_4 | 1 | 0.0851 | 0.2336 | 0.36 | 0.7156 |
| income_d7_4 | 1 | 0.2118 | 0.2331 | 0.91 | 0.3636 |
| income_d8_4 | 1 | 0.1781 | 0.2419 | 0.74 | 0.4616 |
| income_d123_5 | 1 | 1.4131 | 0.2399 | 5.89 | <.0001 |
| income_d4_5 | 1 | 0.2729 | 0.247 | 1.1 | 0.2692 |
| income_d5_5 | 1 | 1.0347 | 0.2224 | 4.65 | <.0001 |
| income_d6_5 | 1 | 0.174 | 0.217 | 0.8 | 0.4226 |
| income_d7_5 | 1 | 0.2102 | 0.2237 | 0.94 | 0.3473 |
| income_d8_5 | 1 | -0.3571 | 0.2368 | -1.51 | 0.1316 |
| income_d123_6 | 1 | -2.4592 | 0.5686 | -4.32 | <.0001 |
| income_d4_6 | 1 | 0.0977 | 0.5552 | 0.18 | 0.8604 |
| income_d5_6 | 1 | -3.9821 | 0.7425 | -5.36 | <.0001 |
| income_d6_6 | 1 | 0.1104 | 0.4945 | 0.22 | 0.8233 |
| income_d7_6 | 1 | 0.2347 | 0.5105 | 0.46 | 0.6457 |
| income_d8_6 | 1 | 0.5573 | 0.4957 | 1.12 | 0.2609 |
| purpose_d1_1 | 1 | 0.2145 | 0.0699 | 3.07 | 0.0021 |
| purpose_d2_1 | 1 | 0.134 | 0.094 | 1.43 | 0.1538 |

| Parameter | DF | Estimate | Standard Error | t Value | Approx Pr > \|t\| |
|---|---|---|---|---|---|
| purpose_d3_1 | 1 | 0.6084 | 0.3076 | 1.98 | 0.0479 |
| purpose_d1_2 | 0 | 0.6261 | . | . | . |
| purpose_d2_2 | 0 | 0.4581 | . | . | . |
| purpose_d3_2 | 0 | 0.1219 | . | . | . |
| purpose_d1_3 | 1 | -0.4783 | 0.0747 | -6.4 | <.0001 |
| purpose_d2_3 | 1 | -0.7814 | 0.1104 | -7.08 | <.0001 |
| purpose_d3_3 | 1 | -0.3995 | 0.2057 | -1.94 | 0.0521 |
| purpose_d1_4 | 1 | 0.8857 | 0.1107 | 8 | <.0001 |
| purpose_d2_4 | 1 | 0.469 | 0.1537 | 3.05 | 0.0023 |
| purpose_d3_4 | 1 | -1.5419 | 0.5432 | -2.84 | 0.0045 |
| purpose_d1_5 | 1 | -0.7945 | 0.1212 | -6.55 | <.0001 |
| purpose_d2_5 | 1 | 0.3032 | 0.1548 | 1.96 | 0.0501 |
| purpose_d3_5 | 1 | 0.0835 | 0.2773 | 0.3 | 0.7633 |
| purpose_d1_6 | 1 | -0.5071 | 0.2407 | -2.11 | 0.0351 |
| purpose_d2_6 | 1 | -0.6653 | 0.3321 | -2 | 0.0452 |
| purpose_d3_6 | 1 | 1.0989 | 0.3847 | 2.86 | 0.0043 |
| pcomtoll_1 | 1 | 0.3702 | 0.0851 | 4.35 | <.0001 |
| pcomtoll_3 | 1 | 18.8432 | 0.0211 | 892.12 | <.0001 |
| pcomtoll_5 | 1 | -16.2426 | 0.044 | -369.38 | <.0001 |
| pwrktoll_1 | 1 | 0.4013 | 0.0857 | 4.68 | <.0001 |
| pwrktoll_3 | 1 | 19.0604 | 0.022 | 868.27 | <.0001 |
| pwrktoll_5 | 1 | -20.2579 | 0.9254 | -21.89 | <.0001 |
| pschtoll_1 | 1 | -0.0447 | 0.1214 | -0.37 | 0.7126 |
| pschtoll_3 | 1 | 19.0842 | 0.0315 | 606.52 | <.0001 |
| pschtoll_5 | 1 | -15.754 | 0.5853 | -26.92 | <.0001 |
| prectoll_1 | 1 | 0.2786 | 0.0846 | 3.29 | 0.001 |
| prectoll_3 | 1 | 18.7918 | 0.0249 | 755.91 | <.0001 |
| prectoll_5 | 1 | -16.5432 | 0.0722 | -229 | <.0001 |
| tollptoll_1 | 1 | -0.0329 | 0.008854 | -3.71 | 0.0002 |
| tollptoll_3 | 1 | -0.0705 | 0.0252 | -2.8 | 0.0052 |
| tollptoll_5 | 1 | -0.2684 | 0.0781 | -3.44 | 0.0006 |

# APPENDIX B

## HOUSTON
## ALTERNATIVE SPECIFIC
## MULTINOMIAL LOGIT MODEL
## PARAMETER ESTIMATES

| Parameter | DF | Estimate | Standard Error | t Value | Approx Pr > \|t\| |
|---|---|---|---|---|---|
| unit_1 | 1 | -0.6636 | 0.2954 | -2.25 | 0.0247 |
| unit_2 | 0 | 2.6034 | . | . | . |
| unit_3 | 1 | -1.6927 | 0.3606 | -4.69 | <.0001 |
| unit_4 | 1 | 3.9648 | 0.4406 | 9 | <.0001 |
| unit_5 | 1 | -1.0959 | 0.47 | -2.33 | 0.0197 |
| unit_6 | 1 | -2.033 | 0.9892 | -2.06 | 0.0399 |
| ttime_1 | 1 | -0.0417 | 0.003399 | -12.26 | <.0001 |
| ttime_2 | 1 | -0.0312 | 0.0012 | -26.04 | <.0001 |
| ttime_3 | 1 | -0.0185 | 0.002743 | -6.76 | <.0001 |
| ttime_4 | 1 | -0.0285 | 0.002045 | -13.96 | <.0001 |
| ttime_5 | 1 | -0.0189 | 0.002142 | -8.84 | <.0001 |
| ttime_6 | 1 | -0.0239 | 0.00287 | -8.34 | <.0001 |
| toll_1 | 1 | -0.1365 | 0.0369 | -3.7 | 0.0002 |
| toll_3 | 1 | -2.2549 | 1.1843 | -1.9 | 0.0569 |
| toll_5 | 1 | -1.6374 | 1.9005 | -0.86 | 0.3889 |
| gender_1 | 1 | 0.3146 | 0.0472 | 6.67 | <.0001 |
| gender_2 | 0 | 0.1624 | . | . | . |
| gender_3 | 1 | 0.2879 | 0.0572 | 5.03 | <.0001 |
| gender_4 | 1 | -0.4847 | 0.0945 | -5.13 | <.0001 |
| gender_5 | 1 | -0.00601 | 0.0752 | -0.08 | 0.9363 |
| gender_6 | 1 | -0.2757 | 0.155 | -1.78 | 0.0754 |
| hhnum_1 | 1 | -0.1288 | 0.0243 | -5.3 | <.0001 |
| hhnum_2 | 0 | -0.2318 | . | . | . |
| hhnum_3 | 1 | 0.1564 | 0.0264 | 5.92 | <.0001 |
| hhnum_4 | 1 | -0.00703 | 0.0434 | -0.16 | 0.8713 |
| hhnum_5 | 1 | 0.2443 | 0.0312 | 7.82 | <.0001 |
| hhnum_6 | 1 | -0.0335 | 0.0549 | -0.61 | 0.5414 |
| motorveh_1 | 1 | 0.0977 | 0.0253 | 3.86 | 0.0001 |
| motorveh_2 | 0 | 0.1724 | . | . | . |
| motorveh_3 | 1 | -0.0495 | 0.0307 | -1.62 | 0.1061 |
| motorveh_4 | 1 | -0.0715 | 0.0495 | -1.44 | 0.1486 |
| motorveh_5 | 1 | -0.154 | 0.0377 | -4.08 | <.0001 |
| motorveh_6 | 1 | 0.006878 | 0.0695 | 0.1 | 0.9211 |
| mlbin_1 | 1 | 1.5331 | 0.0628 | 24.41 | <.0001 |
| mlbin_2 | 0 | -0.4384 | . | . | . |
| mlbin_3 | 1 | 0.4827 | 0.0588 | 8.21 | <.0001 |
| mlbin_4 | 1 | -1.1514 | 0.0822 | -14.01 | <.0001 |
| mlbin_5 | 1 | 0.1304 | 0.0724 | 1.8 | 0.0717 |
| mlbin_6 | 1 | -0.5299 | 0.1353 | -3.92 | <.0001 |
| fampool_1 | 1 | -1.6273 | 0.0939 | -17.34 | <.0001 |
| fampool_2 | 1 | -1.5653 | 0.0787 | -19.89 | <.0001 |
| fampool_3 | 0 | 0.4021 | . | . | . |

| Parameter | DF | Estimate | Standard Error | t Value | Approx Pr > \|t\| |
|---|---|---|---|---|---|
| fampool_4 | 1 | 1.2599 | 0.1039 | 12.13 | <.0001 |
| fampool_5 | 1 | 0.7177 | 0.0885 | 8.11 | <.0001 |
| fampool_6 | 1 | 0.8004 | 0.1623 | 4.93 | <.0001 |
| tollpay_1 | 1 | -0.1415 | 0.061 | -2.32 | 0.0204 |
| tollpay_2 | 0 | -0.1476 | . | . | . |
| tollpay_3 | 1 | -0.1567 | 0.0673 | -2.33 | 0.0198 |
| tollpay_4 | 1 | 0.2375 | 0.0903 | 2.63 | 0.0085 |
| tollpay_5 | 1 | 0.0706 | 0.0732 | 0.96 | 0.3348 |
| tollpay_6 | 1 | 0.1114 | 0.1406 | 0.79 | 0.4285 |
| age_d1_1 | 0 | 0.7639 | . | . | . |
| age_d2_1 | 1 | 0.5775 | 0.1795 | 3.22 | 0.0013 |
| age_d3_1 | 1 | 0.4241 | 0.1829 | 2.32 | 0.0204 |
| age_d4_1 | 1 | 0.4035 | 0.1836 | 2.2 | 0.028 |
| age_d5_1 | 1 | -0.0123 | 0.1867 | -0.07 | 0.9473 |
| age_d1_2 | 1 | 0.0698 | 0.1934 | 0.36 | 0.718 |
| age_d2_2 | 0 | -0.1295 | . | . | . |
| age_d3_2 | 0 | -0.0469 | . | . | . |
| age_d4_2 | 0 | 0.1373 | . | . | . |
| age_d5_2 | 0 | -0.2577 | . | . | . |
| age_d1_3 | 1 | 1.5481 | 0.2968 | 5.22 | <.0001 |
| age_d2_3 | 1 | 0.773 | 0.2585 | 2.99 | 0.0028 |
| age_d3_3 | 1 | 0.9164 | 0.2611 | 3.51 | 0.0004 |
| age_d4_3 | 1 | 1.0924 | 0.2601 | 4.2 | <.0001 |
| age_d5_3 | 1 | 0.5451 | 0.2577 | 2.12 | 0.0344 |
| age_d1_4 | 1 | 0.0118 | 0.346 | 0.03 | 0.9727 |
| age_d2_4 | 1 | -0.6977 | 0.278 | -2.51 | 0.0121 |
| age_d3_4 | 1 | -0.4318 | 0.2815 | -1.53 | 0.125 |
| age_d4_4 | 1 | -0.4562 | 0.2865 | -1.59 | 0.1114 |
| age_d5_4 | 1 | -0.3848 | 0.2809 | -1.37 | 0.1707 |
| age_d1_5 | 1 | -0.8481 | 0.3551 | -2.39 | 0.0169 |
| age_d2_5 | 1 | -1.0465 | 0.3139 | -3.33 | 0.0009 |
| age_d3_5 | 1 | -0.6214 | 0.3165 | -1.96 | 0.0496 |
| age_d4_5 | 1 | -0.9463 | 0.3179 | -2.98 | 0.0029 |
| age_d5_5 | 1 | -0.4921 | 0.3239 | -1.52 | 0.1287 |
| age_d1_6 | 1 | -1.5633 | 0.6678 | -2.34 | 0.0192 |
| age_d2_6 | 1 | 0.5074 | 0.5609 | 0.9 | 0.3657 |
| age_d3_6 | 1 | -0.2557 | 0.5776 | -0.44 | 0.658 |
| age_d4_6 | 1 | -0.2377 | 0.574 | -0.41 | 0.6788 |
| age_d5_6 | 1 | 0.581 | 0.5694 | 1.02 | 0.3075 |
| ethni_d1_1 | 1 | 0.5574 | 0.1079 | 5.17 | <.0001 |
| ethni_d2_1 | 0 | 0.1819 | . | . | . |
| ethni_d3_1 | 1 | 0.4493 | 0.1171 | 3.84 | 0.0001 |
| ethni_d1_2 | 0 | -0.2727 | . | . | . |

| Parameter | DF | Estimate | Standard Error | t Value | Approx Pr > \|t\| |
|---|---|---|---|---|---|
| ethni_d2_2 | 1 | -0.4046 | 0.1201 | -3.37 | 0.0008 |
| ethni_d3_2 | 0 | -0.0117 | . | . | . |
| ethni_d1_3 | 1 | -0.1479 | 0.1104 | -1.34 | 0.1806 |
| ethni_d2_3 | 1 | -0.3195 | 0.152 | -2.1 | 0.0356 |
| ethni_d3_3 | 1 | 0.1134 | 0.1205 | 0.94 | 0.3465 |
| ethni_d1_4 | 1 | -1.382 | 0.129 | -10.72 | <.0001 |
| ethni_d2_4 | 1 | -1.876 | 0.1923 | -9.75 | <.0001 |
| ethni_d3_4 | 1 | -1.9673 | 0.1626 | -12.1 | <.0001 |
| ethni_d1_5 | 1 | -0.4128 | 0.1489 | -2.77 | 0.0056 |
| ethni_d2_5 | 1 | 0.1538 | 0.1825 | 0.84 | 0.3991 |
| ethni_d3_5 | 1 | -0.1651 | 0.1619 | -1.02 | 0.3078 |
| ethni_d1_6 | 1 | 1.6759 | 0.6552 | 2.56 | 0.0105 |
| ethni_d2_6 | 1 | 2.2728 | 0.6691 | 3.4 | 0.0007 |
| ethni_d3_6 | 1 | 1.5994 | 0.6649 | 2.41 | 0.0161 |
| hhtyp_d1_1 | 1 | 0.2163 | 0.1139 | 1.9 | 0.0576 |
| hhtyp_d2_1 | 0 | 0.3801 | . | . | . |
| hhtyp_d3_1 | 1 | 0.0161 | 0.1313 | 0.12 | 0.9026 |
| hhtyp_d4_1 | 1 | 0.2155 | 0.1185 | 1.82 | 0.069 |
| hhtyp_d5_1 | 1 | -0.00059 | 0.1213 | 0 | 0.9961 |
| hhtyp_d1_2 | 0 | 0.1646 | . | . | . |
| hhtyp_d2_2 | 1 | 0.1627 | 0.131 | 1.24 | 0.2141 |
| hhtyp_d3_2 | 0 | 0.024 | . | . | . |
| hhtyp_d4_2 | 0 | 0.224 | . | . | . |
| hhtyp_d5_2 | 0 | -0.0221 | . | . | . |
| hhtyp_d1_3 | 1 | 0.403 | 0.1351 | 2.98 | 0.0029 |
| hhtyp_d2_3 | 1 | 0.4478 | 0.1637 | 2.74 | 0.0062 |
| hhtyp_d3_3 | 1 | 0.6093 | 0.1514 | 4.02 | <.0001 |
| hhtyp_d4_3 | 1 | -0.0176 | 0.1336 | -0.13 | 0.8949 |
| hhtyp_d5_3 | 1 | 0.4725 | 0.1405 | 3.36 | 0.0008 |
| hhtyp_d1_4 | 1 | -0.7404 | 0.2173 | -3.41 | 0.0007 |
| hhtyp_d2_4 | 1 | -1.1667 | 0.2657 | -4.39 | <.0001 |
| hhtyp_d3_4 | 1 | 0.0225 | 0.2286 | 0.1 | 0.9216 |
| hhtyp_d4_4 | 1 | -0.0307 | 0.1906 | -0.16 | 0.8719 |
| hhtyp_d5_4 | 1 | 0.1256 | 0.2008 | 0.63 | 0.5315 |
| hhtyp_d1_5 | 1 | 0.6349 | 0.1922 | 3.3 | 0.001 |
| hhtyp_d2_5 | 1 | 0.4932 | 0.2136 | 2.31 | 0.021 |
| hhtyp_d3_5 | 1 | 1.7199 | 0.2076 | 8.29 | <.0001 |
| hhtyp_d4_5 | 1 | 0.5285 | 0.1833 | 2.88 | 0.0039 |
| hhtyp_d5_5 | 1 | 0.2919 | 0.2051 | 1.42 | 0.1546 |
| hhtyp_d1_6 | 1 | -0.6852 | 0.2623 | -2.61 | 0.009 |
| hhtyp_d2_6 | 1 | -0.3116 | 0.2814 | -1.11 | 0.2682 |
| hhtyp_d3_6 | 1 | -2.4119 | 0.7175 | -3.36 | 0.0008 |
| hhtyp_d4_6 | 1 | -0.9227 | 0.2552 | -3.62 | 0.0003 |

| Parameter | DF | Estimate | Standard Error | t Value | Approx Pr > \|t\| |
|---|---|---|---|---|---|
| hhtyp_d5_6 | 1 | -0.8762 | 0.2846 | -3.08 | 0.0021 |
| occtyp_d1_1 | 1 | 0.34 | 0.1138 | 2.99 | 0.0028 |
| occtyp_d2_1 | 1 | 0.337 | 0.1241 | 2.72 | 0.0066 |
| occtyp_d3_1 | 1 | 0.3752 | 0.157 | 2.39 | 0.0168 |
| occtyp_d4_1 | 1 | 0.2063 | 0.1192 | 1.73 | 0.0834 |
| occtyp_d5_1 | 1 | 1.8859 | 0.2816 | 6.7 | <.0001 |
| occtyp_d6_1 | 1 | 0.3436 | 0.168 | 2.04 | 0.0409 |
| occtyp_d7_1 | 1 | 0.4002 | 0.211 | 1.9 | 0.0579 |
| occtyp_d8_1 | 1 | 0.5797 | 0.1606 | 3.61 | 0.0003 |
| occtyp_d9_1 | 1 | 0.6306 | 0.157 | 4.02 | <.0001 |
| occtyp_d10_1 | 1 | -0.2923 | 0.2852 | -1.02 | 0.3055 |
| occtyp_d11_1 | 1 | 0.8889 | 0.2237 | 3.97 | <.0001 |
| occtyp_d1_2 | 0 | 0.4792 | . | . | . |
| occtyp_d2_2 | 0 | 0.6479 | . | . | . |
| occtyp_d3_2 | 0 | 0.9515 | . | . | . |
| occtyp_d4_2 | 0 | 0.1399 | . | . | . |
| occtyp_d5_2 | 0 | 2.5149 | . | . | . |
| occtyp_d6_2 | 0 | 0.1396 | . | . | . |
| occtyp_d7_2 | 0 | 0.8805 | . | . | . |
| occtyp_d8_2 | 0 | 0.7145 | . | . | . |
| occtyp_d9_2 | 0 | 0.2785 | . | . | . |
| occtyp_d10_2 | 1 | -0.1339 | 0.2255 | -0.59 | 0.5528 |
| occtyp_d11_2 | 0 | 1.433 | . | . | . |
| occtyp_d1_3 | 1 | 0.2902 | 0.1294 | 2.24 | 0.0249 |
| occtyp_d2_3 | 1 | -0.1963 | 0.1472 | -1.33 | 0.1823 |
| occtyp_d3_3 | 1 | 0.1006 | 0.1969 | 0.51 | 0.6092 |
| occtyp_d4_3 | 1 | 0.2778 | 0.1316 | 2.11 | 0.0348 |
| occtyp_d5_3 | 1 | 1.9159 | 0.2476 | 7.74 | <.0001 |
| occtyp_d6_3 | 1 | 0.1718 | 0.1991 | 0.86 | 0.3884 |
| occtyp_d7_3 | 1 | 0.6428 | 0.2182 | 2.95 | 0.0032 |
| occtyp_d8_3 | 1 | 0.7302 | 0.1706 | 4.28 | <.0001 |
| occtyp_d9_3 | 1 | -0.2032 | 0.1923 | -1.06 | 0.2907 |
| occtyp_d10_3 | 0 | 0.8378 | . | . | . |
| occtyp_d11_3 | 1 | 1.3615 | 0.2405 | 5.66 | <.0001 |
| occtyp_d1_4 | 1 | -1.1826 | 0.1695 | -6.98 | <.0001 |
| occtyp_d2_4 | 1 | -1.181 | 0.2019 | -5.85 | <.0001 |
| occtyp_d3_4 | 1 | 0.7297 | 0.2047 | 3.56 | 0.0004 |
| occtyp_d4_4 | 1 | -0.4612 | 0.168 | -2.75 | 0.006 |
| occtyp_d5_4 | 1 | 1.0036 | 0.4586 | 2.19 | 0.0287 |
| occtyp_d6_4 | 1 | -1.0568 | 0.3047 | -3.47 | 0.0005 |
| occtyp_d7_4 | 1 | -1.6773 | 0.3115 | -5.38 | <.0001 |
| occtyp_d8_4 | 1 | -1.274 | 0.2993 | -4.26 | <.0001 |
| occtyp_d9_4 | 1 | -1.5398 | 0.3122 | -4.93 | <.0001 |

| Parameter | DF | Estimate | Standard Error | t Value | Approx Pr > \|t\| |
|---|---|---|---|---|---|
| occtyp_d10_4 | 1 | -1.068 | 0.3834 | -2.79 | 0.0053 |
| occtyp_d11_4 | 1 | -0.5421 | 0.3223 | -1.68 | 0.0926 |
| occtyp_d1_5 | 1 | 0.3486 | 0.1861 | 1.87 | 0.061 |
| occtyp_d2_5 | 1 | 0.5636 | 0.1998 | 2.82 | 0.0048 |
| occtyp_d3_5 | 1 | -1.0113 | 0.4053 | -2.5 | 0.0126 |
| occtyp_d4_5 | 1 | 0.6109 | 0.1876 | 3.26 | 0.0011 |
| occtyp_d5_5 | 1 | -4.0835 | 3.2023 | -1.28 | 0.2022 |
| occtyp_d6_5 | 1 | 1.258 | 0.2191 | 5.74 | <.0001 |
| occtyp_d7_5 | 1 | 1.3582 | 0.2325 | 5.84 | <.0001 |
| occtyp_d8_5 | 1 | 1.1971 | 0.2394 | 5 | <.0001 |
| occtyp_d9_5 | 1 | 0.8854 | 0.2185 | 4.05 | <.0001 |
| occtyp_d10_5 | 1 | 0.3998 | 0.2856 | 1.4 | 0.1616 |
| occtyp_d11_5 | 1 | -2.1302 | 0.737 | -2.89 | 0.0038 |
| occtyp_d1_6 | 1 | -0.2635 | 0.2635 | -1 | 0.3173 |
| occtyp_d2_6 | 1 | -0.1574 | 0.3079 | -0.51 | 0.6092 |
| occtyp_d3_6 | 1 | -1.1233 | 0.66 | -1.7 | 0.0887 |
| occtyp_d4_6 | 1 | -0.7689 | 0.2806 | -2.74 | 0.0061 |
| occtyp_d5_6 | 1 | -3.2334 | 2.6875 | -1.2 | 0.2289 |
| occtyp_d6_6 | 1 | -0.8627 | 0.4645 | -1.86 | 0.0633 |
| occtyp_d7_6 | 1 | -1.5863 | 0.5118 | -3.1 | 0.0019 |
| occtyp_d8_6 | 1 | -1.8735 | 0.4722 | -3.97 | <.0001 |
| occtyp_d9_6 | 1 | -0.0272 | 0.3316 | -0.08 | 0.9346 |
| occtyp_d10_6 | 1 | 0.2466 | 0.3573 | 0.69 | 0.49 |
| occtyp_d11_6 | 1 | -0.9835 | 0.6714 | -1.46 | 0.1429 |
| educ_d1_1 | 1 | -0.3486 | 0.2425 | -1.44 | 0.1505 |
| educ_d2_1 | 0 | -0.3654 | . | . | . |
| educ_d3_1 | 1 | -0.6177 | 0.0734 | -8.41 | <.0001 |
| educ_d4_1 | 1 | -0.248 | 0.0649 | -3.82 | 0.0001 |
| educ_d1_2 | 1 | 0.0876 | 0.2106 | 0.42 | 0.6774 |
| educ_d2_2 | 1 | -0.7248 | 0.0919 | -7.88 | <.0001 |
| educ_d3_2 | 0 | -0.3319 | . | . | . |
| educ_d4_2 | 0 | -0.0219 | . | . | . |
| educ_d1_3 | 1 | -1.7918 | 0.3193 | -5.61 | <.0001 |
| educ_d2_3 | 1 | -0.3268 | 0.1238 | -2.64 | 0.0083 |
| educ_d3_3 | 1 | -0.154 | 0.0946 | -1.63 | 0.1036 |
| educ_d4_3 | 1 | 0.1919 | 0.0863 | 2.22 | 0.0262 |
| educ_d1_4 | 1 | 0.5549 | 0.3159 | 1.76 | 0.079 |
| educ_d2_4 | 1 | -0.0226 | 0.1807 | -0.12 | 0.9007 |
| educ_d3_4 | 1 | -0.3283 | 0.1476 | -2.22 | 0.0261 |
| educ_d4_4 | 1 | -0.2826 | 0.1407 | -2.01 | 0.0445 |
| educ_d1_5 | 0 | 0.4925 | . | . | . |
| educ_d2_5 | 1 | 0.2712 | 0.1495 | 1.81 | 0.0697 |
| educ_d3_5 | 1 | -0.0689 | 0.1227 | -0.56 | 0.5744 |

| Parameter | DF | Estimate | Standard Error | t Value | Approx Pr > \|t\| |
|---|---|---|---|---|---|
| educ_d4_5 | 1 | -0.0349 | 0.1182 | -0.3 | 0.7675 |
| educ_d1_6 | 1 | 1.0241 | 0.5642 | 1.82 | 0.0695 |
| educ_d2_6 | 1 | 1.1583 | 0.333 | 3.48 | 0.0005 |
| educ_d3_6 | 1 | 1.4952 | 0.2917 | 5.13 | <.0001 |
| educ_d4_6 | 1 | 0.3977 | 0.2949 | 1.35 | 0.1775 |
| income_d123_1 | 1 | -0.4227 | 0.1595 | -2.65 | 0.008 |
| income_d4_1 | 1 | 0.5107 | 0.1125 | 4.54 | <.0001 |
| income_d5_1 | 1 | -0.2355 | 0.1035 | -2.27 | 0.0229 |
| income_d6_1 | 1 | -0.1447 | 0.0965 | -1.5 | 0.1337 |
| income_d7_1 | 1 | -0.2252 | 0.102 | -2.21 | 0.0272 |
| income_d8_1 | 1 | -0.2245 | 0.0995 | -2.26 | 0.0241 |
| income_d123_2 | 1 | -0.00693 | 0.1493 | -0.05 | 0.963 |
| income_d4_2 | 0 | 0.8047 | . | . | . |
| income_d5_2 | 0 | 0.3851 | . | . | . |
| income_d6_2 | 0 | -0.0174 | . | . | . |
| income_d7_2 | 0 | -0.1113 | . | . | . |
| income_d8_2 | 0 | -0.0701 | . | . | . |
| income_d123_3 | 0 | 0.587 | . | . | . |
| income_d4_3 | 1 | 0.8449 | 0.1515 | 5.58 | <.0001 |
| income_d5_3 | 1 | 0.127 | 0.1465 | 0.87 | 0.3861 |
| income_d6_3 | 1 | 0.4711 | 0.1363 | 3.46 | 0.0005 |
| income_d7_3 | 1 | 0.4127 | 0.1424 | 2.9 | 0.0038 |
| income_d8_3 | 1 | 0.0969 | 0.145 | 0.67 | 0.504 |
| income_d123_4 | 1 | -0.4339 | 0.2409 | -1.8 | 0.0716 |
| income_d4_4 | 1 | 0.1802 | 0.2098 | 0.86 | 0.3903 |
| income_d5_4 | 1 | -0.1246 | 0.2024 | -0.62 | 0.5381 |
| income_d6_4 | 1 | -0.5527 | 0.2016 | -2.74 | 0.0061 |
| income_d7_4 | 1 | 0.2457 | 0.193 | 1.27 | 0.203 |
| income_d8_4 | 1 | -0.1308 | 0.2058 | -0.64 | 0.525 |
| income_d123_5 | 1 | 0.31 | 0.2188 | 1.42 | 0.1565 |
| income_d4_5 | 1 | -0.0744 | 0.2123 | -0.35 | 0.726 |
| income_d5_5 | 1 | 0.2643 | 0.1903 | 1.39 | 0.1649 |
| income_d6_5 | 1 | 0.366 | 0.1833 | 2 | 0.0458 |
| income_d7_5 | 1 | -0.00091 | 0.1952 | 0 | 0.9963 |
| income_d8_5 | 1 | 0.5411 | 0.1876 | 2.88 | 0.0039 |
| income_d123_6 | 1 | -0.0374 | 0.3964 | -0.09 | 0.9248 |
| income_d4_6 | 1 | -2.2526 | 0.556 | -4.05 | <.0001 |
| income_d5_6 | 1 | -0.4065 | 0.3691 | -1.1 | 0.2707 |
| income_d6_6 | 1 | -0.1225 | 0.3482 | -0.35 | 0.7251 |
| income_d7_6 | 1 | -0.3161 | 0.3663 | -0.86 | 0.3881 |
| income_d8_6 | 1 | -0.2125 | 0.3794 | -0.56 | 0.5754 |
| purpose_d1_1 | 1 | 0.2472 | 0.0845 | 2.93 | 0.0034 |
| purpose_d2_1 | 1 | 0.29 | 0.1072 | 2.7 | 0.0069 |

| Parameter | DF | Estimate | Standard Error | t Value | Approx Pr > \|t\| |
|---|---|---|---|---|---|
| purpose_d3_1 | 1 | 0.5405 | 0.1852 | 2.92 | 0.0035 |
| purpose_d1_2 | 0 | 0.2355 | . | . | . |
| purpose_d2_2 | 0 | 0.3854 | . | . | . |
| purpose_d3_2 | 0 | -0.6432 | . | . | . |
| purpose_d1_3 | 1 | -0.1203 | 0.086 | -1.4 | 0.1618 |
| purpose_d2_3 | 1 | -0.6088 | 0.1136 | -5.36 | <.0001 |
| purpose_d3_3 | 1 | -1.7532 | 0.2066 | -8.49 | <.0001 |
| purpose_d1_4 | 1 | -0.1757 | 0.1207 | -1.46 | 0.1453 |
| purpose_d2_4 | 1 | 0.4241 | 0.149 | 2.85 | 0.0044 |
| purpose_d3_4 | 1 | -0.2347 | 0.2648 | -0.89 | 0.3755 |
| purpose_d1_5 | 1 | -0.0978 | 0.1042 | -0.94 | 0.348 |
| purpose_d2_5 | 1 | 0.1328 | 0.1241 | 1.07 | 0.2845 |
| purpose_d3_5 | 1 | -0.2292 | 0.1894 | -1.21 | 0.2261 |
| purpose_d1_6 | 1 | -0.0949 | 0.1833 | -0.52 | 0.6048 |
| purpose_d2_6 | 1 | -0.6669 | 0.2506 | -2.66 | 0.0078 |
| purpose_d3_6 | 1 | 2.3117 | 0.3179 | 7.27 | <.0001 |
| pcomtoll_1 | 1 | 0.0233 | 0.037 | 0.63 | 0.5285 |
| pcomtoll_3 | 1 | 2.0105 | 1.1845 | 1.7 | 0.0896 |
| pcomtoll_5 | 1 | 1.491 | 1.9004 | 0.78 | 0.4327 |
| pwrktoll_1 | 1 | 0.0493 | 0.0381 | 1.29 | 0.1958 |
| pwrktoll_3 | 1 | 2.2153 | 1.1846 | 1.87 | 0.0615 |
| pwrktoll_5 | 1 | 1.5084 | 1.9007 | 0.79 | 0.4274 |
| pschtoll_1 | 1 | -0.1375 | 0.0529 | -2.6 | 0.0093 |
| pschtoll_3 | 1 | 1.827 | 1.1887 | 1.54 | 0.1243 |
| pschtoll_5 | 1 | -4.439 | 2.1359 | -2.08 | 0.0377 |
| prectoll_1 | 1 | 0.0503 | 0.0364 | 1.38 | 0.1665 |
| prectoll_3 | 1 | 1.9312 | 1.1838 | 1.63 | 0.1028 |
| prectoll_5 | 1 | 0.9796 | 1.9047 | 0.51 | 0.607 |
| tollptoll_1 | 1 | 0.0425 | 0.009661 | 4.4 | <.0001 |
| tollptoll_3 | 1 | 0.0875 | 0.0214 | 4.09 | <.0001 |
| tollptoll_5 | 1 | 0.0271 | 0.0578 | 0.47 | 0.6387 |

# APPENDIX C

# SAS CODE

```
option ls=100;
*mlsov->1
*gpsov->2
*mlho2->3
*gpho2->4
*mlho3->5
*gpho3->6
;
data q_combined;
        infile 'C:\Documents and Settings\kxf7416\Desktop\dataset1.prn';
        input surveyid tollpay location
tt1 tt2 tt3 tt4 tt5 tt6
toll1 toll3 toll5
SPCh SPChMode SPChRoute
age gender ethnicity hhtype hhnum motorveh occtype educ income mlbin fampool
question rnd estim
purpagg pcommute pwork pschool precoth;

*proc print data=q_combined;
run;

%let xt_est_dfw_inc123_ethn1_count=124;
%let xt_est_dfw_inc123_ethn2_count=64;
%let xt_est_dfw_inc123_ethn3_count=34;
%let xt_est_dfw_inc123_ethn456_count=51;
%let xt_est_dfw_inc4_ethn1_count=125;
%let xt_est_dfw_inc4_ethn2_count=54;
%let xt_est_dfw_inc4_ethn3_count=37;
%let xt_est_dfw_inc4_ethn456_count=10;
%let xt_est_dfw_inc5_ethn1_count=417;
%let xt_est_dfw_inc5_ethn2_count=47;
%let xt_est_dfw_inc5_ethn3_count=60;
%let xt_est_dfw_inc5_ethn456_count=37;
%let xt_est_dfw_inc6_ethn1_count=775;
%let xt_est_dfw_inc6_ethn2_count=68;
%let xt_est_dfw_inc6_ethn3_count=48;
%let xt_est_dfw_inc6_ethn456_count=43;
%let xt_est_dfw_inc7_ethn1_count=888;
%let xt_est_dfw_inc7_ethn2_count=61;
%let xt_est_dfw_inc7_ethn3_count=33;
%let xt_est_dfw_inc7_ethn456_count=79;
%let xt_est_dfw_inc8_ethn1_count=1222;
%let xt_est_dfw_inc8_ethn2_count=43;
```

```
%let xt_est_dfw_inc8_ethn3_count=70;
%let xt_est_dfw_inc8_ethn456_count=94;
%let xt_est_dfw_inc910_ethn1_count=789;
%let xt_est_dfw_inc910_ethn2_count=28;
%let xt_est_dfw_inc910_ethn3_count=29;
%let xt_est_dfw_inc910_ethn456_count=66;
/*
%let xt_est_hou_inc123_ethn1_count=173;
%let xt_est_hou_inc123_ethn2_count=95;
%let xt_est_hou_inc123_ethn3_count=145;
%let xt_est_hou_inc123_ethn456_count=32;
%let xt_est_hou_inc4_ethn1_count=206;
%let xt_est_hou_inc4_ethn2_count=50;
%let xt_est_hou_inc4_ethn3_count=100;
%let xt_est_hou_inc4_ethn456_count=23;
%let xt_est_hou_inc5_ethn1_count=542;
%let xt_est_hou_inc5_ethn2_count=94;
%let xt_est_hou_inc5_ethn3_count=110;
%let xt_est_hou_inc5_ethn456_count=33;
%let xt_est_hou_inc6_ethn1_count=1046;
%let xt_est_hou_inc6_ethn2_count=107;
%let xt_est_hou_inc6_ethn3_count=149;
%let xt_est_hou_inc6_ethn456_count=119;
%let xt_est_hou_inc7_ethn1_count=1087;
%let xt_est_hou_inc7_ethn2_count=79;
%let xt_est_hou_inc7_ethn3_count=101;
%let xt_est_hou_inc7_ethn456_count=108;
%let xt_est_hou_inc8_ethn1_count=1408;
%let xt_est_hou_inc8_ethn2_count=77;
%let xt_est_hou_inc8_ethn3_count=111;
%let xt_est_hou_inc8_ethn456_count=125;
%let xt_est_hou_inc910_ethn1_count=919;
%let xt_est_hou_inc910_ethn2_count=32;
%let xt_est_hou_inc910_ethn3_count=53;
%let xt_est_hou_inc910_ethn456_count=60;

*/
%let wt_est_dfw_inc123_ethn1_count=2993;
%let wt_est_dfw_inc123_ethn2_count=1277;
%let wt_est_dfw_inc123_ethn3_count=1166;
%let wt_est_dfw_inc123_ethn456_count=291;
%let wt_est_dfw_inc4_ethn1_count=1855;
%let wt_est_dfw_inc4_ethn2_count=524;
%let wt_est_dfw_inc4_ethn3_count=636;
```

```
%let wt_est_dfw_inc4_ethn456_count=131;
%let wt_est_dfw_inc5_ethn1_count=2663;
%let wt_est_dfw_inc5_ethn2_count=559;
%let wt_est_dfw_inc5_ethn3_count=722;
%let wt_est_dfw_inc5_ethn456_count=200;
%let wt_est_dfw_inc6_ethn1_count=3547;
%let wt_est_dfw_inc6_ethn2_count=576;
%let wt_est_dfw_inc6_ethn3_count=665;
%let wt_est_dfw_inc6_ethn456_count=278;
%let wt_est_dfw_inc7_ethn1_count=2249;
%let wt_est_dfw_inc7_ethn2_count=247;
%let wt_est_dfw_inc7_ethn3_count=266;
%let wt_est_dfw_inc7_ethn456_count=148;
%let wt_est_dfw_inc8_ethn1_count=1956;
%let wt_est_dfw_inc8_ethn2_count=152;
%let wt_est_dfw_inc8_ethn3_count=152;
%let wt_est_dfw_inc8_ethn456_count=132;
%let wt_est_dfw_inc910_ethn1_count=1311;
%let wt_est_dfw_inc910_ethn2_count=59;
%let wt_est_dfw_inc910_ethn3_count=65;
%let wt_est_dfw_inc910_ethn456_count=66;
/*
%let wt_est_hou_inc123_ethn1_count=2013;
%let wt_est_hou_inc123_ethn2_count=1347;
%let wt_est_hou_inc123_ethn3_count=1508;
%let wt_est_hou_inc123_ethn456_count=298;
%let wt_est_hou_inc4_ethn1_count=1137;
%let wt_est_hou_inc4_ethn2_count=484;
%let wt_est_hou_inc4_ethn3_count=723;
%let wt_est_hou_inc4_ethn456_count=136;
%let wt_est_hou_inc5_ethn1_count=1633;
%let wt_est_hou_inc5_ethn2_count=531;
%let wt_est_hou_inc5_ethn3_count=783;
%let wt_est_hou_inc5_ethn456_count=175;
%let wt_est_hou_inc6_ethn1_count=2269;
%let wt_est_hou_inc6_ethn2_count=517;
%let wt_est_hou_inc6_ethn3_count=696;
%let wt_est_hou_inc6_ethn456_count=227;
%let wt_est_hou_inc7_ethn1_count=1515;
%let wt_est_hou_inc7_ethn2_count=245;
%let wt_est_hou_inc7_ethn3_count=271;
%let wt_est_hou_inc7_ethn456_count=139;
%let wt_est_hou_inc8_ethn1_count=1423;
%let wt_est_hou_inc8_ethn2_count=146;
```

```
%let wt_est_hou_inc8_ethn3_count=157;
%let wt_est_hou_inc8_ethn456_count=129;
%let wt_est_hou_inc910_ethn1_count=919;
%let wt_est_hou_inc910_ethn2_count=54;
%let wt_est_hou_inc910_ethn3_count=78;
%let wt_est_hou_inc910_ethn456_count=65;
*/
**************************************;
/*DALLAS-FORT WORTH*/
**************************************;

data xt_est_dfw_inc123_ethn1;
	set q_combined;
	if (location=1 and estim=1 and (income=1 or income=2 or income=3) and
ethnicity=1) then output;
*proc print data=xt_est_dfw_inc123_ethn1;
run;
data rnd_est_dfw_inc123_ethn1;
	set xt_est_dfw_inc123_ethn1;
	rnd=uniform(1)*((&xt_est_dfw_inc123_ethn1_count)-1)+1;
	output;
run;
proc sort data=rnd_est_dfw_inc123_ethn1;
	by rnd;
*proc print data=rnd_est_dfw_inc123_ethn1;
run;
data temp;
set rnd_est_dfw_inc123_ethn1;
if _n_ <= (&wt_est_dfw_inc123_ethn1_count)-
int(&wt_est_dfw_inc123_ethn1_count/&xt_est_dfw_inc123_ethn1_count)*(&xt_est_df
w_inc123_ethn1_count);
*proc print data=temp;
run;
data rp_est_dfw_inc123_ethn1;
	set xt_est_dfw_inc123_ethn1;
	do i=1 to
int(&wt_est_dfw_inc123_ethn1_count/&xt_est_dfw_inc123_ethn1_count);
		output;
	end;
*proc print data=rp_est_dfw_inc123_ethn1;
run;
data wt_est_dfw_inc123_ethn1;
	set rp_est_dfw_inc123_ethn1 temp;
*proc print data=wt_est_dfw_inc123_ethn1;
```

```
run;
****************************************;
data xt_est_dfw_inc123_ethn2;
        set q_combined;
        if (location=1 and estim=1 and (income=1 or income=2 or income=3) and
ethnicity=2) then output;
*proc print data=xt_est_dfw_inc123_ethn2;
run;
data rnd_est_dfw_inc123_ethn2;
        set xt_est_dfw_inc123_ethn2;
        rnd=uniform(2)*((&xt_est_dfw_inc123_ethn2_count)-1)+1;
        output;
run;
proc sort data=rnd_est_dfw_inc123_ethn2;
        by rnd;
*proc print data=rnd_est_dfw_inc123_ethn2;
run;
data temp;
set rnd_est_dfw_inc123_ethn2;
if _n_ <= (&wt_est_dfw_inc123_ethn2_count)-
int(&wt_est_dfw_inc123_ethn2_count/&xt_est_dfw_inc123_ethn2_count)*(&xt_est_df
w_inc123_ethn2_count);
*proc print data=temp;
run;
data rp_est_dfw_inc123_ethn2;
        set xt_est_dfw_inc123_ethn2;
        do i=1 to
int(&wt_est_dfw_inc123_ethn2_count/&xt_est_dfw_inc123_ethn2_count);
                output;
        end;
*proc print data=rp_est_dfw_inc123_ethn2;
run;
data wt_est_dfw_inc123_ethn2;
        set rp_est_dfw_inc123_ethn2 temp;
*proc print data=wt_est_dfw_inc123_ethn2;
run;
****************************************;
data xt_est_dfw_inc123_ethn3;
        set q_combined;
        if (location=1 and estim=1 and (income=1 or income=2 or income=3) and
ethnicity=3) then output;
*proc print data=xt_est_dfw_inc123_ethn3;
run;
data rnd_est_dfw_inc123_ethn3;
```

```
        set xt_est_dfw_inc123_ethn3;
        rnd=uniform(3)*((&xt_est_dfw_inc123_ethn3_count)-1)+1;
        output;
run;
proc sort data=rnd_est_dfw_inc123_ethn3;
        by rnd;
*proc print data=rnd_est_dfw_inc123_ethn3;
run;
data temp;
set rnd_est_dfw_inc123_ethn3;
if _n_ <= (&wt_est_dfw_inc123_ethn3_count)-
int(&wt_est_dfw_inc123_ethn3_count/&xt_est_dfw_inc123_ethn3_count)*(&xt_est_df
w_inc123_ethn3_count);
*proc print data=temp;
run;
data rp_est_dfw_inc123_ethn3;
        set xt_est_dfw_inc123_ethn3;
        do i=1 to
int(&wt_est_dfw_inc123_ethn3_count/&xt_est_dfw_inc123_ethn3_count);
                output;
        end;
*proc print data=rp_est_dfw_inc123_ethn3;
run;
data wt_est_dfw_inc123_ethn3;
        set rp_est_dfw_inc123_ethn3 temp;
*proc print data=wt_est_dfw_inc123_ethn3;
run;
**************************************;
data xt_est_dfw_inc123_ethn456;
        set q_combined;
        if (location=1 and estim=1 and (income=1 or income=2 or income=3) and
(ethnicity=4 or ethnicity=5 or ethnicity=6)) then output;
*proc print data=xt_est_dfw_inc123_ethn456;
run;
data rnd_est_dfw_inc123_ethn456;
        set xt_est_dfw_inc123_ethn456;
        rnd=uniform(4)*((&xt_est_dfw_inc123_ethn456_count)-1)+1;
        output;
run;
proc sort data=rnd_est_dfw_inc123_ethn456;
        by rnd;
*proc print data=rnd_est_dfw_inc123_ethn456;
run;
data temp;
```

```
set rnd_est_dfw_inc123_ethn456;
if _n_ <= (&wt_est_dfw_inc123_ethn456_count)-
int(&wt_est_dfw_inc123_ethn456_count/&xt_est_dfw_inc123_ethn456_count)*(&xt_e
st_dfw_inc123_ethn456_count);
*proc print data=temp;
run;
data rp_est_dfw_inc123_ethn456;
        set xt_est_dfw_inc123_ethn456;
        do i=1 to
int(&wt_est_dfw_inc123_ethn456_count/&xt_est_dfw_inc123_ethn456_count);
                output;
        end;
*proc print data=rp_est_dfw_inc123_ethn456;
run;
data wt_est_dfw_inc123_ethn456;
        set rp_est_dfw_inc123_ethn456 temp;
*proc print data=wt_est_dfw_inc123_ethn456;
run;
**************************************;
data xt_est_dfw_inc8_ethn1;
        set q_combined;
        if (location=1 and estim=1 and income=8 and ethnicity=1) then output;
*proc print data=xt_est_dfw_inc8_ethn1;
run;
data rnd_est_dfw_inc8_ethn1;
        set xt_est_dfw_inc8_ethn1;
        rnd=uniform(5)*((&xt_est_dfw_inc8_ethn1_count)-1)+1;
        output;
run;
proc sort data=rnd_est_dfw_inc8_ethn1;
        by rnd;
*proc print data=rnd_est_dfw_inc8_ethn1;
run;
data temp;
set rnd_est_dfw_inc8_ethn1;
if _n_ <= (&wt_est_dfw_inc8_ethn1_count)-
int(&wt_est_dfw_inc8_ethn1_count/&xt_est_dfw_inc8_ethn1_count)*(&xt_est_dfw_in
c8_ethn1_count);
*proc print data=temp;
run;
data rp_est_dfw_inc8_ethn1;
        set xt_est_dfw_inc8_ethn1;
        do i=1 to int(&wt_est_dfw_inc8_ethn1_count/&xt_est_dfw_inc8_ethn1_count);
                output;
```

```
        end;
*proc print data=rp_est_dfw_inc8_ethn1;
run;
data wt_est_dfw_inc8_ethn1;
        set rp_est_dfw_inc8_ethn1 temp;
*proc print data=wt_est_dfw_inc8_ethn1;
run;
**************************************;
data xt_est_dfw_inc8_ethn2;
        set q_combined;
        if (location=1 and estim=1 and income=8 and ethnicity=2) then output;
*proc print data=xt_est_dfw_inc8_ethn2;
run;
data rnd_est_dfw_inc8_ethn2;
        set xt_est_dfw_inc8_ethn2;
        rnd=uniform(6)*((&xt_est_dfw_inc8_ethn2_count)-1)+1;
        output;
run;
proc sort data=rnd_est_dfw_inc8_ethn2;
        by rnd;
*proc print data=rnd_est_dfw_inc8_ethn2;
run;
data temp;
set rnd_est_dfw_inc8_ethn2;
if _n_ <= (&wt_est_dfw_inc8_ethn2_count)-
int(&wt_est_dfw_inc8_ethn2_count/&xt_est_dfw_inc8_ethn2_count)*(&xt_est_dfw_in
c8_ethn2_count);
*proc print data=temp;
run;
data rp_est_dfw_inc8_ethn2;
        set xt_est_dfw_inc8_ethn2;
        do i=1 to int(&wt_est_dfw_inc8_ethn2_count/&xt_est_dfw_inc8_ethn2_count);
                output;
        end;
*proc print data=rp_est_dfw_inc8_ethn2;
run;
data wt_est_dfw_inc8_ethn2;
        set rp_est_dfw_inc8_ethn2 temp;
*proc print data=wt_est_dfw_inc8_ethn2;
run;
**************************************;
data xt_est_dfw_inc8_ethn3;
        set q_combined;
        if (location=1 and estim=1 and income=8 and ethnicity=3) then output;
```

```
*proc print data=xt_est_dfw_inc8_ethn3;
run;
data rnd_est_dfw_inc8_ethn3;
	set xt_est_dfw_inc8_ethn3;
	rnd=uniform(7)*((&xt_est_dfw_inc8_ethn3_count)-1)+1;
	output;
run;
proc sort data=rnd_est_dfw_inc8_ethn3;
	by rnd;
*proc print data=rnd_est_dfw_inc8_ethn3;
run;
data temp;
set rnd_est_dfw_inc8_ethn3;
if _n_ <= (&wt_est_dfw_inc8_ethn3_count)-
int(&wt_est_dfw_inc8_ethn3_count/&xt_est_dfw_inc8_ethn3_count)*(&xt_est_dfw_in
c8_ethn3_count);
*proc print data=temp;
run;
data rp_est_dfw_inc8_ethn3;
	set xt_est_dfw_inc8_ethn3;
	do i=1 to int(&wt_est_dfw_inc8_ethn3_count/&xt_est_dfw_inc8_ethn3_count);
		output;
	end;
*proc print data=rp_est_dfw_inc8_ethn3;
run;
data wt_est_dfw_inc8_ethn3;
	set rp_est_dfw_inc8_ethn3 temp;
*proc print data=wt_est_dfw_inc8_ethn3;
run;
**************************************;
data xt_est_dfw_inc8_ethn456;
	set q_combined;
	if (location=1 and estim=1 and income=8 and (ethnicity=4 or ethnicity=5 or
ethnicity=6)) then output;
*proc print data=xt_est_dfw_inc8_ethn456;
run;
data rnd_est_dfw_inc8_ethn456;
	set xt_est_dfw_inc8_ethn456;
	rnd=uniform(8)*((&xt_est_dfw_inc8_ethn456_count)-1)+1;
	output;
run;
proc sort data=rnd_est_dfw_inc8_ethn456;
	by rnd;
*proc print data=rnd_est_dfw_inc8_ethn456;
```

```
run;
data temp;
set rnd_est_dfw_inc8_ethn456;
if _n_ <= (&wt_est_dfw_inc8_ethn456_count)-
int(&wt_est_dfw_inc8_ethn456_count/&xt_est_dfw_inc8_ethn456_count)*(&xt_est_df
w_inc8_ethn456_count);
*proc print data=temp;
run;
data rp_est_dfw_inc8_ethn456;
        set xt_est_dfw_inc8_ethn456;
        do i=1 to
int(&wt_est_dfw_inc8_ethn456_count/&xt_est_dfw_inc8_ethn456_count);
                output;
        end;
*proc print data=rp_est_dfw_inc8_ethn456;
run;
data wt_est_dfw_inc8_ethn456;
        set rp_est_dfw_inc8_ethn456 temp;
*proc print data=wt_est_dfw_inc8_ethn456;
run;
**************************************;
data xt_est_dfw_inc4_ethn1;
        set q_combined;
        if (location=1 and estim=1 and income=4 and ethnicity=1) then output;
*proc print data=xt_est_dfw_inc4_ethn1;
run;
data rnd_est_dfw_inc4_ethn1;
        set xt_est_dfw_inc4_ethn1;
        rnd=uniform(9)*((&xt_est_dfw_inc4_ethn1_count)-1)+1;
        output;
run;
proc sort data=rnd_est_dfw_inc4_ethn1;
        by rnd;
*proc print data=rnd_est_dfw_inc4_ethn1;
run;
data temp;
set rnd_est_dfw_inc4_ethn1;
if _n_ <= (&wt_est_dfw_inc4_ethn1_count)-
int(&wt_est_dfw_inc4_ethn1_count/&xt_est_dfw_inc4_ethn1_count)*(&xt_est_dfw_in
c4_ethn1_count);
*proc print data=temp;
run;
data rp_est_dfw_inc4_ethn1;
        set xt_est_dfw_inc4_ethn1;
```

```
	do i=1 to int(&wt_est_dfw_inc4_ethn1_count/&xt_est_dfw_inc4_ethn1_count);
		output;
	end;
*proc print data=rp_est_dfw_inc4_ethn1;
run;
data wt_est_dfw_inc4_ethn1;
	set rp_est_dfw_inc4_ethn1 temp;
*proc print data=wt_est_dfw_inc4_ethn1;
run;
**************************************;
data xt_est_dfw_inc4_ethn2;
	set q_combined;
	if (location=1 and estim=1 and income=4 and ethnicity=2) then output;
*proc print data=xt_est_dfw_inc4_ethn2;
run;
data rnd_est_dfw_inc4_ethn2;
	set xt_est_dfw_inc4_ethn2;
	rnd=uniform(10)*((&xt_est_dfw_inc4_ethn2_count)-1)+1;
	output;
run;
proc sort data=rnd_est_dfw_inc4_ethn2;
	by rnd;
*proc print data=rnd_est_dfw_inc4_ethn2;
run;
data temp;
set rnd_est_dfw_inc4_ethn2;
if _n_ <= (&wt_est_dfw_inc4_ethn2_count)-
int(&wt_est_dfw_inc4_ethn2_count/&xt_est_dfw_inc4_ethn2_count)*(&xt_est_dfw_in
c4_ethn2_count);
*proc print data=temp;
run;
data rp_est_dfw_inc4_ethn2;
	set xt_est_dfw_inc4_ethn2;
	do i=1 to int(&wt_est_dfw_inc4_ethn2_count/&xt_est_dfw_inc4_ethn2_count);
		output;
	end;
*proc print data=rp_est_dfw_inc4_ethn2;
run;
data wt_est_dfw_inc4_ethn2;
	set rp_est_dfw_inc4_ethn2 temp;
*proc print data=wt_est_dfw_inc4_ethn2;
run;
**************************************;
data xt_est_dfw_inc4_ethn3;
```

```
	set q_combined;
	if (location=1 and estim=1 and income=4 and ethnicity=3) then output;
*proc print data=xt_est_dfw_inc4_ethn3;
run;
data rnd_est_dfw_inc4_ethn3;
	set xt_est_dfw_inc4_ethn3;
	rnd=uniform(11)*((&xt_est_dfw_inc4_ethn3_count)-1)+1;
	output;
run;
proc sort data=rnd_est_dfw_inc4_ethn3;
	by rnd;
*proc print data=rnd_est_dfw_inc4_ethn3;
run;
data temp;
set rnd_est_dfw_inc4_ethn3;
if _n_ <= (&wt_est_dfw_inc4_ethn3_count)-
int(&wt_est_dfw_inc4_ethn3_count/&xt_est_dfw_inc4_ethn3_count)*(&xt_est_dfw_in
c4_ethn3_count);
*proc print data=temp;
run;
data rp_est_dfw_inc4_ethn3;
	set xt_est_dfw_inc4_ethn3;
	do i=1 to int(&wt_est_dfw_inc4_ethn3_count/&xt_est_dfw_inc4_ethn3_count);
		output;
	end;
*proc print data=rp_est_dfw_inc4_ethn3;
run;
data wt_est_dfw_inc4_ethn3;
	set rp_est_dfw_inc4_ethn3 temp;
*proc print data=wt_est_dfw_inc4_ethn3;
run;
**************************************;
data xt_est_dfw_inc4_ethn456;
	set q_combined;
	if (location=1 and estim=1 and income=4 and (ethnicity=4 or ethnicity=5 or
ethnicity=6)) then output;
*proc print data=xt_est_dfw_inc4_ethn456;
run;
data rnd_est_dfw_inc4_ethn456;
	set xt_est_dfw_inc4_ethn456;
	rnd=uniform(12)*((&xt_est_dfw_inc4_ethn456_count)-1)+1;
	output;
run;
proc sort data=rnd_est_dfw_inc4_ethn456;
```

```
        by rnd;
*proc print data=rnd_est_dfw_inc4_ethn456;
run;
data temp;
set rnd_est_dfw_inc4_ethn456;
if _n_ <= (&wt_est_dfw_inc4_ethn456_count)-
int(&wt_est_dfw_inc4_ethn456_count/&xt_est_dfw_inc4_ethn456_count)*(&xt_est_df
w_inc4_ethn456_count);
*proc print data=temp;
run;
data rp_est_dfw_inc4_ethn456;
        set xt_est_dfw_inc4_ethn456;
        do i=1 to
int(&wt_est_dfw_inc4_ethn456_count/&xt_est_dfw_inc4_ethn456_count);
                output;
        end;
*proc print data=rp_est_dfw_inc4_ethn456;
run;
data wt_est_dfw_inc4_ethn456;
        set rp_est_dfw_inc4_ethn456 temp;
*proc print data=wt_est_dfw_inc4_ethn456;
run;
**************************************;
data xt_est_dfw_inc5_ethn1;
        set q_combined;
        if (location=1 and estim=1 and income=5 and ethnicity=1) then output;
*proc print data=xt_est_dfw_inc5_ethn1;
run;
data rnd_est_dfw_inc5_ethn1;
        set xt_est_dfw_inc5_ethn1;
        rnd=uniform(13)*((&xt_est_dfw_inc5_ethn1_count)-1)+1;
        output;
run;
proc sort data=rnd_est_dfw_inc5_ethn1;
        by rnd;
*proc print data=rnd_est_dfw_inc5_ethn1;
run;
data temp;
set rnd_est_dfw_inc5_ethn1;
if _n_ <= (&wt_est_dfw_inc5_ethn1_count)-
int(&wt_est_dfw_inc5_ethn1_count/&xt_est_dfw_inc5_ethn1_count)*(&xt_est_dfw_in
c5_ethn1_count);
*proc print data=temp;
run;
```

```
data rp_est_dfw_inc5_ethn1;
	set xt_est_dfw_inc5_ethn1;
	do i=1 to int(&wt_est_dfw_inc5_ethn1_count/&xt_est_dfw_inc5_ethn1_count);
		output;
	end;
*proc print data=rp_est_dfw_inc5_ethn1;
run;
data wt_est_dfw_inc5_ethn1;
	set rp_est_dfw_inc5_ethn1 temp;
*proc print data=wt_est_dfw_inc5_ethn1;
run;
**************************************;
data xt_est_dfw_inc5_ethn2;
	set q_combined;
	if (location=1 and estim=1 and income=5 and ethnicity=2) then output;
*proc print data=xt_est_dfw_inc5_ethn2;
run;
data rnd_est_dfw_inc5_ethn2;
	set xt_est_dfw_inc5_ethn2;
	rnd=uniform(14)*((&xt_est_dfw_inc5_ethn2_count)-1)+1;
	output;
run;
proc sort data=rnd_est_dfw_inc5_ethn2;
	by rnd;
*proc print data=rnd_est_dfw_inc5_ethn2;
run;
data temp;
set rnd_est_dfw_inc5_ethn2;
if _n_ <= (&wt_est_dfw_inc5_ethn2_count)-
int(&wt_est_dfw_inc5_ethn2_count/&xt_est_dfw_inc5_ethn2_count)*(&xt_est_dfw_in
c5_ethn2_count);
*proc print data=temp;
run;
data rp_est_dfw_inc5_ethn2;
	set xt_est_dfw_inc5_ethn2;
	do i=1 to int(&wt_est_dfw_inc5_ethn2_count/&xt_est_dfw_inc5_ethn2_count);
		output;
	end;
*proc print data=rp_est_dfw_inc5_ethn2;
run;
data wt_est_dfw_inc5_ethn2;
	set rp_est_dfw_inc5_ethn2 temp;
*proc print data=wt_est_dfw_inc5_ethn2;
run;
```

```
**************************************;
data xt_est_dfw_inc5_ethn3;
        set q_combined;
        if (location=1 and estim=1 and income=5 and ethnicity=3) then output;
*proc print data=xt_est_dfw_inc5_ethn3;
run;
data rnd_est_dfw_inc5_ethn3;
        set xt_est_dfw_inc5_ethn3;
        rnd=uniform(15)*((&xt_est_dfw_inc5_ethn3_count)-1)+1;
        output;
run;
proc sort data=rnd_est_dfw_inc5_ethn3;
        by rnd;
*proc print data=rnd_est_dfw_inc5_ethn3;
run;
data temp;
set rnd_est_dfw_inc5_ethn1;
if _n_ <= (&wt_est_dfw_inc5_ethn3_count)-
int(&wt_est_dfw_inc5_ethn3_count/&xt_est_dfw_inc5_ethn3_count)*(&xt_est_dfw_in
c5_ethn3_count);
*proc print data=temp;
run;
data rp_est_dfw_inc5_ethn3;
        set xt_est_dfw_inc5_ethn3;
        do i=1 to int(&wt_est_dfw_inc5_ethn3_count/&xt_est_dfw_inc5_ethn3_count);
                output;
        end;
*proc print data=rp_est_dfw_inc5_ethn3;
run;
data wt_est_dfw_inc5_ethn3;
        set rp_est_dfw_inc5_ethn3 temp;
*proc print data=wt_est_dfw_inc5_ethn3;
run;
**************************************;
data xt_est_dfw_inc5_ethn456;
        set q_combined;
        if (location=1 and estim=1 and income=5 and (ethnicity=4 or ethnicity=5 or
ethnicity=6)) then output;
*proc print data=xt_est_dfw_inc5_ethn456;
run;
data rnd_est_dfw_inc5_ethn456;
        set xt_est_dfw_inc5_ethn456;
        rnd=uniform(16)*((&xt_est_dfw_inc5_ethn456_count)-1)+1;
        output;
```

```
run;
proc sort data=rnd_est_dfw_inc5_ethn456;
	by rnd;
*proc print data=rnd_est_dfw_inc5_ethn456;
run;
data temp;
set rnd_est_dfw_inc5_ethn1;
if _n_ <= (&wt_est_dfw_inc5_ethn456_count)-
int(&wt_est_dfw_inc5_ethn456_count/&xt_est_dfw_inc5_ethn456_count)*(&xt_est_df
w_inc5_ethn456_count);
*proc print data=temp;
run;
data rp_est_dfw_inc5_ethn456;
	set xt_est_dfw_inc5_ethn456;
	do i=1 to
int(&wt_est_dfw_inc5_ethn456_count/&xt_est_dfw_inc5_ethn456_count);
		output;
	end;
*proc print data=rp_est_dfw_inc5_ethn456;
run;
data wt_est_dfw_inc5_ethn456;
	set rp_est_dfw_inc5_ethn456 temp;
*proc print data=wt_est_dfw_inc5_ethn456;
run;
****************************************;
data xt_est_dfw_inc6_ethn1;
	set q_combined;
	if (location=1 and estim=1 and income=6 and ethnicity=1) then output;
*proc print data=xt_est_dfw_inc6_ethn1;
run;
data rnd_est_dfw_inc6_ethn1;
	set xt_est_dfw_inc6_ethn1;
	rnd=uniform(17)*((&xt_est_dfw_inc6_ethn1_count)-1)+1;
	output;
run;
proc sort data=rnd_est_dfw_inc6_ethn1;
	by rnd;
*proc print data=rnd_est_dfw_inc6_ethn1;
run;
data temp;
set rnd_est_dfw_inc6_ethn1;
if _n_ <= (&wt_est_dfw_inc6_ethn1_count)-
int(&wt_est_dfw_inc6_ethn1_count/&xt_est_dfw_inc6_ethn1_count)*(&xt_est_dfw_in
c6_ethn1_count);
```

```
*proc print data=temp;
run;
data rp_est_dfw_inc6_ethn1;
	set xt_est_dfw_inc6_ethn1;
	do i=1 to int(&wt_est_dfw_inc6_ethn1_count/&xt_est_dfw_inc6_ethn1_count);
		output;
	end;
*proc print data=rp_est_dfw_inc6_ethn1;
run;
data wt_est_dfw_inc6_ethn1;
	set rp_est_dfw_inc6_ethn1 temp;
*proc print data=wt_est_dfw_inc6_ethn1;
run;
**************************************;
data xt_est_dfw_inc6_ethn2;
	set q_combined;
	if (location=1 and estim=1 and income=6 and ethnicity=2) then output;
*proc print data=xt_est_dfw_inc6_ethn2;
run;
data rnd_est_dfw_inc6_ethn2;
	set xt_est_dfw_inc6_ethn2;
	rnd=uniform(18)*((&xt_est_dfw_inc6_ethn2_count)-1)+1;
	output;
run;
proc sort data=rnd_est_dfw_inc6_ethn2;
	by rnd;
*proc print data=rnd_est_dfw_inc6_ethn2;
run;
data temp;
set rnd_est_dfw_inc6_ethn2;
if _n_ <= (&wt_est_dfw_inc6_ethn2_count)-
int(&wt_est_dfw_inc6_ethn2_count/&xt_est_dfw_inc6_ethn2_count)*(&xt_est_dfw_in
c6_ethn2_count);
*proc print data=temp;
run;
data rp_est_dfw_inc6_ethn2;
	set xt_est_dfw_inc6_ethn2;
	do i=1 to int(&wt_est_dfw_inc6_ethn2_count/&xt_est_dfw_inc6_ethn2_count);
		output;
	end;
*proc print data=rp_est_dfw_inc6_ethn2;
run;
data wt_est_dfw_inc6_ethn2;
	set rp_est_dfw_inc6_ethn2 temp;
```

```
*proc print data=wt_est_dfw_inc6_ethn2;
run;
**************************************;
data xt_est_dfw_inc6_ethn3;
        set q_combined;
        if (location=1 and estim=1 and income=6 and ethnicity=3) then output;
*proc print data=xt_est_dfw_inc6_ethn3;
run;
data rnd_est_dfw_inc6_ethn3;
        set xt_est_dfw_inc6_ethn3;
        rnd=uniform(19)*((&xt_est_dfw_inc6_ethn3_count)-1)+1;
        output;
run;
proc sort data=rnd_est_dfw_inc6_ethn3;
        by rnd;
*proc print data=rnd_est_dfw_inc6_ethn3;
run;
data temp;
set rnd_est_dfw_inc6_ethn3;
if _n_ <= (&wt_est_dfw_inc6_ethn3_count)-
int(&wt_est_dfw_inc6_ethn3_count/&xt_est_dfw_inc6_ethn3_count)*(&xt_est_dfw_in
c6_ethn3_count);
*proc print data=temp;
run;
data rp_est_dfw_inc6_ethn3;
        set xt_est_dfw_inc6_ethn3;
        do i=1 to int(&wt_est_dfw_inc6_ethn3_count/&xt_est_dfw_inc6_ethn3_count);
                output;
        end;
*proc print data=rp_est_dfw_inc6_ethn3;
run;
data wt_est_dfw_inc6_ethn3;
        set rp_est_dfw_inc6_ethn3 temp;
*proc print data=wt_est_dfw_inc6_ethn3;
run;
**************************************;
data xt_est_dfw_inc6_ethn456;
        set q_combined;
        if (location=1 and estim=1 and income=6 and (ethnicity=4 or ethnicity=5 or
ethnicity=6)) then output;
*proc print data=xt_est_dfw_inc6_ethn456;
run;
data rnd_est_dfw_inc6_ethn456;
        set xt_est_dfw_inc6_ethn456;
```

```
        rnd=uniform(20)*((&xt_est_dfw_inc6_ethn456_count)-1)+1;
        output;
run;
proc sort data=rnd_est_dfw_inc6_ethn456;
        by rnd;
*proc print data=rnd_est_dfw_inc6_ethn456;
run;
data temp;
set rnd_est_dfw_inc6_ethn456;
if _n_ <= (&wt_est_dfw_inc6_ethn456_count)-
int(&wt_est_dfw_inc6_ethn456_count/&xt_est_dfw_inc6_ethn456_count)*(&xt_est_df
w_inc6_ethn456_count);
*proc print data=temp;
run;
data rp_est_dfw_inc6_ethn456;
        set xt_est_dfw_inc6_ethn456;
        do i=1 to
int(&wt_est_dfw_inc6_ethn456_count/&xt_est_dfw_inc6_ethn456_count);
                output;
        end;
*proc print data=rp_est_dfw_inc6_ethn456;
run;
data wt_est_dfw_inc6_ethn456;
        set rp_est_dfw_inc6_ethn456 temp;
*proc print data=wt_est_dfw_inc6_ethn456;
run;
****************************************;
data xt_est_dfw_inc7_ethn1;
        set q_combined;
        if (location=1 and estim=1 and income=7 and ethnicity=1) then output;
*proc print data=xt_est_dfw_inc7_ethn1;
run;
data rnd_est_dfw_inc7_ethn1;
        set xt_est_dfw_inc7_ethn1;
        rnd=uniform(21)*((&xt_est_dfw_inc7_ethn1_count)-1)+1;
        output;
run;
proc sort data=rnd_est_dfw_inc7_ethn1;
        by rnd;
*proc print data=rnd_est_dfw_inc7_ethn1;
run;
data temp;
set rnd_est_dfw_inc7_ethn1;
```

```
if _n_ <= (&wt_est_dfw_inc7_ethn1_count)-
int(&wt_est_dfw_inc7_ethn1_count/&xt_est_dfw_inc7_ethn1_count)*(&xt_est_dfw_in
c7_ethn1_count);
*proc print data=temp;
run;
data rp_est_dfw_inc7_ethn1;
        set xt_est_dfw_inc7_ethn1;
        do i=1 to int(&wt_est_dfw_inc7_ethn1_count/&xt_est_dfw_inc7_ethn1_count);
                output;
        end;
*proc print data=rp_est_dfw_inc7_ethn1;
run;
data wt_est_dfw_inc7_ethn1;
        set rp_est_dfw_inc7_ethn1 temp;
*proc print data=wt_est_dfw_inc7_ethn1;
run;
**************************************;
data xt_est_dfw_inc7_ethn2;
        set q_combined;
        if (location=1 and estim=1 and income=7 and ethnicity=2) then output;
*proc print data=xt_est_dfw_inc7_ethn2;
run;
data rnd_est_dfw_inc7_ethn2;
        set xt_est_dfw_inc7_ethn2;
        rnd=uniform(22)*((&xt_est_dfw_inc7_ethn2_count)-1)+1;
        output;
run;
proc sort data=rnd_est_dfw_inc7_ethn2;
        by rnd;
*proc print data=rnd_est_dfw_inc7_ethn2;
run;
data temp;
set rnd_est_dfw_inc7_ethn2;
if _n_ <= (&wt_est_dfw_inc7_ethn2_count)-
int(&wt_est_dfw_inc7_ethn2_count/&xt_est_dfw_inc7_ethn2_count)*(&xt_est_dfw_in
c7_ethn2_count);
*proc print data=temp;
run;
data rp_est_dfw_inc7_ethn2;
        set xt_est_dfw_inc7_ethn2;
        do i=1 to int(&wt_est_dfw_inc7_ethn2_count/&xt_est_dfw_inc7_ethn2_count);
                output;
        end;
*proc print data=rp_est_dfw_inc7_ethn2;
```

```
run;
data wt_est_dfw_inc7_ethn2;
	set rp_est_dfw_inc7_ethn2 temp;
*proc print data=wt_est_dfw_inc7_ethn2;
run;
****************************************;
data xt_est_dfw_inc7_ethn3;
	set q_combined;
	if (location=1 and estim=1 and income=7 and ethnicity=3) then output;
*proc print data=xt_est_dfw_inc7_ethn3;
run;
data rnd_est_dfw_inc7_ethn3;
	set xt_est_dfw_inc7_ethn3;
	rnd=uniform(23)*((&xt_est_dfw_inc7_ethn3_count)-1)+1;
	output;
run;
proc sort data=rnd_est_dfw_inc7_ethn3;
	by rnd;
*proc print data=rnd_est_dfw_inc7_ethn3;
run;
data temp;
set rnd_est_dfw_inc7_ethn3;
if _n_ <= (&wt_est_dfw_inc7_ethn3_count)-
int(&wt_est_dfw_inc7_ethn3_count/&xt_est_dfw_inc7_ethn3_count)*(&xt_est_dfw_in
c7_ethn3_count);
*proc print data=temp;
run;
data rp_est_dfw_inc7_ethn3;
	set xt_est_dfw_inc7_ethn3;
	do i=1 to int(&wt_est_dfw_inc7_ethn3_count/&xt_est_dfw_inc7_ethn3_count);
		output;
	end;
*proc print data=rp_est_dfw_inc7_ethn3;
run;
data wt_est_dfw_inc7_ethn3;
	set rp_est_dfw_inc7_ethn3 temp;
*proc print data=wt_est_dfw_inc7_ethn3;
run;
****************************************;
data xt_est_dfw_inc7_ethn456;
	set q_combined;
	if (location=1 and estim=1 and income=7 and (ethnicity=4 or ethnicity=5 or
ethnicity=6)) then output;
*proc print data=xt_est_dfw_inc7_ethn456;
```

```
run;
data rnd_est_dfw_inc7_ethn456;
	set xt_est_dfw_inc7_ethn456;
	rnd=uniform(24)*((&xt_est_dfw_inc7_ethn456_count)-1)+1;
	output;
run;
proc sort data=rnd_est_dfw_inc7_ethn456;
	by rnd;
*proc print data=rnd_est_dfw_inc7_ethn456;
run;
data temp;
set rnd_est_dfw_inc7_ethn456;
if _n_ <= (&wt_est_dfw_inc7_ethn456_count)-
int(&wt_est_dfw_inc7_ethn456_count/&xt_est_dfw_inc7_ethn456_count)*(&xt_est_df
w_inc7_ethn456_count);
*proc print data=temp;
run;
data rp_est_dfw_inc7_ethn456;
	set xt_est_dfw_inc7_ethn456;
	do i=1 to
int(&wt_est_dfw_inc7_ethn456_count/&xt_est_dfw_inc7_ethn456_count);
		output;
	end;
*proc print data=rp_est_dfw_inc7_ethn456;
run;
data wt_est_dfw_inc7_ethn456;
	set rp_est_dfw_inc7_ethn456 temp;
*proc print data=wt_est_dfw_inc7_ethn456;
run;
**************************************;
data xt_est_dfw_inc910_ethn1;
	set q_combined;
	if (location=1 and estim=1 and (income=9 or income=10) and ethnicity=1) then
output;
*proc print data=xt_est_dfw_inc910_ethn1;
run;
data rnd_est_dfw_inc910_ethn1;
	set xt_est_dfw_inc910_ethn1;
	rnd=uniform(25)*((&xt_est_dfw_inc910_ethn1_count)-1)+1;
	output;
run;
proc sort data=rnd_est_dfw_inc910_ethn1;
	by rnd;
*proc print data=rnd_est_dfw_inc910_ethn1;
```

```
run;
data temp;
set rnd_est_dfw_inc910_ethn1;
if _n_ <= (&wt_est_dfw_inc910_ethn1_count)-
int(&wt_est_dfw_inc910_ethn1_count/&xt_est_dfw_inc910_ethn1_count)*(&xt_est_df
w_inc910_ethn1_count);
*proc print data=temp;
run;
data rp_est_dfw_inc910_ethn1;
        set xt_est_dfw_inc910_ethn1;
        do i=1 to
int(&wt_est_dfw_inc910_ethn1_count/&xt_est_dfw_inc910_ethn1_count);
                output;
        end;
*proc print data=rp_est_dfw_inc910_ethn1;
run;
data wt_est_dfw_inc910_ethn1;
        set rp_est_dfw_inc910_ethn1 temp;
*proc print data=wt_est_dfw_inc910_ethn1;
run;
**************************************;
data xt_est_dfw_inc910_ethn2;
        set q_combined;
        if (location=1 and estim=1 and (income=9 or income=10) and ethnicity=2) then
output;
*proc print data=xt_est_dfw_inc910_ethn2;
run;
data rnd_est_dfw_inc910_ethn2;
        set xt_est_dfw_inc910_ethn2;
        rnd=uniform(26)*((&xt_est_dfw_inc910_ethn2_count)-1)+1;
        output;
run;
proc sort data=rnd_est_dfw_inc910_ethn2;
        by rnd;
*proc print data=rnd_est_dfw_inc910_ethn2;
run;
data temp;
set rnd_est_dfw_inc910_ethn2;
if _n_ <= (&wt_est_dfw_inc910_ethn2_count)-
int(&wt_est_dfw_inc910_ethn2_count/&xt_est_dfw_inc910_ethn2_count)*(&xt_est_df
w_inc910_ethn2_count);
*proc print data=temp;
run;
data rp_est_dfw_inc910_ethn2;
```

```
	set xt_est_dfw_inc910_ethn2;
	do i=1 to
int(&wt_est_dfw_inc910_ethn2_count/&xt_est_dfw_inc910_ethn2_count);
		output;
	end;
*proc print data=rp_est_dfw_inc910_ethn2;
run;
data wt_est_dfw_inc910_ethn2;
	set rp_est_dfw_inc910_ethn2 temp;
*proc print data=wt_est_dfw_inc910_ethn2;
run;
**************************************;
data xt_est_dfw_inc910_ethn3;
	set q_combined;
	if (location=1 and estim=1 and (income=9 or income=10) and ethnicity=3) then
output;
*proc print data=xt_est_dfw_inc910_ethn3;
run;
data rnd_est_dfw_inc910_ethn3;
	set xt_est_dfw_inc910_ethn3;
	rnd=uniform(27)*((&xt_est_dfw_inc910_ethn3_count)-1)+1;
	output;
run;
proc sort data=rnd_est_dfw_inc910_ethn3;
	by rnd;
*proc print data=rnd_est_dfw_inc910_ethn3;
run;
data temp;
set rnd_est_dfw_inc910_ethn3;
if _n_ <= (&wt_est_dfw_inc910_ethn3_count)-
int(&wt_est_dfw_inc910_ethn3_count/&xt_est_dfw_inc910_ethn3_count)*(&xt_est_df
w_inc910_ethn3_count);
*proc print data=temp;
run;
data rp_est_dfw_inc910_ethn3;
	set xt_est_dfw_inc910_ethn3;
	do i=1 to
int(&wt_est_dfw_inc910_ethn3_count/&xt_est_dfw_inc910_ethn3_count);
		output;
	end;
*proc print data=rp_est_dfw_inc910_ethn3;
run;
data wt_est_dfw_inc910_ethn3;
	set rp_est_dfw_inc910_ethn3 temp;
```

```
*proc print data=wt_est_dfw_inc910_ethn3;
run;
**************************************;
data xt_est_dfw_inc910_ethn456;
        set q_combined;
        if (location=1 and estim=1 and (income=9 or income=10) and (ethnicity=4 or
ethnicity=5 or ethnicity=6)) then output;
*proc print data=xt_est_dfw_inc910_ethn456;
run;
data rnd_est_dfw_inc910_ethn456;
        set xt_est_dfw_inc910_ethn456;
        rnd=uniform(28)*((&xt_est_dfw_inc910_ethn456_count)-1)+1;
        output;
run;
proc sort data=rnd_est_dfw_inc910_ethn456;
        by rnd;
*proc print data=rnd_est_dfw_inc910_ethn456;
run;
data temp;
set rnd_est_dfw_inc910_ethn456;
if _n_ <= (&wt_est_dfw_inc910_ethn456_count)-
int(&wt_est_dfw_inc910_ethn456_count/&xt_est_dfw_inc910_ethn456_count)*(&xt_e
st_dfw_inc910_ethn456_count);
*proc print data=temp;
run;
data rp_est_dfw_inc910_ethn456;
        set xt_est_dfw_inc910_ethn456;
        do i=1 to
int(&wt_est_dfw_inc910_ethn456_count/&xt_est_dfw_inc910_ethn456_count);
                output;
        end;
*proc print data=rp_est_dfw_inc910_ethn456;
run;
data wt_est_dfw_inc910_ethn456;
        set rp_est_dfw_inc910_ethn456 temp;
*proc print data=wt_est_dfw_inc910_ethn456;
run;
data wt_est_dfw;
set
wt_est_dfw_inc123_ethn1
wt_est_dfw_inc123_ethn2
wt_est_dfw_inc123_ethn3
wt_est_dfw_inc123_ethn456
wt_est_dfw_inc8_ethn1
```

```
wt_est_dfw_inc8_ethn2
wt_est_dfw_inc8_ethn3
wt_est_dfw_inc8_ethn456
wt_est_dfw_inc4_ethn1
wt_est_dfw_inc4_ethn2
wt_est_dfw_inc4_ethn3
wt_est_dfw_inc4_ethn456
wt_est_dfw_inc5_ethn1
wt_est_dfw_inc5_ethn2
wt_est_dfw_inc5_ethn3
wt_est_dfw_inc5_ethn456
wt_est_dfw_inc6_ethn1
wt_est_dfw_inc6_ethn2
wt_est_dfw_inc6_ethn3
wt_est_dfw_inc6_ethn456
wt_est_dfw_inc7_ethn1
wt_est_dfw_inc7_ethn2
wt_est_dfw_inc7_ethn3
wt_est_dfw_inc7_ethn456
wt_est_dfw_inc910_ethn1
wt_est_dfw_inc910_ethn2
wt_est_dfw_inc910_ethn3
wt_est_dfw_inc910_ethn456;
*proc print data=wt_est_dfw;
run;

****************************************;
/*HOUSTON*/
****************************************;
/*
data xt_est_hou_inc123_ethn1;
        set q_combined;
        if (location=2 and estim=1 and (income=1 or income=2 or income=3) and
ethnicity=1) then output;
*proc print data=xt_est_hou_inc123_ethn1;
run;
data rnd_est_hou_inc123_ethn1;
        set xt_est_hou_inc123_ethn1;
        rnd=uniform(29)*((&xt_est_hou_inc123_ethn1_count)-1)+1;
        output;
run;
proc sort data=rnd_est_hou_inc123_ethn1;
        by rnd;
*proc print data=rnd_est_hou_inc123_ethn1;
```

```
run;
data temp;
set rnd_est_hou_inc123_ethn1;
if _n_ <= (&wt_est_hou_inc123_ethn1_count)-
int(&wt_est_hou_inc123_ethn1_count/&xt_est_hou_inc123_ethn1_count)*(&xt_est_ho
u_inc123_ethn1_count);
*proc print data=temp;
run;
data rp_est_hou_inc123_ethn1;
        set xt_est_hou_inc123_ethn1;
        do i=1 to
int(&wt_est_hou_inc123_ethn1_count/&xt_est_hou_inc123_ethn1_count);
                output;
        end;
*proc print data=rp_est_hou_inc123_ethn1;
run;
data wt_est_hou_inc123_ethn1;
        set rp_est_hou_inc123_ethn1 temp;
*proc print data=wt_est_hou_inc123_ethn1;
run;
**************************************;
data xt_est_hou_inc123_ethn2;
        set q_combined;
        if (location=2 and estim=1 and (income=1 or income=2 or income=3) and
ethnicity=2) then output;
*proc print data=xt_est_hou_inc123_ethn2;
run;
data rnd_est_hou_inc123_ethn2;
        set xt_est_hou_inc123_ethn2;
        rnd=uniform(30)*((&xt_est_hou_inc123_ethn2_count)-1)+1;
        output;
run;
proc sort data=rnd_est_hou_inc123_ethn2;
        by rnd;
*proc print data=rnd_est_hou_inc123_ethn2;
run;
data temp;
set rnd_est_hou_inc123_ethn2;
if _n_ <= (&wt_est_hou_inc123_ethn2_count)-
int(&wt_est_hou_inc123_ethn2_count/&xt_est_hou_inc123_ethn2_count)*(&xt_est_ho
u_inc123_ethn2_count);
*proc print data=temp;
run;
data rp_est_hou_inc123_ethn2;
```

```
	set xt_est_hou_inc123_ethn2;
	do i=1 to
int(&wt_est_hou_inc123_ethn2_count/&xt_est_hou_inc123_ethn2_count);
		output;
	end;
*proc print data=rp_est_hou_inc123_ethn2;
run;
data wt_est_hou_inc123_ethn2;
	set rp_est_hou_inc123_ethn2 temp;
*proc print data=wt_est_hou_inc123_ethn2;
run;
**************************************;
data xt_est_hou_inc123_ethn3;
	set q_combined;
	if (location=2 and estim=1 and (income=1 or income=2 or income=3) and
ethnicity=3) then output;
*proc print data=xt_est_hou_inc123_ethn3;
run;
data rnd_est_hou_inc123_ethn3;
	set xt_est_hou_inc123_ethn3;
	rnd=uniform(31)*((&xt_est_hou_inc123_ethn3_count)-1)+1;
	output;
run;
proc sort data=rnd_est_hou_inc123_ethn3;
	by rnd;
*proc print data=rnd_est_hou_inc123_ethn3;
run;
data temp;
set rnd_est_hou_inc123_ethn3;
if _n_ <= (&wt_est_hou_inc123_ethn3_count)-
int(&wt_est_hou_inc123_ethn3_count/&xt_est_hou_inc123_ethn3_count)*(&xt_est_ho
u_inc123_ethn3_count);
*proc print data=temp;
run;
data rp_est_hou_inc123_ethn3;
	set xt_est_hou_inc123_ethn3;
	do i=1 to
int(&wt_est_hou_inc123_ethn3_count/&xt_est_hou_inc123_ethn3_count);
		output;
	end;
*proc print data=rp_est_hou_inc123_ethn3;
run;
data wt_est_hou_inc123_ethn3;
	set rp_est_hou_inc123_ethn3 temp;
```

```
*proc print data=wt_est_hou_inc123_ethn3;
run;
**************************************;
data xt_est_hou_inc123_ethn456;
        set q_combined;
        if (location=2 and estim=1 and (income=1 or income=2 or income=3) and
(ethnicity=4 or ethnicity=5 or ethnicity=6)) then output;
*proc print data=xt_est_hou_inc123_ethn456;
run;
data rnd_est_hou_inc123_ethn456;
        set xt_est_hou_inc123_ethn456;
        rnd=uniform(32)*((&xt_est_hou_inc123_ethn456_count)-1)+1;
        output;
run;
proc sort data=rnd_est_hou_inc123_ethn456;
        by rnd;
*proc print data=rnd_est_hou_inc123_ethn456;
run;
data temp;
set rnd_est_hou_inc123_ethn456;
if _n_ <= (&wt_est_hou_inc123_ethn456_count)-
int(&wt_est_hou_inc123_ethn456_count/&xt_est_hou_inc123_ethn456_count)*(&xt_e
st_hou_inc123_ethn456_count);
*proc print data=temp;
run;
data rp_est_hou_inc123_ethn456;
        set xt_est_hou_inc123_ethn456;
        do i=1 to
int(&wt_est_hou_inc123_ethn456_count/&xt_est_hou_inc123_ethn456_count);
                output;
        end;
*proc print data=rp_est_hou_inc123_ethn456;
run;
data wt_est_hou_inc123_ethn456;
        set rp_est_hou_inc123_ethn456 temp;
*proc print data=wt_est_hou_inc123_ethn456;
run;
**************************************;
data xt_est_hou_inc8_ethn1;
        set q_combined;
        if (location=2 and estim=1 and income=8 and ethnicity=1) then output;
*proc print data=xt_est_hou_inc8_ethn1;
run;
data rnd_est_hou_inc8_ethn1;
```

```
	set xt_est_hou_inc8_ethn1;
	rnd=uniform(33)*((&xt_est_hou_inc8_ethn1_count)-1)+1;
	output;
run;
proc sort data=rnd_est_hou_inc8_ethn1;
	by rnd;
*proc print data=rnd_est_hou_inc8_ethn1;
run;
data temp;
set rnd_est_hou_inc8_ethn1;
if _n_ <= (&wt_est_hou_inc8_ethn1_count)-
int(&wt_est_hou_inc8_ethn1_count/&xt_est_hou_inc8_ethn1_count)*(&xt_est_hou_in
c8_ethn1_count);
*proc print data=temp;
run;
data rp_est_hou_inc8_ethn1;
	set xt_est_hou_inc8_ethn1;
	do i=1 to int(&wt_est_hou_inc8_ethn1_count/&xt_est_hou_inc8_ethn1_count);
		output;
	end;
*proc print data=rp_est_hou_inc8_ethn1;
run;
data wt_est_hou_inc8_ethn1;
	set rp_est_hou_inc8_ethn1 temp;
*proc print data=wt_est_hou_inc8_ethn1;
run;
****************************************;
data xt_est_hou_inc8_ethn2;
	set q_combined;
	if (location=2 and estim=1 and income=8 and ethnicity=2) then output;
*proc print data=xt_est_hou_inc8_ethn2;
run;
data rnd_est_hou_inc8_ethn2;
	set xt_est_hou_inc8_ethn2;
	rnd=uniform(34)*((&xt_est_hou_inc8_ethn2_count)-1)+1;
	output;
run;
proc sort data=rnd_est_hou_inc8_ethn2;
	by rnd;
*proc print data=rnd_est_hou_inc8_ethn2;
run;
data temp;
set rnd_est_hou_inc8_ethn2;
```

```
if _n_ <= (&wt_est_hou_inc8_ethn2_count)-
int(&wt_est_hou_inc8_ethn2_count/&xt_est_hou_inc8_ethn2_count)*(&xt_est_hou_in
c8_ethn2_count);
*proc print data=temp;
run;
data rp_est_hou_inc8_ethn2;
        set xt_est_hou_inc8_ethn2;
        do i=1 to int(&wt_est_hou_inc8_ethn2_count/&xt_est_hou_inc8_ethn2_count);
                output;
        end;
*proc print data=rp_est_hou_inc8_ethn2;
run;
data wt_est_hou_inc8_ethn2;
        set rp_est_hou_inc8_ethn2 temp;
*proc print data=wt_est_hou_inc8_ethn2;
run;
**************************************;
data xt_est_hou_inc8_ethn3;
        set q_combined;
        if (location=2 and estim=1 and income=8 and ethnicity=3) then output;
*proc print data=xt_est_hou_inc8_ethn3;
run;
data rnd_est_hou_inc8_ethn3;
        set xt_est_hou_inc8_ethn3;
        rnd=uniform(35)*((&xt_est_hou_inc8_ethn3_count)-1)+1;
        output;
run;
proc sort data=rnd_est_hou_inc8_ethn3;
        by rnd;
*proc print data=rnd_est_hou_inc8_ethn3;
run;
data temp;
set rnd_est_hou_inc8_ethn3;
if _n_ <= (&wt_est_hou_inc8_ethn3_count)-
int(&wt_est_hou_inc8_ethn3_count/&xt_est_hou_inc8_ethn3_count)*(&xt_est_hou_in
c8_ethn3_count);
*proc print data=temp;
run;
data rp_est_hou_inc8_ethn3;
        set xt_est_hou_inc8_ethn3;
        do i=1 to int(&wt_est_hou_inc8_ethn3_count/&xt_est_hou_inc8_ethn3_count);
                output;
        end;
*proc print data=rp_est_hou_inc8_ethn3;
```

```
run;
data wt_est_hou_inc8_ethn3;
	set rp_est_hou_inc8_ethn3 temp;
*proc print data=wt_est_hou_inc8_ethn3;
run;
****************************************;
data xt_est_hou_inc8_ethn456;
	set q_combined;
	if (location=2 and estim=1 and income=8 and (ethnicity=4 or ethnicity=5 or
ethnicity=6)) then output;
*proc print data=xt_est_hou_inc8_ethn456;
run;
data rnd_est_hou_inc8_ethn456;
	set xt_est_hou_inc8_ethn456;
	rnd=uniform(36)*((&xt_est_hou_inc8_ethn456_count)-1)+1;
	output;
run;
proc sort data=rnd_est_hou_inc8_ethn456;
	by rnd;
*proc print data=rnd_est_hou_inc8_ethn456;
run;
data temp;
set rnd_est_hou_inc8_ethn456;
if _n_ <= (&wt_est_hou_inc8_ethn456_count)-
int(&wt_est_hou_inc8_ethn456_count/&xt_est_hou_inc8_ethn456_count)*(&xt_est_ho
u_inc8_ethn456_count);
*proc print data=temp;
run;
data rp_est_hou_inc8_ethn456;
	set xt_est_hou_inc8_ethn456;
	do i=1 to
int(&wt_est_hou_inc8_ethn456_count/&xt_est_hou_inc8_ethn456_count);
		output;
	end;
*proc print data=rp_est_hou_inc8_ethn456;
run;
data wt_est_hou_inc8_ethn456;
	set rp_est_hou_inc8_ethn456 temp;
*proc print data=wt_est_hou_inc8_ethn456;
run;
****************************************;
data xt_est_hou_inc4_ethn1;
	set q_combined;
	if (location=2 and estim=1 and income=4 and ethnicity=1) then output;
```

```
*proc print data=xt_est_hou_inc4_ethn1;
run;
data rnd_est_hou_inc4_ethn1;
	set xt_est_hou_inc4_ethn1;
	rnd=uniform(37)*((&xt_est_hou_inc4_ethn1_count)-1)+1;
	output;
run;
proc sort data=rnd_est_hou_inc4_ethn1;
	by rnd;
*proc print data=rnd_est_hou_inc4_ethn1;
run;
data temp;
set rnd_est_hou_inc4_ethn1;
if _n_ <= (&wt_est_hou_inc4_ethn1_count)-
int(&wt_est_hou_inc4_ethn1_count/&xt_est_hou_inc4_ethn1_count)*(&xt_est_hou_in
c4_ethn1_count);
*proc print data=temp;
run;
data rp_est_hou_inc4_ethn1;
	set xt_est_hou_inc4_ethn1;
	do i=1 to int(&wt_est_hou_inc4_ethn1_count/&xt_est_hou_inc4_ethn1_count);
		output;
	end;
*proc print data=rp_est_hou_inc4_ethn1;
run;
data wt_est_hou_inc4_ethn1;
	set rp_est_hou_inc4_ethn1 temp;
*proc print data=wt_est_hou_inc4_ethn1;
run;
**************************************;
data xt_est_hou_inc4_ethn2;
	set q_combined;
	if (location=2 and estim=1 and income=4 and ethnicity=2) then output;
*proc print data=xt_est_hou_inc4_ethn2;
run;
data rnd_est_hou_inc4_ethn2;
	set xt_est_hou_inc4_ethn2;
	rnd=uniform(38)*((&xt_est_hou_inc4_ethn2_count)-1)+1;
	output;
run;
proc sort data=rnd_est_hou_inc4_ethn2;
	by rnd;
*proc print data=rnd_est_hou_inc4_ethn2;
run;
```

```
data temp;
set rnd_est_hou_inc4_ethn2;
if _n_ <= (&wt_est_hou_inc4_ethn2_count)-
int(&wt_est_hou_inc4_ethn2_count/&xt_est_hou_inc4_ethn2_count)*(&xt_est_hou_in
c4_ethn2_count);
*proc print data=temp;
run;
data rp_est_hou_inc4_ethn2;
	set xt_est_hou_inc4_ethn2;
	do i=1 to int(&wt_est_hou_inc4_ethn2_count/&xt_est_hou_inc4_ethn2_count);
		output;
	end;
*proc print data=rp_est_hou_inc4_ethn2;
run;
data wt_est_hou_inc4_ethn2;
	set rp_est_hou_inc4_ethn2 temp;
*proc print data=wt_est_hou_inc4_ethn2;
run;
**************************************;
data xt_est_hou_inc4_ethn3;
	set q_combined;
	if (location=2 and estim=1 and income=4 and ethnicity=3) then output;
*proc print data=xt_est_hou_inc4_ethn3;
run;
data rnd_est_hou_inc4_ethn3;
	set xt_est_hou_inc4_ethn3;
	rnd=uniform(39)*((&xt_est_hou_inc4_ethn3_count)-1)+1;
	output;
run;
proc sort data=rnd_est_hou_inc4_ethn3;
	by rnd;
*proc print data=rnd_est_hou_inc4_ethn3;
run;
data temp;
set rnd_est_hou_inc4_ethn3;
if _n_ <= (&wt_est_hou_inc4_ethn3_count)-
int(&wt_est_hou_inc4_ethn3_count/&xt_est_hou_inc4_ethn3_count)*(&xt_est_hou_in
c4_ethn3_count);
*proc print data=temp;
run;
data rp_est_hou_inc4_ethn3;
	set xt_est_hou_inc4_ethn3;
	do i=1 to int(&wt_est_hou_inc4_ethn3_count/&xt_est_hou_inc4_ethn3_count);
		output;
```

```
        end;
*proc print data=rp_est_hou_inc4_ethn3;
run;
data wt_est_hou_inc4_ethn3;
        set rp_est_hou_inc4_ethn3 temp;
*proc print data=wt_est_hou_inc4_ethn3;
run;
**************************************;
data xt_est_hou_inc4_ethn456;
        set q_combined;
        if (location=2 and estim=1 and income=4 and (ethnicity=4 or ethnicity=5 or
ethnicity=6)) then output;
*proc print data=xt_est_hou_inc4_ethn456;
run;
data rnd_est_hou_inc4_ethn456;
        set xt_est_hou_inc4_ethn456;
        rnd=uniform(40)*((&xt_est_hou_inc4_ethn456_count)-1)+1;
        output;
run;
proc sort data=rnd_est_hou_inc4_ethn456;
        by rnd;
*proc print data=rnd_est_hou_inc4_ethn456;
run;
data temp;
set rnd_est_hou_inc4_ethn456;
if _n_ <= (&wt_est_hou_inc4_ethn456_count)-
int(&wt_est_hou_inc4_ethn456_count/&xt_est_hou_inc4_ethn456_count)*(&xt_est_ho
u_inc4_ethn456_count);
*proc print data=temp;
run;
data rp_est_hou_inc4_ethn456;
        set xt_est_hou_inc4_ethn456;
        do i=1 to
int(&wt_est_hou_inc4_ethn456_count/&xt_est_hou_inc4_ethn456_count);
                output;
        end;
*proc print data=rp_est_hou_inc4_ethn456;
run;
data wt_est_hou_inc4_ethn456;
        set rp_est_hou_inc4_ethn456 temp;
*proc print data=wt_est_hou_inc4_ethn456;
run;
**************************************;
data xt_est_hou_inc5_ethn1;
```

```
        set q_combined;
        if (location=2 and estim=1 and income=5 and ethnicity=1) then output;
*proc print data=xt_est_hou_inc5_ethn1;
run;
data rnd_est_hou_inc5_ethn1;
        set xt_est_hou_inc5_ethn1;
        rnd=uniform(41)*((&xt_est_hou_inc5_ethn1_count)-1)+1;
        output;
run;
proc sort data=rnd_est_hou_inc5_ethn1;
        by rnd;
*proc print data=rnd_est_hou_inc5_ethn1;
run;
data temp;
set rnd_est_hou_inc5_ethn1;
if _n_ <= (&wt_est_hou_inc5_ethn1_count)-
int(&wt_est_hou_inc5_ethn1_count/&xt_est_hou_inc5_ethn1_count)*(&xt_est_hou_in
c5_ethn1_count);
*proc print data=temp;
run;
data rp_est_hou_inc5_ethn1;
        set xt_est_hou_inc5_ethn1;
        do i=1 to int(&wt_est_hou_inc5_ethn1_count/&xt_est_hou_inc5_ethn1_count);
                output;
        end;
*proc print data=rp_est_hou_inc5_ethn1;
run;
data wt_est_hou_inc5_ethn1;
        set rp_est_hou_inc5_ethn1 temp;
*proc print data=wt_est_hou_inc5_ethn1;
run;
**************************************;
data xt_est_hou_inc5_ethn2;
        set q_combined;
        if (location=2 and estim=1 and income=5 and ethnicity=2) then output;
*proc print data=xt_est_hou_inc5_ethn2;
run;
data rnd_est_hou_inc5_ethn2;
        set xt_est_hou_inc5_ethn2;
        rnd=uniform(42)*((&xt_est_hou_inc5_ethn2_count)-1)+1;
        output;
run;
proc sort data=rnd_est_hou_inc5_ethn2;
        by rnd;
```

```
*proc print data=rnd_est_hou_inc5_ethn2;
run;
data temp;
set rnd_est_hou_inc5_ethn2;
if _n_ <= (&wt_est_hou_inc5_ethn2_count)-
int(&wt_est_hou_inc5_ethn2_count/&xt_est_hou_inc5_ethn2_count)*(&xt_est_hou_in
c5_ethn2_count);
*proc print data=temp;
run;
data rp_est_hou_inc5_ethn2;
        set xt_est_hou_inc5_ethn2;
        do i=1 to int(&wt_est_hou_inc5_ethn2_count/&xt_est_hou_inc5_ethn2_count);
                output;
        end;
*proc print data=rp_est_hou_inc5_ethn2;
run;
data wt_est_hou_inc5_ethn2;
        set rp_est_hou_inc5_ethn2 temp;
*proc print data=wt_est_hou_inc5_ethn2;
run;
**************************************;
data xt_est_hou_inc5_ethn3;
        set q_combined;
        if (location=2 and estim=1 and income=5 and ethnicity=3) then output;
*proc print data=xt_est_hou_inc5_ethn3;
run;
data rnd_est_hou_inc5_ethn3;
        set xt_est_hou_inc5_ethn3;
        rnd=uniform(43)*((&xt_est_hou_inc5_ethn3_count)-1)+1;
        output;
run;
proc sort data=rnd_est_hou_inc5_ethn3;
        by rnd;
*proc print data=rnd_est_hou_inc5_ethn3;
run;
data temp;
set rnd_est_hou_inc5_ethn1;
if _n_ <= (&wt_est_hou_inc5_ethn3_count)-
int(&wt_est_hou_inc5_ethn3_count/&xt_est_hou_inc5_ethn3_count)*(&xt_est_hou_in
c5_ethn3_count);
*proc print data=temp;
run;
data rp_est_hou_inc5_ethn3;
        set xt_est_hou_inc5_ethn3;
```

```
	do i=1 to int(&wt_est_hou_inc5_ethn3_count/&xt_est_hou_inc5_ethn3_count);
		output;
	end;
*proc print data=rp_est_hou_inc5_ethn3;
run;
data wt_est_hou_inc5_ethn3;
	set rp_est_hou_inc5_ethn3 temp;
*proc print data=wt_est_hou_inc5_ethn3;
run;
**************************************;
data xt_est_hou_inc5_ethn456;
	set q_combined;
	if (location=2 and estim=1 and income=5 and (ethnicity=4 or ethnicity=5 or
ethnicity=6)) then output;
*proc print data=xt_est_hou_inc5_ethn456;
run;
data rnd_est_hou_inc5_ethn456;
	set xt_est_hou_inc5_ethn456;
	rnd=uniform(44)*((&xt_est_hou_inc5_ethn456_count)-1)+1;
	output;
run;
proc sort data=rnd_est_hou_inc5_ethn456;
	by rnd;
*proc print data=rnd_est_hou_inc5_ethn456;
run;
data temp;
set rnd_est_hou_inc5_ethn1;
if _n_ <= (&wt_est_hou_inc5_ethn456_count)-
int(&wt_est_hou_inc5_ethn456_count/&xt_est_hou_inc5_ethn456_count)*(&xt_est_ho
u_inc5_ethn456_count);
*proc print data=temp;
run;
data rp_est_hou_inc5_ethn456;
	set xt_est_hou_inc5_ethn456;
	do i=1 to
int(&wt_est_hou_inc5_ethn456_count/&xt_est_hou_inc5_ethn456_count);
		output;
	end;
*proc print data=rp_est_hou_inc5_ethn456;
run;
data wt_est_hou_inc5_ethn456;
	set rp_est_hou_inc5_ethn456 temp;
*proc print data=wt_est_hou_inc5_ethn456;
run;
```

```
**************************************;
data xt_est_hou_inc6_ethn1;
        set q_combined;
        if (location=2 and estim=1 and income=6 and ethnicity=1) then output;
*proc print data=xt_est_hou_inc6_ethn1;
run;
data rnd_est_hou_inc6_ethn1;
        set xt_est_hou_inc6_ethn1;
        rnd=uniform(45)*((&xt_est_hou_inc6_ethn1_count)-1)+1;
        output;
run;
proc sort data=rnd_est_hou_inc6_ethn1;
        by rnd;
*proc print data=rnd_est_hou_inc6_ethn1;
run;
data temp;
set rnd_est_hou_inc6_ethn1;
if _n_ <= (&wt_est_hou_inc6_ethn1_count)-
int(&wt_est_hou_inc6_ethn1_count/&xt_est_hou_inc6_ethn1_count)*(&xt_est_hou_in
c6_ethn1_count);
*proc print data=temp;
run;
data rp_est_hou_inc6_ethn1;
        set xt_est_hou_inc6_ethn1;
        do i=1 to int(&wt_est_hou_inc6_ethn1_count/&xt_est_hou_inc6_ethn1_count);
                output;
        end;
*proc print data=rp_est_hou_inc6_ethn1;
run;
data wt_est_hou_inc6_ethn1;
        set rp_est_hou_inc6_ethn1 temp;
*proc print data=wt_est_hou_inc6_ethn1;
run;
**************************************;
data xt_est_hou_inc6_ethn2;
        set q_combined;
        if (location=2 and estim=1 and income=6 and ethnicity=2) then output;
*proc print data=xt_est_hou_inc6_ethn2;
run;
data rnd_est_hou_inc6_ethn2;
        set xt_est_hou_inc6_ethn2;
        rnd=uniform(46)*((&xt_est_hou_inc6_ethn2_count)-1)+1;
        output;
run;
```

```
proc sort data=rnd_est_hou_inc6_ethn2;
	by rnd;
*proc print data=rnd_est_hou_inc6_ethn2;
run;
data temp;
set rnd_est_hou_inc6_ethn2;
if _n_ <= (&wt_est_hou_inc6_ethn2_count)-
int(&wt_est_hou_inc6_ethn2_count/&xt_est_hou_inc6_ethn2_count)*(&xt_est_hou_in
c6_ethn2_count);
*proc print data=temp;
run;
data rp_est_hou_inc6_ethn2;
	set xt_est_hou_inc6_ethn2;
	do i=1 to int(&wt_est_hou_inc6_ethn2_count/&xt_est_hou_inc6_ethn2_count);
		output;
	end;
*proc print data=rp_est_hou_inc6_ethn2;
run;
data wt_est_hou_inc6_ethn2;
	set rp_est_hou_inc6_ethn2 temp;
*proc print data=wt_est_hou_inc6_ethn2;
run;
**************************************;
data xt_est_hou_inc6_ethn3;
	set q_combined;
	if (location=2 and estim=1 and income=6 and ethnicity=3) then output;
*proc print data=xt_est_hou_inc6_ethn3;
run;
data rnd_est_hou_inc6_ethn3;
	set xt_est_hou_inc6_ethn3;
	rnd=uniform(47)*((&xt_est_hou_inc6_ethn3_count)-1)+1;
	output;
run;
proc sort data=rnd_est_hou_inc6_ethn3;
	by rnd;
*proc print data=rnd_est_hou_inc6_ethn3;
run;
data temp;
set rnd_est_hou_inc6_ethn3;
if _n_ <= (&wt_est_hou_inc6_ethn3_count)-
int(&wt_est_hou_inc6_ethn3_count/&xt_est_hou_inc6_ethn3_count)*(&xt_est_hou_in
c6_ethn3_count);
*proc print data=temp;
run;
```

```
data rp_est_hou_inc6_ethn3;
	set xt_est_hou_inc6_ethn3;
	do i=1 to int(&wt_est_hou_inc6_ethn3_count/&xt_est_hou_inc6_ethn3_count);
		output;
	end;
*proc print data=rp_est_hou_inc6_ethn3;
run;
data wt_est_hou_inc6_ethn3;
	set rp_est_hou_inc6_ethn3 temp;
*proc print data=wt_est_hou_inc6_ethn3;
run;
**************************************;
data xt_est_hou_inc6_ethn456;
	set q_combined;
	if (location=2 and estim=1 and income=6 and (ethnicity=4 or ethnicity=5 or
ethnicity=6)) then output;
*proc print data=xt_est_hou_inc6_ethn456;
run;
data rnd_est_hou_inc6_ethn456;
	set xt_est_hou_inc6_ethn456;
	rnd=uniform(48)*((&xt_est_hou_inc6_ethn456_count)-1)+1;
	output;
run;
proc sort data=rnd_est_hou_inc6_ethn456;
	by rnd;
*proc print data=rnd_est_hou_inc6_ethn456;
run;
data temp;
set rnd_est_hou_inc6_ethn456;
if _n_ <= (&wt_est_hou_inc6_ethn456_count)-
int(&wt_est_hou_inc6_ethn456_count/&xt_est_hou_inc6_ethn456_count)*(&xt_est_ho
u_inc6_ethn456_count);
*proc print data=temp;
run;
data rp_est_hou_inc6_ethn456;
	set xt_est_hou_inc6_ethn456;
	do i=1 to
int(&wt_est_hou_inc6_ethn456_count/&xt_est_hou_inc6_ethn456_count);
		output;
	end;
*proc print data=rp_est_hou_inc6_ethn456;
run;
data wt_est_hou_inc6_ethn456;
	set rp_est_hou_inc6_ethn456 temp;
```

```
*proc print data=wt_est_hou_inc6_ethn456;
run;
*************************************;
data xt_est_hou_inc7_ethn1;
        set q_combined;
        if (location=2 and estim=1 and income=7 and ethnicity=1) then output;
*proc print data=xt_est_hou_inc7_ethn1;
run;
data rnd_est_hou_inc7_ethn1;
        set xt_est_hou_inc7_ethn1;
        rnd=uniform(49)*((&xt_est_hou_inc7_ethn1_count)-1)+1;
        output;
run;
proc sort data=rnd_est_hou_inc7_ethn1;
        by rnd;
*proc print data=rnd_est_hou_inc7_ethn1;
run;
data temp;
set rnd_est_hou_inc7_ethn1;
if _n_ <= (&wt_est_hou_inc7_ethn1_count)-
int(&wt_est_hou_inc7_ethn1_count/&xt_est_hou_inc7_ethn1_count)*(&xt_est_hou_in
c7_ethn1_count);
*proc print data=temp;
run;
data rp_est_hou_inc7_ethn1;
        set xt_est_hou_inc7_ethn1;
        do i=1 to int(&wt_est_hou_inc7_ethn1_count/&xt_est_hou_inc7_ethn1_count);
                output;
        end;
*proc print data=rp_est_hou_inc7_ethn1;
run;
data wt_est_hou_inc7_ethn1;
        set rp_est_hou_inc7_ethn1 temp;
*proc print data=wt_est_hou_inc7_ethn1;
run;
*************************************;
data xt_est_hou_inc7_ethn2;
        set q_combined;
        if (location=2 and estim=1 and income=7 and ethnicity=2) then output;
*proc print data=xt_est_hou_inc7_ethn2;
run;
data rnd_est_hou_inc7_ethn2;
        set xt_est_hou_inc7_ethn2;
        rnd=uniform(50)*((&xt_est_hou_inc7_ethn2_count)-1)+1;
```

```
        output;
run;
proc sort data=rnd_est_hou_inc7_ethn2;
        by rnd;
*proc print data=rnd_est_hou_inc7_ethn2;
run;
data temp;
set rnd_est_hou_inc7_ethn2;
if _n_ <= (&wt_est_hou_inc7_ethn2_count)-
int(&wt_est_hou_inc7_ethn2_count/&xt_est_hou_inc7_ethn2_count)*(&xt_est_hou_in
c7_ethn2_count);
*proc print data=temp;
run;
data rp_est_hou_inc7_ethn2;
        set xt_est_hou_inc7_ethn2;
        do i=1 to int(&wt_est_hou_inc7_ethn2_count/&xt_est_hou_inc7_ethn2_count);
                output;
        end;
*proc print data=rp_est_hou_inc7_ethn2;
run;
data wt_est_hou_inc7_ethn2;
        set rp_est_hou_inc7_ethn2 temp;
*proc print data=wt_est_hou_inc7_ethn2;
run;
****************************************;
data xt_est_hou_inc7_ethn3;
        set q_combined;
        if (location=2 and estim=1 and income=7 and ethnicity=3) then output;
*proc print data=xt_est_hou_inc7_ethn3;
run;
data rnd_est_hou_inc7_ethn3;
        set xt_est_hou_inc7_ethn3;
        rnd=uniform(51)*((&xt_est_hou_inc7_ethn3_count)-1)+1;
        output;
run;
proc sort data=rnd_est_hou_inc7_ethn3;
        by rnd;
*proc print data=rnd_est_hou_inc7_ethn3;
run;
data temp;
set rnd_est_hou_inc7_ethn3;
if _n_ <= (&wt_est_hou_inc7_ethn3_count)-
int(&wt_est_hou_inc7_ethn3_count/&xt_est_hou_inc7_ethn3_count)*(&xt_est_hou_in
c7_ethn3_count);
```

```
*proc print data=temp;
run;
data rp_est_hou_inc7_ethn3;
	set xt_est_hou_inc7_ethn3;
	do i=1 to int(&wt_est_hou_inc7_ethn3_count/&xt_est_hou_inc7_ethn3_count);
		output;
	end;
*proc print data=rp_est_hou_inc7_ethn3;
run;
data wt_est_hou_inc7_ethn3;
	set rp_est_hou_inc7_ethn3 temp;
*proc print data=wt_est_hou_inc7_ethn3;
run;
**************************************;
data xt_est_hou_inc7_ethn456;
	set q_combined;
	if (location=2 and estim=1 and income=7 and (ethnicity=4 or ethnicity=5 or
ethnicity=6)) then output;
*proc print data=xt_est_hou_inc7_ethn456;
run;
data rnd_est_hou_inc7_ethn456;
	set xt_est_hou_inc7_ethn456;
	rnd=uniform(52)*((&xt_est_hou_inc7_ethn456_count)-1)+1;
	output;
run;
proc sort data=rnd_est_hou_inc7_ethn456;
	by rnd;
*proc print data=rnd_est_hou_inc7_ethn456;
run;
data temp;
set rnd_est_hou_inc7_ethn456;
if _n_ <= (&wt_est_hou_inc7_ethn456_count)-
int(&wt_est_hou_inc7_ethn456_count/&xt_est_hou_inc7_ethn456_count)*(&xt_est_ho
u_inc7_ethn456_count);
*proc print data=temp;
run;
data rp_est_hou_inc7_ethn456;
	set xt_est_hou_inc7_ethn456;
	do i=1 to
int(&wt_est_hou_inc7_ethn456_count/&xt_est_hou_inc7_ethn456_count);
		output;
	end;
*proc print data=rp_est_hou_inc7_ethn456;
run;
```

```
data wt_est_hou_inc7_ethn456;
	set rp_est_hou_inc7_ethn456 temp;
*proc print data=wt_est_hou_inc7_ethn456;
run;
**************************************;
data xt_est_hou_inc910_ethn1;
	set q_combined;
	if (location=2 and estim=1 and (income=9 or income=10) and ethnicity=1) then
output;
*proc print data=xt_est_hou_inc910_ethn1;
run;
data rnd_est_hou_inc910_ethn1;
	set xt_est_hou_inc910_ethn1;
	rnd=uniform(53)*((&xt_est_hou_inc910_ethn1_count)-1)+1;
	output;
run;
proc sort data=rnd_est_hou_inc910_ethn1;
	by rnd;
*proc print data=rnd_est_hou_inc910_ethn1;
run;
data temp;
set rnd_est_hou_inc910_ethn1;
if _n_ <= (&wt_est_hou_inc910_ethn1_count)-
int(&wt_est_hou_inc910_ethn1_count/&xt_est_hou_inc910_ethn1_count)*(&xt_est_ho
u_inc910_ethn1_count);
*proc print data=temp;
run;
data rp_est_hou_inc910_ethn1;
	set xt_est_hou_inc910_ethn1;
	do i=1 to
int(&wt_est_hou_inc910_ethn1_count/&xt_est_hou_inc910_ethn1_count);
		output;
	end;
*proc print data=rp_est_hou_inc910_ethn1;
run;
data wt_est_hou_inc910_ethn1;
	set rp_est_hou_inc910_ethn1 temp;
*proc print data=wt_est_hou_inc910_ethn1;
run;
**************************************;
data xt_est_hou_inc910_ethn2;
	set q_combined;
	if (location=2 and estim=1 and (income=9 or income=10) and ethnicity=2) then
output;
```

```
*proc print data=xt_est_hou_inc910_ethn2;
run;
data rnd_est_hou_inc910_ethn2;
	set xt_est_hou_inc910_ethn2;
	rnd=uniform(54)*((&xt_est_hou_inc910_ethn2_count)-1)+1;
	output;
run;
proc sort data=rnd_est_hou_inc910_ethn2;
	by rnd;
*proc print data=rnd_est_hou_inc910_ethn2;
run;
data temp;
set rnd_est_hou_inc910_ethn2;
if _n_ <= (&wt_est_hou_inc910_ethn2_count)-
int(&wt_est_hou_inc910_ethn2_count/&xt_est_hou_inc910_ethn2_count)*(&xt_est_ho
u_inc910_ethn2_count);
*proc print data=temp;
run;
data rp_est_hou_inc910_ethn2;
	set xt_est_hou_inc910_ethn2;
	do i=1 to
int(&wt_est_hou_inc910_ethn2_count/&xt_est_hou_inc910_ethn2_count);
		output;
	end;
*proc print data=rp_est_hou_inc910_ethn2;
run;
data wt_est_hou_inc910_ethn2;
	set rp_est_hou_inc910_ethn2 temp;
*proc print data=wt_est_hou_inc910_ethn2;
run;
****************************************;
data xt_est_hou_inc910_ethn3;
	set q_combined;
	if (location=2 and estim=1 and (income=9 or income=10) and ethnicity=3) then
output;
*proc print data=xt_est_hou_inc910_ethn3;
run;
data rnd_est_hou_inc910_ethn3;
	set xt_est_hou_inc910_ethn3;
	rnd=uniform(55)*((&xt_est_hou_inc910_ethn3_count)-1)+1;
	output;
run;
proc sort data=rnd_est_hou_inc910_ethn3;
	by rnd;
```

```
*proc print data=rnd_est_hou_inc910_ethn3;
run;
data temp;
set rnd_est_hou_inc910_ethn3;
if _n_ <= (&wt_est_hou_inc910_ethn3_count)-
int(&wt_est_hou_inc910_ethn3_count/&xt_est_hou_inc910_ethn3_count)*(&xt_est_ho
u_inc910_ethn3_count);
*proc print data=temp;
run;
data rp_est_hou_inc910_ethn3;
        set xt_est_hou_inc910_ethn3;
        do i=1 to
int(&wt_est_hou_inc910_ethn3_count/&xt_est_hou_inc910_ethn3_count);
                output;
        end;
*proc print data=rp_est_hou_inc910_ethn3;
run;
data wt_est_hou_inc910_ethn3;
        set rp_est_hou_inc910_ethn3 temp;
*proc print data=wt_est_hou_inc910_ethn3;
run;
****************************************;
data xt_est_hou_inc910_ethn456;
        set q_combined;
        if (location=2 and estim=1 and (income=9 or income=10) and (ethnicity=4 or
ethnicity=5 or ethnicity=6)) then output;
*proc print data=xt_est_hou_inc910_ethn456;
run;
data rnd_est_hou_inc910_ethn456;
        set xt_est_hou_inc910_ethn456;
        rnd=uniform(56)*((&xt_est_hou_inc910_ethn456_count)-1)+1;
        output;
run;
proc sort data=rnd_est_hou_inc910_ethn456;
        by rnd;
*proc print data=rnd_est_hou_inc910_ethn456;
run;
data temp;
set rnd_est_hou_inc910_ethn456;
if _n_ <= (&wt_est_hou_inc910_ethn456_count)-
int(&wt_est_hou_inc910_ethn456_count/&xt_est_hou_inc910_ethn456_count)*(&xt_e
st_hou_inc910_ethn456_count);
*proc print data=temp;
run;
```

```
data rp_est_hou_inc910_ethn456;
        set xt_est_hou_inc910_ethn456;
        do i=1 to
int(&wt_est_hou_inc910_ethn456_count/&xt_est_hou_inc910_ethn456_count);
                output;
        end;
*proc print data=rp_est_hou_inc910_ethn456;
run;
data wt_est_hou_inc910_ethn456;
        set rp_est_hou_inc910_ethn456 temp;
*proc print data=wt_est_hou_inc910_ethn456;
run;
****************************************;
****************************************;
data wt_est_hou;
set
wt_est_hou_inc123_ethn1
wt_est_hou_inc123_ethn2
wt_est_hou_inc123_ethn3
wt_est_hou_inc123_ethn456
wt_est_hou_inc8_ethn1
wt_est_hou_inc8_ethn2
wt_est_hou_inc8_ethn3
wt_est_hou_inc8_ethn456
wt_est_hou_inc4_ethn1
wt_est_hou_inc4_ethn2
wt_est_hou_inc4_ethn3
wt_est_hou_inc4_ethn456
wt_est_hou_inc5_ethn1
wt_est_hou_inc5_ethn2
wt_est_hou_inc5_ethn3
wt_est_hou_inc5_ethn456
wt_est_hou_inc6_ethn1
wt_est_hou_inc6_ethn2
wt_est_hou_inc6_ethn3
wt_est_hou_inc6_ethn456
wt_est_hou_inc7_ethn1
wt_est_hou_inc7_ethn2
wt_est_hou_inc7_ethn3
wt_est_hou_inc7_ethn456
wt_est_hou_inc910_ethn1
wt_est_hou_inc910_ethn2
wt_est_hou_inc910_ethn3
wt_est_hou_inc910_ethn456;
```

```
*proc print data = wt_est_hou;
run;
*/
data xt_val_dfw;
        set q_combined;
        if (location=1 and estim=0) then output;
*proc print data=xt_val_dfw;
run;
data wt1_est_dfw(drop= i);
        set wt_est_dfw;
        output;
run;
data dfw;
set wt1_est_dfw xt_val_dfw;
run;
/*
data xt_val_hou;
        set q_combined;
        if (location=2 and estim=0) then output;
*proc print data=xt_val_hou;
run;
data wt1_est_hou(drop= i);
        set wt_est_hou;
        output;
run;
data hou;
set wt1_est_hou xt_val_hou;
run;

data xt_val_tex;
        set q_combined;
        if (estim=0) then output;
*proc print data=xt_val_tex;
run;
data wt1_est_tex(drop= i);
        set wt_est_tex;
        output;
run;
data tex;
set wt1_est_tex xt_val_tex;
run;
*/
data newdata(keep= pid id decision decision1 dec_mode dec_route
mode_route mode route
```

ttime
ttime_1 ttime_2 ttime_3 ttime_4 ttime_5 ttime_6
ttime_1_3_5 ttime_2_4_6
toll tollptoll
toll_1 toll_3 toll_5
toll_1_3_5
age gender ethnicity hhtype hhnum motorveh occtype educ income mlbin fampool
tollpay
unit
unit_r unit_m
unit_1 unit_2 unit_3 unit_4 unit_5 unit_6
unit_1_3_5 unit_2_4_6
age_d1 age_d234
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d345
occtyp_d1246 occtyp_d35 occtyp_d8 occtyp_d710
educ_d12
income_d1234 income_d56
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll

gender_1 gender_2 gender_3 gender_4 gender_5 gender_6
hhnum_1 hhnum_2 hhnum_3 hhnum_4 hhnum_5 hhnum_6
motorveh_1 motorveh_2 motorveh_3 motorveh_4 motorveh_5 motorveh_6
mlbin_1 mlbin_2 mlbin_3 mlbin_4 mlbin_5 mlbin_6
fampool_1 fampool_2 fampool_3 fampool_4 fampool_5 fampool_6
tollpay_1 tollpay_2 tollpay_3 tollpay_4 tollpay_5 tollpay_6
age_d1_1 age_d234_1
age_d1_2 age_d234_2
age_d1_3 age_d234_3
age_d1_4 age_d234_4
age_d1_5 age_d234_5
age_d1_6 age_d234_6
ethni_d1_1 ethni_d2_1 ethni_d3_1
ethni_d1_2 ethni_d2_2 ethni_d3_2
ethni_d1_3 ethni_d2_3 ethni_d3_3
ethni_d1_4 ethni_d2_4 ethni_d3_4
ethni_d1_5 ethni_d2_5 ethni_d3_5
ethni_d1_6 ethni_d2_6 ethni_d3_6
hhtyp_d1_1 hhtyp_d2_1 hhtyp_d345_1
hhtyp_d1_2 hhtyp_d2_2 hhtyp_d345_2
hhtyp_d1_3 hhtyp_d2_3 hhtyp_d345_3
hhtyp_d1_4 hhtyp_d2_4 hhtyp_d345_4
hhtyp_d1_5 hhtyp_d2_5 hhtyp_d345_5

```
hhtyp_d1_6 hhtyp_d2_6 hhtyp_d345_6
occtyp_d1246_1 occtyp_d35_1 occtyp_d8_1 occtyp_d710_1
occtyp_d1246_2 occtyp_d35_2 occtyp_d8_2 occtyp_d710_2
occtyp_d1246_3 occtyp_d35_3 occtyp_d8_3 occtyp_d710_3
occtyp_d1246_4 occtyp_d35_4 occtyp_d8_4 occtyp_d710_4
occtyp_d1246_5 occtyp_d35_5 occtyp_d8_5 occtyp_d710_5
occtyp_d1246_6 occtyp_d35_6 occtyp_d8_6 occtyp_d710_6
educ_d12_1
educ_d12_2
educ_d12_3
educ_d12_4
educ_d12_5
educ_d12_6
income_d1234_1 income_d56_1
income_d1234_2 income_d56_2
income_d1234_3 income_d56_3
income_d1234_4 income_d56_4
income_d1234_5 income_d56_5
income_d1234_6 income_d56_6
purpose_d1_1 purpose_d2_1 purpose_d3_1
purpose_d1_2 purpose_d2_2 purpose_d3_2
purpose_d1_3 purpose_d2_3 purpose_d3_3
purpose_d1_4 purpose_d2_4 purpose_d3_4
purpose_d1_5 purpose_d2_5 purpose_d3_5
purpose_d1_6 purpose_d2_6 purpose_d3_6
pcomtoll_1 pcomtoll_3 pcomtoll_5
pwrktoll_1 pwrktoll_3 pwrktoll_5
pschtoll_1 pschtoll_3 pschtoll_5
prectoll_1 prectoll_3 prectoll_5
tollptoll_1 tollptoll_3 tollptoll_5
);

        set dfw;
*       set q_combined;
        array tvec{6} tt1 - tt6;
        array tollvec{6} toll1 - toll6;

        retain pid 0;
        pid + 1;
        do i = 1 to 6;
                id = surveyid;
                if estim=1 then decision = ( SPCh = i );
                if estim=0 then decision = .;
                if estim=0 then decision1 = ( SPCh = i );
```

```
                mode_route = i;
                if (i=1 or i=2) then mode=1;
                if (i=3 or i=4) then mode=2;
                if (i=5 or i=6) then mode=3;
                if (i=1 or i=3 or i=5) then route=1;
                if (i=2 or i=4 or i=6) then route=2;
                ttime = tvec{i};
                toll = tollvec{i};
                if (i=2 or i=4 or i=6) then toll=0;
*****************************;
*INTERACTIONS;
*****************************;
                pcomtoll=0;    pwrktoll=0;    pschtoll=0;     prectoll=0;
                if purpagg=1 then pcomtoll = toll;
                if purpagg=3 then pwrktoll = toll;
                if purpagg=4 then pschtoll = toll;
                if purpagg=2 then prectoll = toll;

                tollptoll=0;
                if tollpay=1 then tollptoll = toll;
*****************************;
*DUMMIES;
*****************************;
                age_d1=0; age_d234=0;
                if age=1 then age_d1=1;
                if (age=2 or age=3 or age=4) then age_d234=1;
                ethni_d1=0;    ethni_d2=0;    ethni_d3=0;
                if ethnicity=1 then ethni_d1=1;
                if ethnicity=2 then ethni_d2=1;
                if ethnicity=3 then ethni_d3=1;
                hhtyp_d1=0;   hhtyp_d2=0;   hhtyp_d345=0;
                if hhtype=1 then hhtyp_d1=1;
                if hhtype=2 then hhtyp_d2=1;
                if (hhtype=3 or hhtype=4 or hhtype=5) then hhtyp_d345=1;
                occtyp_d1246=0; occtyp_d35=0; occtyp_d8=0; occtyp_d710=0;
                if (occtype=1 or occtype=2 or occtype=4 or occtype=6) then
occtyp_d1246=1;
                if (occtype=3 or occtype=5) then occtyp_d35=1;
                if occtype=8 then occtyp_d8=1;
                if (occtype=7 or occtype=10) then occtyp_d710=1;
                educ_d12=0;
                if (educ=1 or educ=2) then educ_d12=1;
                income_d1234=0; income_d56=0;
```

```
            if (income=1 or income=2 or income=3 or income=4) then
income_d1234=1;
            if (income=5 or income=6) then income_d56=1;
            purpose_d1=0; purpose_d2=0;         purpose_d3=0;
            if purpagg=1 then purpose_d1 = 1;
            if purpagg=3 then purpose_d2 = 1;
            if purpagg=4 then purpose_d3 = 1;


*****************************;
*MODE SPECIFICS;
*****************************;
            ttime_1=0;              toll_1=0;
            ttime_2=0;
            ttime_3=0;              toll_3=0;
            ttime_4=0;
            ttime_5=0;              toll_5=0;
            ttime_6=0;
            if (i EQ 1) then ttime_1=ttime;             if (i EQ 1) then toll_1=toll;
            if (i EQ 2) then ttime_2=ttime;
            if (i EQ 3) then ttime_3=ttime;             if (i EQ 3) then toll_3=toll;
            if (i EQ 4) then ttime_4=ttime;
            if (i EQ 5) then ttime_5=ttime;             if (i EQ 5) then toll_5=toll;
            if (i EQ 6) then ttime_6=ttime;
            unit = 1; unit_r=1; unit_m=1;
            unit_1 = 0;
            unit_2 = 0;
            unit_3 = 0;
            unit_4 = 0;
            unit_5 = 0;
            unit_6 = 0;

            if (i EQ 1) then unit_1=1;
            if (i EQ 2) then unit_2=1;
            if (i EQ 3) then unit_3=1;
            if (i EQ 4) then unit_4=1;
            if (i EQ 5) then unit_5=1;
            if (i EQ 6) then unit_6=1;
*********************************;
            toll_1_3_5 = tollvec{i};
            if (i=2 or i=4 or i=6) then toll_1_3_5=0;
            ttime_1_3_5=0;
            ttime_2_4_6=0;
            if (i=1 or i=3 or i=5) then ttime_1_3_5=ttime;
```

```
                if (i=2 or i=4 or i=6) then ttime_2_4_6=ttime;
                unit_1_3_5 = 0;
                unit_2_4_6 = 0;
                if (i=1 or i=3 or i=5) then unit_1_3_5=1;
                if (i=2 or i=4 or i=6) then unit_2_4_6=1;
*********************************;
                if (i EQ 1) then unit_1=1;
                if (i EQ 2) then unit_2=1;
                if (i EQ 3) then unit_3=1;
                if (i EQ 4) then unit_4=1;
                if (i EQ 5) then unit_5=1;
                if (i EQ 6) then unit_6=1;
*******************************;
                gender_1=0; gender_2=0; gender_3=0; gender_4=0; gender_5=0;
gender_6=0;
                if i=1 then gender_1=gender;
                if i=2 then gender_2=gender;
                if i=3 then gender_3=gender;
                if i=4 then gender_4=gender;
                if i=5 then gender_5=gender;
                if i=6 then gender_6=gender;
                hhnum_1=0; hhnum_2=0; hhnum_3=0; hhnum_4=0; hhnum_5=0;
hhnum_6=0;
                if i=1 then hhnum_1=hhnum;
                if i=2 then hhnum_2=hhnum;
                if i=3 then hhnum_3=hhnum;
                if i=4 then hhnum_4=hhnum;
                if i=5 then hhnum_5=hhnum;
                if i=6 then hhnum_6=hhnum;
                motorveh_1=0; motorveh_2=0; motorveh_3=0; motorveh_4=0;
motorveh_5=0; motorveh_6=0;
                if i=1 then motorveh_1=motorveh;
                if i=2 then motorveh_2=motorveh;
                if i=3 then motorveh_3=motorveh;
                if i=4 then motorveh_4=motorveh;
                if i=5 then motorveh_5=motorveh;
                if i=6 then motorveh_6=motorveh;
                mlbin_1=0; mlbin_2=0; mlbin_3=0; mlbin_4=0; mlbin_5=0; mlbin_6=0;
                if i=1 then mlbin_1=mlbin;
                if i=2 then mlbin_2=mlbin;
                if i=3 then mlbin_3=mlbin;
                if i=4 then mlbin_4=mlbin;
                if i=5 then mlbin_5=mlbin;
                if i=6 then mlbin_6=mlbin;
```

```
                fampool_1=0; fampool_2=0; fampool_3=0; fampool_4=0; fampool_5=0;
fampool_6=0;
                if i=1 then fampool_1=fampool;
                if i=2 then fampool_2=fampool;
                if i=3 then fampool_3=fampool;
                if i=4 then fampool_4=fampool;
                if i=5 then fampool_5=fampool;
                if i=6 then fampool_6=fampool;
                tollpay_1=0; tollpay_2=0; tollpay_3=0; tollpay_4=0; tollpay_5=0;
tollpay_6=0;
                if i=1 then tollpay_1=tollpay;
                if i=2 then tollpay_2=tollpay;
                if i=3 then tollpay_3=tollpay;
                if i=4 then tollpay_4=tollpay;
                if i=5 then tollpay_5=tollpay;
                if i=6 then tollpay_6=tollpay;
                age_d1_1=0; age_d1_2=0; age_d1_3=0; age_d1_4=0; age_d1_5=0;
age_d1_6=0;
                age_d234_1=0; age_d234_2=0; age_d234_3=0; age_d234_4=0;
age_d234_5=0; age_d234_6=0;
                if i=1 then age_d1_1=age_d1;
                if i=1 then age_d234_1=age_d234;
                if i=2 then age_d1_2=age_d1;
                if i=2 then age_d234_2=age_d234;
                if i=3 then age_d1_3=age_d1;
                if i=3 then age_d234_3=age_d234;
                if i=4 then age_d1_4=age_d1;
                if i=4 then age_d234_4=age_d234;
                if i=5 then age_d1_5=age_d1;
                if i=5 then age_d234_5=age_d234;
                if i=6 then age_d1_6=age_d1;
                if i=6 then age_d234_6=age_d234;
                ethni_d1_1=0; ethni_d1_2=0; ethni_d1_3=0; ethni_d1_4=0;
ethni_d1_5=0; ethni_d1_6=0;
                ethni_d2_1=0; ethni_d2_2=0; ethni_d2_3=0; ethni_d2_4=0;
ethni_d2_5=0; ethni_d2_6=0;
                ethni_d3_1=0; ethni_d3_2=0; ethni_d3_3=0; ethni_d3_4=0;
ethni_d3_5=0; ethni_d3_6=0;
                if i=1 then ethni_d1_1=ethni_d1;
                if i=1 then ethni_d2_1=ethni_d2;
                if i=1 then ethni_d3_1=ethni_d3;
                if i=2 then ethni_d1_2=ethni_d1;
                if i=2 then ethni_d2_2=ethni_d2;
                if i=2 then ethni_d3_2=ethni_d3;
```

```
                if i=3 then ethni_d1_3=ethni_d1;
                if i=3 then ethni_d2_3=ethni_d2;
                if i=3 then ethni_d3_3=ethni_d3;
                if i=4 then ethni_d1_4=ethni_d1;
                if i=4 then ethni_d2_4=ethni_d2;
                if i=4 then ethni_d3_4=ethni_d3;
                if i=5 then ethni_d1_5=ethni_d1;
                if i=5 then ethni_d2_5=ethni_d2;
                if i=5 then ethni_d3_5=ethni_d3;
                if i=6 then ethni_d1_6=ethni_d1;
                if i=6 then ethni_d2_6=ethni_d2;
                if i=6 then ethni_d3_6=ethni_d3;
                hhtyp_d1_1=0; hhtyp_d1_2=0; hhtyp_d1_3=0; hhtyp_d1_4=0;
hhtyp_d1_5=0; hhtyp_d1_6=0;
                hhtyp_d2_1=0; hhtyp_d2_2=0; hhtyp_d2_3=0; hhtyp_d2_4=0;
hhtyp_d2_5=0; hhtyp_d2_6=0;
                hhtyp_d345_1=0; hhtyp_d345_2=0; hhtyp_d345_3=0; hhtyp_d345_4=0;
hhtyp_d345_5=0; hhtyp_d345_6=0;
                if i=1 then hhtyp_d1_1=hhtyp_d1;
                if i=1 then hhtyp_d2_1=hhtyp_d2;
                if i=1 then hhtyp_d345_1=hhtyp_d345;
                if i=2 then hhtyp_d1_2=hhtyp_d1;
                if i=2 then hhtyp_d2_2=hhtyp_d2;
                if i=2 then hhtyp_d345_2=hhtyp_d345;
                if i=3 then hhtyp_d1_3=hhtyp_d1;
                if i=3 then hhtyp_d2_3=hhtyp_d2;
                if i=3 then hhtyp_d345_3=hhtyp_d345;
                if i=4 then hhtyp_d1_4=hhtyp_d1;
                if i=4 then hhtyp_d2_4=hhtyp_d2;
                if i=4 then hhtyp_d345_4=hhtyp_d345;
                if i=5 then hhtyp_d1_5=hhtyp_d1;
                if i=5 then hhtyp_d2_5=hhtyp_d2;
                if i=5 then hhtyp_d345_5=hhtyp_d345;
                if i=6 then hhtyp_d1_6=hhtyp_d1;
                if i=6 then hhtyp_d2_6=hhtyp_d2;
                if i=6 then hhtyp_d345_6=hhtyp_d345;
                occtyp_d1246_1=0; occtyp_d1246_2=0; occtyp_d1246_3=0;
occtyp_d1246_4=0; occtyp_d1246_5=0; occtyp_d1246_6=0;
                occtyp_d35_1=0; occtyp_d35_2=0; occtyp_d35_3=0; occtyp_d35_4=0;
occtyp_d35_5=0; occtyp_d35_6=0;
                occtyp_d8_1=0; occtyp_d8_2=0; occtyp_d8_3=0; occtyp_d8_4=0;
occtyp_d8_5=0; occtyp_d8_6=0;
                occtyp_d710_1=0; occtyp_d710_2=0; occtyp_d710_3=0;
occtyp_d710_4=0; occtyp_d710_5=0; occtyp_d710_6=0;
```

```
                if i=1 then occtyp_d1246_1=occtyp_d1246;
                if i=1 then occtyp_d35_1=occtyp_d35;
                if i=1 then occtyp_d8_1=occtyp_d8;
                if i=1 then occtyp_d710_1=occtyp_d710;
                if i=2 then occtyp_d1246_2=occtyp_d1246;
                if i=2 then occtyp_d35_2=occtyp_d35;
                if i=2 then occtyp_d8_2=occtyp_d8;
                if i=2 then occtyp_d710_2=occtyp_d710;
                if i=3 then occtyp_d1246_3=occtyp_d1246;
                if i=3 then occtyp_d35_3=occtyp_d35;
                if i=3 then occtyp_d8_3=occtyp_d8;
                if i=3 then occtyp_d710_3=occtyp_d710;
                if i=4 then occtyp_d1246_4=occtyp_d1246;
                if i=4 then occtyp_d35_4=occtyp_d35;
                if i=4 then occtyp_d8_4=occtyp_d8;
                if i=4 then occtyp_d710_4=occtyp_d710;
                if i=5 then occtyp_d1246_5=occtyp_d1246;
                if i=5 then occtyp_d35_5=occtyp_d35;
                if i=5 then occtyp_d8_5=occtyp_d8;
                if i=5 then occtyp_d710_5=occtyp_d710;
                if i=6 then occtyp_d1246_6=occtyp_d1246;
                if i=6 then occtyp_d35_6=occtyp_d35;
                if i=6 then occtyp_d8_6=occtyp_d8;
                if i=6 then occtyp_d710_6=occtyp_d710;
                educ_d12_1=0; educ_d12_2=0; educ_d12_3=0; educ_d12_4=0;
educ_d12_5=0; educ_d12_6=0;
                if i=1 then educ_d12_1=educ_d12;
                if i=2 then educ_d12_2=educ_d12;
                if i=3 then educ_d12_3=educ_d12;
                if i=4 then educ_d12_4=educ_d12;
                if i=5 then educ_d12_5=educ_d12;
                if i=6 then educ_d12_6=educ_d12;
                income_d1234_1=0; income_d1234_2=0; income_d1234_3=0;
income_d1234_4=0; income_d1234_5=0; income_d1234_6=0;
                income_d56_1=0; income_d56_2=0; income_d56_3=0;
income_d56_4=0; income_d56_5=0; income_d56_6=0;
                if i=1 then income_d1234_1=income_d1234;
                if i=1 then income_d56_1=income_d56;
                if i=2 then income_d1234_2=income_d1234;
                if i=2 then income_d56_2=income_d56;
                if i=3 then income_d1234_3=income_d1234;
                if i=3 then income_d56_3=income_d56;
                if i=4 then income_d1234_4=income_d1234;
                if i=4 then income_d56_4=income_d56;
```

```
                if i=5 then income_d1234_5=income_d1234;
                if i=5 then income_d56_5=income_d56;
                if i=6 then income_d1234_6=income_d1234;
                if i=6 then income_d56_6=income_d56;
                purpose_d1_1=0; purpose_d1_2=0; purpose_d1_3=0; purpose_d1_4=0;
purpose_d1_5=0; purpose_d1_6=0;
                purpose_d2_1=0; purpose_d2_2=0; purpose_d2_3=0; purpose_d2_4=0;
purpose_d2_5=0; purpose_d2_6=0;
                purpose_d3_1=0; purpose_d3_2=0; purpose_d3_3=0; purpose_d3_4=0;
purpose_d3_5=0; purpose_d3_6=0;
                if i=1 then purpose_d1_1=purpose_d1;
                if i=1 then purpose_d2_1=purpose_d2;
                if i=1 then purpose_d3_1=purpose_d3;
                if i=2 then purpose_d1_2=purpose_d1;
                if i=2 then purpose_d2_2=purpose_d2;
                if i=2 then purpose_d3_2=purpose_d3;
                if i=3 then purpose_d1_3=purpose_d1;
                if i=3 then purpose_d2_3=purpose_d2;
                if i=3 then purpose_d3_3=purpose_d3;
                if i=4 then purpose_d1_4=purpose_d1;
                if i=4 then purpose_d2_4=purpose_d2;
                if i=4 then purpose_d3_4=purpose_d3;
                if i=5 then purpose_d1_5=purpose_d1;
                if i=5 then purpose_d2_5=purpose_d2;
                if i=5 then purpose_d3_5=purpose_d3;
                if i=6 then purpose_d1_6=purpose_d1;
                if i=6 then purpose_d2_6=purpose_d2;
                if i=6 then purpose_d3_6=purpose_d3;
*****************************;
                pcomtoll_1=0; pcomtoll_3=0; pcomtoll_5=0;
                if i=1 then pcomtoll_1=pcomtoll;
                if i=3 then pcomtoll_3=pcomtoll;
                if i=5 then pcomtoll_5=pcomtoll;
                pwrktoll_1=0; pwrktoll_3=0; pwrktoll_5=0;
                if i=1 then pwrktoll_1=pwrktoll;
                if i=3 then pwrktoll_3=pwrktoll;
                if i=5 then pwrktoll_5=pwrktoll;
                pschtoll_1=0; pschtoll_3=0; pschtoll_5=0;
                if i=1 then pschtoll_1=pschtoll;
                if i=3 then pschtoll_3=pschtoll;
                if i=5 then pschtoll_5=pschtoll;
                prectoll_1=0; prectoll_3=0; prectoll_5=0;
                if i=1 then prectoll_1=prectoll;
                if i=3 then prectoll_3=prectoll;
```

```
                if i=5 then prectoll_5=prectoll;
                tollptoll_1=0; tollptoll_3=0; tollptoll_5=0;
                if i=1 then tollptoll_1=tollptoll;
                if i=3 then tollptoll_3=tollptoll;
                if i=5 then tollptoll_5=tollptoll;
**************************************************************************
*;
                output;
        end;
*proc print data=newdata;
run;
*/************************************************/
* MULTINOMIAL LOGIT WITH---GENERIC---PARAMETERS (ALL VARS)
proc mdc data=newdata maxit=300;
        model decision = unit ttime toll
gender hhnum motorveh mlbin fampool tollpay
age_d1 age_d234
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d345
occtyp_d1246 occtyp_d35 occtyp_d8 occtyp_d710
educ_d12
income_d1234 income_d56
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll
/ type=clogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
        id pid;
        output out=probdata pred=p;
*proc print data=probdata;
*run;
*/************************************************/
* MULTINOMIAL LOGIT WITH---GENERIC---PARAMETERS (ALL VARS)-->
NONSIG VAR REMOVED
proc mdc data=newdata maxit=300;
        model decision = unit ttime toll
gender /*hhnum motorveh*/ mlbin fampool tollpay
age_d1 age_d234
ethni_d1 ethni_d2 ethni_d3
/*hhtyp_d1 hhtyp_d2 hhtyp_d345*/
occtyp_d1246 occtyp_d35 occtyp_d8 occtyp_d710
educ_d12
income_d1234 income_d56
purpose_d1 purpose_d2 purpose_d3
pcomtoll /*pwrktoll*/ pschtoll prectoll /*tollptoll*/
/ type=clogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
```

```
        id pid;
        output out=probdata pred=p;
*proc print data=probdata;
*run;
*/**********************************************/

* MULTINOMIAL LOGIT WITH---ALTERNATIVE SPECIFIC---PARAMETERS
(ALL VARS);
proc mdc data=newdata maxit=600;
        model decision = unit_1 unit_2 unit_3 unit_4 unit_5 unit_6
ttime_1 ttime_2 ttime_3 ttime_4 ttime_5 ttime_6
toll_1 toll_3 toll_5
gender_1 gender_2 gender_3 gender_4 gender_5 gender_6
hhnum_1 hhnum_2 hhnum_3 hhnum_4 hhnum_5 hhnum_6
motorveh_1 motorveh_2 motorveh_3 motorveh_4 motorveh_5 motorveh_6
mlbin_1 mlbin_2 mlbin_3 mlbin_4 mlbin_5 mlbin_6
fampool_1 fampool_2 fampool_3 fampool_4 fampool_5 fampool_6
tollpay_1 tollpay_2 tollpay_3 tollpay_4 tollpay_5 tollpay_6
age_d1_1 age_d234_1
age_d1_2 age_d234_2
age_d1_3 age_d234_3
age_d1_4 age_d234_4
age_d1_5 age_d234_5
age_d1_6 age_d234_6
ethni_d1_1 ethni_d2_1 ethni_d3_1
ethni_d1_2 ethni_d2_2 ethni_d3_2
ethni_d1_3 ethni_d2_3 ethni_d3_3
ethni_d1_4 ethni_d2_4 ethni_d3_4
ethni_d1_5 ethni_d2_5 ethni_d3_5
ethni_d1_6 ethni_d2_6 ethni_d3_6
hhtyp_d1_1 hhtyp_d2_1 hhtyp_d345_1
hhtyp_d1_2 hhtyp_d2_2 hhtyp_d345_2
hhtyp_d1_3 hhtyp_d2_3 hhtyp_d345_3
hhtyp_d1_4 hhtyp_d2_4 hhtyp_d345_4
hhtyp_d1_5 hhtyp_d2_5 hhtyp_d345_5
hhtyp_d1_6 hhtyp_d2_6 hhtyp_d345_6
occtyp_d1246_1 occtyp_d35_1 occtyp_d8_1 occtyp_d710_1
occtyp_d1246_2 occtyp_d35_2 occtyp_d8_2 occtyp_d710_2
occtyp_d1246_3 occtyp_d35_3 occtyp_d8_3 occtyp_d710_3
occtyp_d1246_4 occtyp_d35_4 occtyp_d8_4 occtyp_d710_4
occtyp_d1246_5 occtyp_d35_5 occtyp_d8_5 occtyp_d710_5
occtyp_d1246_6 occtyp_d35_6 occtyp_d8_6 occtyp_d710_6
educ_d12_1
educ_d12_2
```

```
educ_d12_3
educ_d12_4
educ_d12_5
educ_d12_6
income_d1234_1 income_d56_1
income_d1234_2 income_d56_2
income_d1234_3 income_d56_3
income_d1234_4 income_d56_4
income_d1234_5 income_d56_5
income_d1234_6 income_d56_6
purpose_d1_1 purpose_d2_1 purpose_d3_1
purpose_d1_2 purpose_d2_2 purpose_d3_2
purpose_d1_3 purpose_d2_3 purpose_d3_3
purpose_d1_4 purpose_d2_4 purpose_d3_4
purpose_d1_5 purpose_d2_5 purpose_d3_5
purpose_d1_6 purpose_d2_6 purpose_d3_6
pcomtoll_1 pcomtoll_3 pcomtoll_5
pwrktoll_1 pwrktoll_3 pwrktoll_5
pschtoll_1 pschtoll_3 pschtoll_5
prectoll_1 prectoll_3 prectoll_5
tollptoll_1 tollptoll_3 tollptoll_5
/ type=clogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
        id pid;
        output out=probdata pred=p;
*proc print data=probdata;
run;
*/***********************************************/
* MULTINOMIAL LOGIT WITH---ALTERNATIVE SPECIFIC---PARAMETERS
(NON-SIGS REMOVED)
proc mdc data=newdata maxit=600;
        model decision = unit_1 unit_2 unit_3 unit_4 /*unit_5*/ unit_6
ttime_1 ttime_2 ttime_3 ttime_4 /*ttime_5*/ ttime_6
toll_1 toll_3 toll_5
gender_1 gender_2 gender_3 /*gender_4 gender_5*/ gender_6
hhnum_1 hhnum_2 hhnum_3 hhnum_4 hhnum_5 hhnum_6
motorveh_1 motorveh_2 motorveh_3 motorveh_4 motorveh_5 motorveh_6
mlbin_1 mlbin_2 mlbin_3 mlbin_4 mlbin_5 mlbin_6
fampool_1 fampool_2 fampool_3 fampool_4 fampool_5 /*fampool_6*/
tollpay_1 tollpay_2 /*tollpay_3*/ tollpay_4 tollpay_5 tollpay_6
age_d1_1 /*age_d234_1 */
age_d1_2 age_d234_2
/*age_d1_3*/ age_d234_3
age_d1_4 age_d234_4
age_d1_5 age_d234_5
```

```
age_d1_6 /*age_d234_6 */
ethni_d1_1 ethni_d2_1 /*ethni_d3_1*/
ethni_d1_2 ethni_d2_2 ethni_d3_2
ethni_d1_3 ethni_d2_3 ethni_d3_3
ethni_d1_4 ethni_d2_4 ethni_d3_4
ethni_d1_5 ethni_d2_5 ethni_d3_5
/*ethni_d1_6 ethni_d2_6 ethni_d3_6*/
hhtyp_d1_1 hhtyp_d2_1 hhtyp_d345_1
hhtyp_d1_2 hhtyp_d2_2 hhtyp_d345_2
hhtyp_d1_3 hhtyp_d2_3 hhtyp_d345_3
hhtyp_d1_4 hhtyp_d2_4 hhtyp_d345_4
hhtyp_d1_5 hhtyp_d2_5 hhtyp_d345_5
hhtyp_d1_6 hhtyp_d2_6 hhtyp_d345_6
occtyp_d1246_1 occtyp_d35_1 occtyp_d8_1 occtyp_d710_1
occtyp_d1246_2 occtyp_d35_2 occtyp_d8_2 occtyp_d710_2
occtyp_d1246_3 occtyp_d35_3 occtyp_d8_3 occtyp_d710_3
occtyp_d1246_4 /*occtyp_d35_4*/ occtyp_d8_4 occtyp_d710_4
occtyp_d1246_5 /*occtyp_d35_5 occtyp_d8_5*/ occtyp_d710_5
occtyp_d1246_6 occtyp_d35_6 occtyp_d8_6 occtyp_d710_6
educ_d12_1
educ_d12_2
educ_d12_3
educ_d12_4
/*educ_d12_5 */
educ_d12_6
income_d1234_1 /*income_d56_1 */
income_d1234_2 income_d56_2
income_d1234_3 income_d56_3
/*income_d1234_4 income_d56_4 */
income_d1234_5 income_d56_5
income_d1234_6 income_d56_6
purpose_d1_1 /*purpose_d2_1 purpose_d3_1 */
purpose_d1_2 purpose_d2_2 purpose_d3_2
purpose_d1_3 purpose_d2_3 purpose_d3_3
purpose_d1_4 purpose_d2_4 purpose_d3_4
purpose_d1_5 purpose_d2_5 /*purpose_d3_5 */
/*purpose_d1_6 purpose_d2_6*/ purpose_d3_6
pcomtoll_1 pcomtoll_3 pcomtoll_5
pwrktoll_1 pwrktoll_3 pwrktoll_5
/*pschtoll_1*/ pschtoll_3 pschtoll_5
prectoll_1 prectoll_3 prectoll_5
tollptoll_1 tollptoll_3 tollptoll_5
/ type=clogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
        id pid;
```

```
        output out=probdata pred=p;
*proc print data=probdata;
*run;
*/***********************************************/
* MULTINOMIAL LOGIT WITH---GENERIC---PARAMETERS (CONST. TTIME
TOLL)
proc mdc data=newdata maxit=300;
        model decision = unit ttime toll
/ type=clogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
        id pid;
        output out=probdata pred=p;
*proc print data=probdata;
*run;
*/***********************************************/
* MULTINOMIAL LOGIT WITH---MODE SPECIFIC---PARAMETERS (CONST.
TTIME TOLL)
proc mdc data=newdata maxit=300;
        model decision = unit_1 unit_2 unit_3 unit_4 unit_5 unit_6
ttime_1 ttime_2 ttime_3 ttime_4 ttime_5 ttime_6
toll_1 toll_3 toll_5
/ type=clogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
        id pid;
        output out=probdata pred=p;
*proc print data=probdata;
*run;
*/***********************************************/
* MULTINOMIAL LOGIT WITH---MODE SPECIFIC---PARAMETERS (CONST.
TTIME TOLL)-->NON SIG REMOVED
proc mdc data=newdata maxit=300;
        model decision = unit_1 unit_2 unit_3 unit_4 unit_5 unit_6
ttime_1 ttime_2 ttime_3 ttime_4 /*ttime_5*/ ttime_6
toll_1 toll_3 toll_5
/ type=clogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
        id pid;
        output out=probdata pred=p;
*proc print data=probdata;
*run;
*/***********************************************/
/* NESTED LOGIT ( MODE CHOICE -->ROUTE CHOICE (ALL VARS) )
proc mdc data=newdata maxit=300;
        model decision = unit ttime toll
gender hhnum motorveh mlbin fampool tollpay
age_d1 age_d234
ethni_d1 ethni_d2 ethni_d3
```

```
hhtyp_d1 hhtyp_d2 hhtyp_d345
occtyp_d1246 occtyp_d35 occtyp_d8 occtyp_d710
educ_d12
income_d1234 income_d56
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll
/ type=nlogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
        id pid;
        utility u(1,) = unit ttime toll
gender hhnum motorveh mlbin fampool tollpay
age_d1 age_d234
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d345
occtyp_d1246 occtyp_d35 occtyp_d8 occtyp_d710
educ_d12
income_d1234 income_d56
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll;
nest level(1) = (1 2 @ 1, 3 4 @ 2, 5 6 @ 3),
                        level(2) = (1 2 3 @ 1);

        output out=probdata pred=p;
*run;
*/*/***********************************************/
/* NESTED LOGIT ( MODE CHOICE -->ROUTE CHOICE (ALL VARS))---
INSIGNIFICANT VARS REMOVED(STEP1)
proc mdc data=newdata maxit=300;
        model decision = unit ttime toll
gender hhnum motorveh /*mlbin*/ fampool tollpay
/*age_d1*/ age_d234
ethni_d1 /*ethni_d2*/ ethni_d3
hhtyp_d1 /*hhtyp_d2*/ hhtyp_d345
occtyp_d1246 occtyp_d35 /*occtyp_d8*/ occtyp_d710
educ_d12
income_d1234 income_d56
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll
/ type=nlogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
        id pid;
        utility u(1,) = unit ttime toll
gender hhnum motorveh /*mlbin*/ fampool tollpay
/*age_d1*/ age_d234
ethni_d1 /*ethni_d2*/ ethni_d3
hhtyp_d1 /*hhtyp_d2*/ hhtyp_d345
```

```
occtyp_d1246 occtyp_d35 /*occtyp_d8*/ occtyp_d710
educ_d12
income_d1234 income_d56
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll;
nest level(1) = (1 2 @ 1, 3 4 @ 2, 5 6 @ 3),
                        level(2) = (1 2 3 @ 1);

        output out=probdata pred=p;
*run;
*/**********************************************/
/* NESTED LOGIT ( MODE CHOICE -->ROUTE CHOICE (ALL VARS))---
INSIGNIFICANT VARS REMOVED(STEP2)
proc mdc data=newdata maxit=300;
        model decision = unit ttime toll
gender hhnum motorveh /*mlbin*/ fampool tollpay
/*age_d1*/ age_d234
ethni_d1 /*ethni_d2*/ ethni_d3
hhtyp_d1 /*hhtyp_d2*/ hhtyp_d345
occtyp_d1246 occtyp_d35 /*occtyp_d8*/ occtyp_d710
educ_d12
income_d1234 /*income_d56*/
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll
/ type=nlogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
        id pid;
        utility u(1,) = unit ttime toll
gender hhnum motorveh /*mlbin*/ fampool tollpay
/*age_d1*/ age_d234
ethni_d1 /*ethni_d2*/ ethni_d3
hhtyp_d1 /*hhtyp_d2*/ hhtyp_d345
occtyp_d1246 occtyp_d35 /*occtyp_d8*/ occtyp_d710
educ_d12
income_d1234 /*income_d56*/
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll;
nest level(1) = (1 2 @ 1, 3 4 @ 2, 5 6 @ 3),
                        level(2) = (1 2 3 @ 1);

        output out=probdata pred=p;
*run;

*/**********************************************/
```

```
*/**********************************************/

*/*/**********************************************/

*/*/**********************************************/

*/*/**********************************************/

*/*/**********************************************/

*/*/**********************************************/

/**********************************************/
NESTED LOGIT ( ROUTE CHOICE -->MODE CHOICE (ALL VARS) )
proc mdc data=newdata maxit=300;
        model decision = unit ttime toll
gender hhnum motorveh mlbin fampool tollpay
age_d1 age_d234
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d345
occtyp_d1246 occtyp_d35 occtyp_d8 occtyp_d710
educ_d12
income_d1234 income_d56
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll
/ type=nlogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
        id pid;
        utility u(1,) = unit ttime toll
gender hhnum motorveh mlbin fampool tollpay
age_d1 age_d234
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d345
occtyp_d1246 occtyp_d35 occtyp_d8 occtyp_d710
educ_d12
income_d1234 income_d56
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll;
        nest level(1) = (1 3 5 @ 1, 2 4 6 @ 2),
                        level(2) = (1 2 @ 1);

        output out=probdata pred=p;
*run;
*/
/**********************************************/
```

```
NESTED LOGIT ( ROUTE CHOICE -->MODE CHOICE (ALL VARS) )----
INSIGNIFICANT VARS REMOVED (STEP 1)
proc mdc data=newdata maxit=300;
        model decision = unit ttime toll
gender /*hhnum motorveh*/ mlbin fampool /*tollpay */
age_d1 age_d234
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d345
occtyp_d1246 occtyp_d35 occtyp_d8 occtyp_d710
educ_d12
income_d1234 income_d56
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll
/ type=nlogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
        id pid;
        utility u(1,) = unit ttime toll
gender /*hhnum motorveh*/ mlbin fampool /*tollpay */
age_d1 age_d234
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d345
occtyp_d1246 occtyp_d35 occtyp_d8 occtyp_d710
educ_d12
income_d1234 income_d56
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll;
        nest level(1) = (1 3 5 @ 1, 2 4 6 @ 2),
                        level(2) = (1 2 @ 1);

        output out=probdata pred=p;
*run;
*/
/*********************************************/;

/*********************************************/
*NESTED LOGIT ( ROUTE CHOICE -->MODE CHOICE (ALL VARS) )----ALT
SPECIFIC
proc mdc data=newdata maxit=500;

        model decision = unit_1 unit_2 unit_3 unit_4 unit_5 unit_6
ttime_1 ttime_2 ttime_3 ttime_4 ttime_5 ttime_6
toll_1 toll_3 toll_5
gender_1 gender_2 gender_3 gender_4 gender_5 gender_6
hhnum_1 hhnum_2 hhnum_3 hhnum_4 hhnum_5 hhnum_6
motorveh_1 motorveh_2 motorveh_3 motorveh_4 motorveh_5 motorveh_6
```

mlbin_1 mlbin_2 mlbin_3 mlbin_4 mlbin_5 mlbin_6
fampool_1 fampool_2 fampool_3 fampool_4 fampool_5 fampool_6
tollpay_1 tollpay_2 tollpay_3 tollpay_4 tollpay_5 tollpay_6
age_d1_1 age_d2_1 age_d3_1 age_d4_1 age_d5_1
age_d1_2 age_d2_2 age_d3_2 age_d4_2 age_d5_2
age_d1_3 age_d2_3 age_d3_3 age_d4_3 age_d5_3
age_d1_4 age_d2_4 age_d3_4 age_d4_4 age_d5_4
age_d1_5 age_d2_5 age_d3_5 age_d4_5 age_d5_5
age_d1_6 age_d2_6 age_d3_6 age_d4_6 age_d5_6
ethni_d1_1 ethni_d2_1 ethni_d3_1
ethni_d1_2 ethni_d2_2 ethni_d3_2
ethni_d1_3 ethni_d2_3 ethni_d3_3
ethni_d1_4 ethni_d2_4 ethni_d3_4
ethni_d1_5 ethni_d2_5 ethni_d3_5
ethni_d1_6 ethni_d2_6 ethni_d3_6
hhtyp_d1_1 hhtyp_d2_1 hhtyp_d3_1 hhtyp_d4_1 hhtyp_d5_1
hhtyp_d1_2 hhtyp_d2_2 hhtyp_d3_2 hhtyp_d4_2 hhtyp_d5_2
hhtyp_d1_3 hhtyp_d2_3 hhtyp_d3_3 hhtyp_d4_3 hhtyp_d5_3
hhtyp_d1_4 hhtyp_d2_4 hhtyp_d3_4 hhtyp_d4_4 hhtyp_d5_4
hhtyp_d1_5 hhtyp_d2_5 hhtyp_d3_5 hhtyp_d4_5 hhtyp_d5_5
hhtyp_d1_6 hhtyp_d2_6 hhtyp_d3_6 hhtyp_d4_6 hhtyp_d5_6
occtyp_d1_1 occtyp_d2_1 occtyp_d3_1 occtyp_d4_1 occtyp_d5_1 occtyp_d6_1
occtyp_d7_1 occtyp_d8_1 occtyp_d9_1 occtyp_d10_1 occtyp_d11_1
occtyp_d1_2 occtyp_d2_2 occtyp_d3_2 occtyp_d4_2 occtyp_d5_2 occtyp_d6_2
occtyp_d7_2 occtyp_d8_2 occtyp_d9_2 occtyp_d10_2 occtyp_d11_2
occtyp_d1_3 occtyp_d2_3 occtyp_d3_3 occtyp_d4_3 occtyp_d5_3 occtyp_d6_3
occtyp_d7_3 occtyp_d8_3 occtyp_d9_3 occtyp_d10_3 occtyp_d11_3
occtyp_d1_4 occtyp_d2_4 occtyp_d3_4 occtyp_d4_4 occtyp_d5_4 occtyp_d6_4
occtyp_d7_4 occtyp_d8_4 occtyp_d9_4 occtyp_d10_4 occtyp_d11_4
occtyp_d1_5 occtyp_d2_5 occtyp_d3_5 occtyp_d4_5 occtyp_d5_5 occtyp_d6_5
occtyp_d7_5 occtyp_d8_5 occtyp_d9_5 occtyp_d10_5 occtyp_d11_5
occtyp_d1_6 occtyp_d2_6 occtyp_d3_6 occtyp_d4_6 occtyp_d5_6 occtyp_d6_6
occtyp_d7_6 occtyp_d8_6 occtyp_d9_6 occtyp_d10_6 occtyp_d11_6
educ_d1_1 educ_d2_1 educ_d3_1 educ_d4_1
educ_d1_2 educ_d2_2 educ_d3_2 educ_d4_2
educ_d1_3 educ_d2_3 educ_d3_3 educ_d4_3
educ_d1_4 educ_d2_4 educ_d3_4 educ_d4_4
educ_d1_5 educ_d2_5 educ_d3_5 educ_d4_5
educ_d1_6 educ_d2_6 educ_d3_6 educ_d4_6
income_d123_1 income_d4_1 income_d5_1 income_d6_1 income_d7_1 income_d8_1
income_d123_2 income_d4_2 income_d5_2 income_d6_2 income_d7_2 income_d8_2
income_d123_3 income_d4_3 income_d5_3 income_d6_3 income_d7_3 income_d8_3
income_d123_4 income_d4_4 income_d5_4 income_d6_4 income_d7_4 income_d8_4
income_d123_5 income_d4_5 income_d5_5 income_d6_5 income_d7_5 income_d8_5

```
income_d123_6 income_d4_6 income_d5_6 income_d6_6 income_d7_6 income_d8_6
purpose_d1_1 purpose_d2_1 purpose_d3_1
purpose_d1_2 purpose_d2_2 purpose_d3_2
purpose_d1_3 purpose_d2_3 purpose_d3_3
purpose_d1_4 purpose_d2_4 purpose_d3_4
purpose_d1_5 purpose_d2_5 purpose_d3_5
purpose_d1_6 purpose_d2_6 purpose_d3_6
pcomtoll_1 pcomtoll_3 pcomtoll_5
pwrktoll_1 pwrktoll_3 pwrktoll_5
pschtoll_1 pschtoll_3 pschtoll_5
prectoll_1 prectoll_3 prectoll_5
tollptoll_1 tollptoll_3 tollptoll_5
/ type=nlogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
        id pid;
        utility u(1, 1 3 5 @ 1) = unit_1 ttime_1 toll_1 gender_1 hhnum_1 motorveh_1
mlbin_1 fampool_1 tollpay_1
age_d1_1 age_d2_1 age_d3_1 age_d4_1 age_d5_1
ethni_d1_1 ethni_d2_1 ethni_d3_1
hhtyp_d1_1 hhtyp_d2_1 hhtyp_d3_1 hhtyp_d4_1 hhtyp_d5_1
occtyp_d1_1 occtyp_d2_1 occtyp_d3_1 occtyp_d4_1 occtyp_d5_1 occtyp_d6_1
occtyp_d7_1 occtyp_d8_1 occtyp_d9_1 occtyp_d10_1 occtyp_d11_1
educ_d1_1 educ_d2_1 educ_d3_1 educ_d4_1
income_d123_1 income_d4_1 income_d5_1 income_d6_1 income_d7_1 income_d8_1
purpose_d1_1 purpose_d2_1 purpose_d3_1
pcomtoll_1 pwrktoll_1 pschtoll_1 prectoll_1 tollptoll_1
unit_3 ttime_3 toll_3 gender_3 hhnum_3 motorveh_3 mlbin_3 fampool_3 tollpay_3
age_d1_3 age_d2_3 age_d3_3 age_d4_3 age_d5_3
ethni_d1_3 ethni_d2_3 ethni_d3_3
hhtyp_d1_3 hhtyp_d2_3 hhtyp_d3_3 hhtyp_d4_3 hhtyp_d5_3
occtyp_d1_3 occtyp_d2_3 occtyp_d3_3 occtyp_d4_3 occtyp_d5_3 occtyp_d6_3
occtyp_d7_3 occtyp_d8_3 occtyp_d9_3 occtyp_d10_3 occtyp_d11_3
educ_d1_3 educ_d2_3 educ_d3_3 educ_d4_3
income_d123_3 income_d4_3 income_d5_3 income_d6_3 income_d7_3 income_d8_3
purpose_d1_3 purpose_d2_3 purpose_d3_3
pcomtoll_3 pwrktoll_3 pschtoll_3 prectoll_3 tollptoll_3
unit_5 ttime_5 toll_5 gender_5 hhnum_5 motorveh_5 mlbin_5 fampool_5 tollpay_5
age_d1_5 age_d2_5 age_d3_5 age_d4_5 age_d5_5
ethni_d1_5 ethni_d2_5 ethni_d3_5
hhtyp_d1_5 hhtyp_d2_5 hhtyp_d3_5 hhtyp_d4_5 hhtyp_d5_5
occtyp_d1_5 occtyp_d2_5 occtyp_d3_5 occtyp_d4_5 occtyp_d5_5 occtyp_d6_5
occtyp_d7_5 occtyp_d8_5 occtyp_d9_5 occtyp_d10_5 occtyp_d11_5
educ_d1_5 educ_d2_5 educ_d3_5 educ_d4_5
income_d123_5 income_d4_5 income_d5_5 income_d6_5 income_d7_5 income_d8_5
purpose_d1_5 purpose_d2_5 purpose_d3_5
```

```
pcomtoll_5 pwrktoll_5 pschtoll_5 prectoll_5 tollptoll_5,

u(1, 2 4 6 @ 2) = unit_2 ttime_2 gender_2 hhnum_2 motorveh_2 mlbin_2 fampool_2
tollpay_2
age_d1_2 age_d2_2 age_d3_2 age_d4_2 age_d5_2
ethni_d1_2 ethni_d2_2 ethni_d3_2
hhtyp_d1_2 hhtyp_d2_2 hhtyp_d3_2 hhtyp_d4_2 hhtyp_d5_2
occtyp_d1_2 occtyp_d2_2 occtyp_d3_2 occtyp_d4_2 occtyp_d5_2 occtyp_d6_2
occtyp_d7_2 occtyp_d8_2 occtyp_d9_2 occtyp_d10_2 occtyp_d11_2
educ_d1_2 educ_d2_2 educ_d3_2 educ_d4_2
income_d123_2 income_d4_2 income_d5_2 income_d6_2 income_d7_2 income_d8_2
purpose_d1_2 purpose_d2_2 purpose_d3_2
unit_4 ttime_4 gender_4 hhnum_4 motorveh_4 mlbin_4 fampool_4 tollpay_4
age_d1_4 age_d2_4 age_d3_4 age_d4_4 age_d5_4
ethni_d1_4 ethni_d2_4 ethni_d3_4
hhtyp_d1_4 hhtyp_d2_4 hhtyp_d3_4 hhtyp_d4_4 hhtyp_d5_4
occtyp_d1_4 occtyp_d2_4 occtyp_d3_4 occtyp_d4_4 occtyp_d5_4 occtyp_d6_4
occtyp_d7_4 occtyp_d8_4 occtyp_d9_4 occtyp_d10_4 occtyp_d11_4
educ_d1_4 educ_d2_4 educ_d3_4 educ_d4_4
income_d123_4 income_d4_4 income_d5_4 income_d6_4 income_d7_4 income_d8_4
purpose_d1_4 purpose_d2_4 purpose_d3_4
unit_6 ttime_6 gender_6 hhnum_6 motorveh_6 mlbin_6 fampool_6 tollpay_6
age_d1_6 age_d2_6 age_d3_6 age_d4_6 age_d5_6
ethni_d1_6 ethni_d2_6 ethni_d3_6
hhtyp_d1_6 hhtyp_d2_6 hhtyp_d3_6 hhtyp_d4_6 hhtyp_d5_6
occtyp_d1_6 occtyp_d2_6 occtyp_d3_6 occtyp_d4_6 occtyp_d5_6 occtyp_d6_6
occtyp_d7_6 occtyp_d8_6 occtyp_d9_6 occtyp_d10_6 occtyp_d11_6
educ_d1_6 educ_d2_6 educ_d3_6 educ_d4_6
income_d123_6 income_d4_6 income_d5_6 income_d6_6 income_d7_6 income_d8_6
purpose_d1_6 purpose_d2_6 purpose_d3_6;

nest level(1) = (1 3 5 @ 1, 2 4 6 @ 2),
                        level(2) = (1 2 @ 1);

        output out=probdata pred=p;
*run;

/*********************************************/
*/
NESTED LOGIT ( ROUTE CHOICE -->MODE CHOICE (ALL VARS) )----ALT
SPECIFIC---NONSIGS REMOVED (STEP 1)
proc mdc data=newdata maxit=500;

        model decision = unit_1 unit_2 unit_3 /*unit_4*/ unit_5 /*unit_6*/
```

ttime_1 ttime_2 ttime_3 ttime_4 ttime_5 ttime_6
toll_1 toll_3 /*toll_5*/
gender_1 /*gender_2*/ gender_3 /*gender_4*/ gender_5 gender_6
hhnum_1 /*hhnum_2*/ hhnum_3 hhnum_4 /*hhnum_5*/ hhnum_6
motorveh_1 motorveh_2 motorveh_3 /*motorveh_4 /*motorveh_5*/ motorveh_6
mlbin_1 mlbin_2 mlbin_3 mlbin_4 mlbin_5 mlbin_6
fampool_1 fampool_2 fampool_3 fampool_4 fampool_5 /*fampool_6*/
tollpay_1 /*tollpay_2 /*tollpay_3 tollpay_4 /*tollpay_5*/ tollpay_6
age_d1_1 age_d2_1 age_d3_1 age_d4_1 age_d5_1
age_d1_2 age_d2_2 age_d3_2 age_d4_2 age_d5_2
age_d1_3 age_d2_3 /*age_d3_3 age_d4_3*/ age_d5_3
age_d1_4 age_d2_4 age_d3_4 age_d4_4 /*age_d5_4*/
age_d1_5 age_d2_5 age_d3_5 age_d4_5 age_d5_5
age_d1_6 age_d2_6 age_d3_6 age_d4_6 age_d5_6
ethni_d1_1 /*ethni_d2_1 ethni_d3_1*/
/*ethni_d1_2*/ ethni_d2_2 /*ethni_d3_2*/
ethni_d1_3 ethni_d2_3 ethni_d3_3
ethni_d1_4 ethni_d2_4 ethni_d3_4
/*ethni_d1_5 ethni_d2_5*/ ethni_d3_5
/*ethni_d1_6 ethni_d2_6*/ ethni_d3_6
hhtyp_d1_1 hhtyp_d2_1 hhtyp_d3_1 hhtyp_d4_1 hhtyp_d5_1
hhtyp_d1_2 /*hhtyp_d2_2*/ hhtyp_d3_2 hhtyp_d4_2 /*hhtyp_d5_2*/
hhtyp_d1_3 hhtyp_d2_3 /*hhtyp_d3_3*/ hhtyp_d4_3 hhtyp_d5_3
/*hhtyp_d1_4*/ hhtyp_d2_4 /*hhtyp_d3_4 hhtyp_d4_4*/ hhtyp_d5_4
hhtyp_d1_5 /*hhtyp_d2_5*/ hhtyp_d3_5 /*hhtyp_d4_5*/ hhtyp_d5_5
hhtyp_d1_6 hhtyp_d2_6 hhtyp_d3_6 /*hhtyp_d4_6*/ hhtyp_d5_6
occtyp_d1_1 occtyp_d2_1 occtyp_d3_1 occtyp_d4_1 occtyp_d5_1 occtyp_d6_1
occtyp_d7_1 occtyp_d8_1 occtyp_d9_1 occtyp_d10_1 occtyp_d11_1
/*occtyp_d1_2 occtyp_d2_2 occtyp_d3_2 occtyp_d4_2*/ occtyp_d5_2 /*occtyp_d6_2
occtyp_d7_2 occtyp_d8_2 occtyp_d9_2*/ occtyp_d10_2 occtyp_d11_2
occtyp_d1_3 /*occtyp_d2_3 occtyp_d3_3 occtyp_d4_3 occtyp_d5_3*/ occtyp_d6_3
occtyp_d7_3 occtyp_d8_3 occtyp_d9_3 occtyp_d10_3 occtyp_d11_3
occtyp_d1_4 occtyp_d2_4 occtyp_d3_4 occtyp_d4_4 occtyp_d5_4 occtyp_d6_4
/*occtyp_d7_4*/ occtyp_d8_4 occtyp_d9_4 occtyp_d10_4 occtyp_d11_4
occtyp_d1_5 /*occtyp_d2_5 occtyp_d3_5 occtyp_d4_5*/ occtyp_d5_5 /*occtyp_d6_5*/
occtyp_d7_5 /*occtyp_d8_5*/ occtyp_d9_5 /*occtyp_d10_5*/ occtyp_d11_5
occtyp_d1_6 occtyp_d2_6 occtyp_d3_6 occtyp_d4_6 /*occtyp_d5_6 occtyp_d6_6*/
occtyp_d7_6 /*occtyp_d8_6*/ occtyp_d9_6 occtyp_d10_6 occtyp_d11_6
/*educ_d1_1*/ educ_d2_1 educ_d3_1 educ_d4_1
/*educ_d1_2*/ educ_d2_2 educ_d3_2 educ_d4_2
educ_d1_3 educ_d2_3 educ_d3_3 educ_d4_3
educ_d1_4 educ_d2_4 /*educ_d3_4*/ educ_d4_4
educ_d1_5 educ_d2_5 educ_d3_5 educ_d4_5
educ_d1_6 educ_d2_6 educ_d3_6 educ_d4_6

```
income_d123_1 income_d4_1 income_d5_1 income_d6_1 income_d7_1 income_d8_1
income_d123_2 /*income_d4_2*/ income_d5_2 /*income_d6_2 income_d7_2
income_d8_2*/
income_d123_3 income_d4_3 income_d5_3 income_d6_3 income_d7_3 income_d8_3
/*income_d123_4*/ income_d4_4 income_d5_4 /*income_d6_4 income_d7_4
income_d8_4*/
income_d123_5 /*income_d4_5*/ income_d5_5 /*income_d6_5*/ income_d7_5
/*income_d8_5*/
income_d123_6 /*income_d4_6*/ income_d5_6 /*income_d6_6 income_d7_6*/
income_d8_6
purpose_d1_1 purpose_d2_1 purpose_d3_1
purpose_d1_2 purpose_d2_2 /*purpose_d3_2 */
purpose_d1_3 purpose_d2_3 purpose_d3_3
purpose_d1_4 purpose_d2_4 /*purpose_d3_4 */
purpose_d1_5 /*purpose_d2_5 purpose_d3_5 */
/*purpose_d1_6 purpose_d2_6*/ purpose_d3_6
pcomtoll_1 pcomtoll_3 /*pcomtoll_5*/
pwrktoll_1 pwrktoll_3 pwrktoll_5
/*pschtoll_1*/ pschtoll_3 /*pschtoll_5*/
prectoll_1 prectoll_3 /*prectoll_5*/
tollptoll_1 tollptoll_3 tollptoll_5
/ type=nlogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
        id pid;
        utility u(1, 1 3 5 @ 1) = unit_1 ttime_1 toll_1 gender_1 hhnum_1 motorveh_1
mlbin_1 fampool_1 tollpay_1
age_d1_1 age_d2_1 age_d3_1 age_d4_1 age_d5_1
ethni_d1_1 /*ethni_d2_1 ethni_d3_1*/
hhtyp_d1_1 hhtyp_d2_1 hhtyp_d3_1 hhtyp_d4_1 hhtyp_d5_1
occtyp_d1_1 occtyp_d2_1 occtyp_d3_1 occtyp_d4_1 occtyp_d5_1 occtyp_d6_1
occtyp_d7_1 occtyp_d8_1 occtyp_d9_1 occtyp_d10_1 occtyp_d11_1
/*educ_d1_1*/ educ_d2_1 educ_d3_1 educ_d4_1
income_d123_1 income_d4_1 income_d5_1 income_d6_1 income_d7_1 income_d8_1
purpose_d1_1 purpose_d2_1 purpose_d3_1
pcomtoll_1 pwrktoll_1 /*pschtoll_1*/ prectoll_1 tollptoll_1
unit_3 ttime_3 toll_3 gender_3 hhnum_3 motorveh_3 mlbin_3 fampool_3 /*tollpay_3
*/
age_d1_3 age_d2_3 /*age_d3_3 age_d4_3*/ age_d5_3
ethni_d1_3 ethni_d2_3 ethni_d3_3
hhtyp_d1_3 hhtyp_d2_3 /*hhtyp_d3_3*/ hhtyp_d4_3 hhtyp_d5_3
occtyp_d1_3 /*occtyp_d2_3 occtyp_d3_3 occtyp_d4_3 occtyp_d5_3*/ occtyp_d6_3
occtyp_d7_3 occtyp_d8_3 occtyp_d9_3 occtyp_d10_3 occtyp_d11_3
educ_d1_3 educ_d2_3 educ_d3_3 educ_d4_3
income_d123_3 income_d4_3 income_d5_3 income_d6_3 income_d7_3 income_d8_3
purpose_d1_3 purpose_d2_3 purpose_d3_3
```

pcomtoll_3 pwrktoll_3 pschtoll_3 prectoll_3 tollptoll_3
unit_5 ttime_5 /*toll_5*/ gender_5 /*hhnum_5 motorveh_5*/ mlbin_5 fampool_5
/*tollpay_5 */
age_d1_5 age_d2_5 age_d3_5 age_d4_5 age_d5_5
/*ethni_d1_5 ethni_d2_5*/ ethni_d3_5
hhtyp_d1_5 /*hhtyp_d2_5*/ hhtyp_d3_5 /*hhtyp_d4_5*/ hhtyp_d5_5
occtyp_d1_5 /*occtyp_d2_5 occtyp_d3_5 occtyp_d4_5*/ occtyp_d5_5 /*occtyp_d6_5*/
occtyp_d7_5 /*occtyp_d8_5*/ occtyp_d9_5 /*occtyp_d10_5*/ occtyp_d11_5
educ_d1_5 educ_d2_5 educ_d3_5 educ_d4_5
income_d123_5 /*income_d4_5*/ income_d5_5 /*income_d6_5*/ income_d7_5
/*income_d8_5*/
purpose_d1_5 /*purpose_d2_5 purpose_d3_5 */
/*pcomtoll_5*/ pwrktoll_5 /*pschtoll_5 prectoll_5*/ tollptoll_5,

u(1, 2 4 6 @ 2) = unit_2 ttime_2 /*gender_2 hhnum_2*/ motorveh_2 mlbin_2
fampool_2 /*tollpay_2 */
age_d1_2 age_d2_2 age_d3_2 age_d4_2 age_d5_2
/*ethni_d1_2*/ ethni_d2_2 /*ethni_d3_2*/
hhtyp_d1_2 /*hhtyp_d2_2*/ hhtyp_d3_2 hhtyp_d4_2 /*hhtyp_d5_2*/
/*occtyp_d1_2 occtyp_d2_2 occtyp_d3_2 occtyp_d4_2*/ occtyp_d5_2 /*occtyp_d6_2
occtyp_d7_2 occtyp_d8_2 occtyp_d9_2*/ occtyp_d10_2 occtyp_d11_2
/*educ_d1_2*/ educ_d2_2 educ_d3_2 educ_d4_2
income_d123_2 /*income_d4_2*/ income_d5_2 /*income_d6_2 income_d7_2
income_d8_2*/
purpose_d1_2 purpose_d2_2 /*purpose_d3_2 */
/*unit_4*/ ttime_4 /*gender_4*/ hhnum_4 /*motorveh_4*/ mlbin_4 fampool_4
/*tollpay_4*/
age_d1_4 age_d2_4 age_d3_4 age_d4_4 /*age_d5_4*/
ethni_d1_4 ethni_d2_4 ethni_d3_4
/*hhtyp_d1_4*/ hhtyp_d2_4 /*hhtyp_d3_4 hhtyp_d4_4*/ hhtyp_d5_4
occtyp_d1_4 occtyp_d2_4 occtyp_d3_4 occtyp_d4_4 occtyp_d5_4 occtyp_d6_4
/*occtyp_d7_4*/ occtyp_d8_4 occtyp_d9_4 occtyp_d10_4 occtyp_d11_4
educ_d1_4 educ_d2_4 /*educ_d3_4*/ educ_d4_4
/*income_d123_4*/ income_d4_4 income_d5_4 /*income_d6_4 income_d7_4
income_d8_4*/
purpose_d1_4 purpose_d2_4 /*purpose_d3_4 */
/*unit_6*/ ttime_6 gender_6 hhnum_6 motorveh_6 mlbin_6 /*fampool_6*/ tollpay_6
age_d1_6 age_d2_6 age_d3_6 age_d4_6 age_d5_6
/*ethni_d1_6 ethni_d2_6*/ ethni_d3_6
hhtyp_d1_6 hhtyp_d2_6 hhtyp_d3_6 /*hhtyp_d4_6*/ hhtyp_d5_6
occtyp_d1_6 occtyp_d2_6 occtyp_d3_6 occtyp_d4_6 /*occtyp_d5_6 occtyp_d6_6*/
occtyp_d7_6 /*occtyp_d8_6*/ occtyp_d9_6 occtyp_d10_6 occtyp_d11_6
educ_d1_6 educ_d2_6 educ_d3_6 educ_d4_6

```
income_d123_6 /*income_d4_6*/ income_d5_6 /*income_d6_6 income_d7_6*/
income_d8_6
/*purpose_d1_6 purpose_d2_6*/ purpose_d3_6;

nest level(1) = (1 3 5 @ 1, 2 4 6 @ 2),
                        level(2) = (1 2 @ 1);

        output out=probdata pred=p;
*run;

/*********************************************/
*NESTED LOGIT ( ROUTE CHOICE -->MODE CHOICE (SOV VS HOV/HOV2 VS
HOV3+) (ALL VARS) )
proc mdc data=newdata maxit=300;
        model decision = unit ttime toll
gender hhnum motorveh mlbin fampool tollpay
age_d1 age_d2 age_d3 age_d4 age_d5
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d3 hhtyp_d4 hhtyp_d5
occtyp_d1 occtyp_d2 occtyp_d3 occtyp_d4 occtyp_d5 occtyp_d6 occtyp_d7 occtyp_d8
occtyp_d9 occtyp_d10 occtyp_d11
educ_d1 educ_d2 educ_d3 educ_d4
income_d123 income_d4 income_d5 income_d6 income_d7 income_d8
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll
/ type=nlogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
        id pid;
        utility u(1,) = unit ttime toll
gender hhnum motorveh mlbin fampool tollpay
age_d1 age_d2 age_d3 age_d4 age_d5
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d3 hhtyp_d4 hhtyp_d5
occtyp_d1 occtyp_d2 occtyp_d3 occtyp_d4 occtyp_d5 occtyp_d6 occtyp_d7 occtyp_d8
occtyp_d9 occtyp_d10 occtyp_d11
educ_d1 educ_d2 educ_d3 educ_d4
income_d123 income_d4 income_d5 income_d6 income_d7 income_d8
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll;
        nest level(1) = (1 @ 1, 3 5 @ 2, 2 @ 3, 4 6 @ 4),
                        level(2) = (1 2 @ 1, 3 4 @ 2),
                        level(3) = (1 2 @ 1);

        output out=probdata pred=p;
*run;
```

```
*/
/*********************************************/
*TITLE "NESTED LOGIT ( ROUTE CHOICE -->MODE CHOICE (ML/SOV VS
HOV/HOV2 VS HOV3+) (ALL VARS) )"
proc mdc data=newdata maxit=300;
        model decision = unit ttime toll
gender hhnum motorveh mlbin fampool tollpay
age_d1 age_d2 age_d3 age_d4 age_d5
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d3 hhtyp_d4 hhtyp_d5
occtyp_d1 occtyp_d2 occtyp_d3 occtyp_d4 occtyp_d5 occtyp_d6 occtyp_d7 occtyp_d8
occtyp_d9 occtyp_d10 occtyp_d11
educ_d1 educ_d2 educ_d3 educ_d4
income_d123 income_d4 income_d5 income_d6 income_d7 income_d8
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll
/ type=nlogit choice=(mode_route 1 2 3 4 5 6) optmethod=qn covest=hess;
        id pid;
        utility u(1,) = unit ttime toll
gender hhnum motorveh mlbin fampool tollpay
age_d1 age_d2 age_d3 age_d4 age_d5
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d3 hhtyp_d4 hhtyp_d5
occtyp_d1 occtyp_d2 occtyp_d3 occtyp_d4 occtyp_d5 occtyp_d6 occtyp_d7 occtyp_d8
occtyp_d9 occtyp_d10 occtyp_d11
educ_d1 educ_d2 educ_d3 educ_d4
income_d123 income_d4 income_d5 income_d6 income_d7 income_d8
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll;
        nest level(1) = (1 @ 1, 3 5 @ 2, 2 4 6 @ 3),
                        level(2) = (1 2 @ 1, 3 @ 2),
                        level(3) = (1 2 @ 1);

        output out=probdata pred=p;
*run;
*/
/*********************************************/;
data dfw12;
set dfw;
if (SPCh=1 or SPCh=2) then output;
run;
data dfw34;
set dfw;
if (SPCh=3 or SPCh=4) then output;
```

```
run;
data dfw56;
set dfw;
if (SPCh=5 or SPCh=6) then output;
run;
data dfw135;
set dfw;
if (SPCh=1 or SPCh=3 or SPCh=5) then output;
run;
data newdata135(keep= pid id decision decision1 dec_mode dec_route
mode_route mode route
ttime
ttime_1 ttime_2 ttime_3 ttime_4 ttime_5 ttime_6
ttime_1_3_5 ttime_2_4_6
toll tollptoll
toll_1 toll_3 toll_5
toll_1_3_5
age gender ethnicity hhtype hhnum motorveh occtype educ income mlbin fampool
tollpay
unit
unit_r unit_m
unit_1 unit_2 unit_3 unit_4 unit_5 unit_6
unit_1_3_5 unit_2_4_6
age_d1 age_d234
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d345
occtyp_d1246 occtyp_d35 occtyp_d8 occtyp_d710
educ_d12
income_d1234 income_d56
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll

gender_1 gender_2 gender_3 gender_4 gender_5 gender_6
hhnum_1 hhnum_2 hhnum_3 hhnum_4 hhnum_5 hhnum_6
motorveh_1 motorveh_2 motorveh_3 motorveh_4 motorveh_5 motorveh_6
mlbin_1 mlbin_2 mlbin_3 mlbin_4 mlbin_5 mlbin_6
fampool_1 fampool_2 fampool_3 fampool_4 fampool_5 fampool_6
tollpay_1 tollpay_2 tollpay_3 tollpay_4 tollpay_5 tollpay_6
age_d1_1 age_d234_1
age_d1_2 age_d234_2
age_d1_3 age_d234_3
age_d1_4 age_d234_4
age_d1_5 age_d234_5
age_d1_6 age_d234_6
```

ethni_d1_1 ethni_d2_1 ethni_d3_1
ethni_d1_2 ethni_d2_2 ethni_d3_2
ethni_d1_3 ethni_d2_3 ethni_d3_3
ethni_d1_4 ethni_d2_4 ethni_d3_4
ethni_d1_5 ethni_d2_5 ethni_d3_5
ethni_d1_6 ethni_d2_6 ethni_d3_6
hhtyp_d1_1 hhtyp_d2_1 hhtyp_d345_1
hhtyp_d1_2 hhtyp_d2_2 hhtyp_d345_2
hhtyp_d1_3 hhtyp_d2_3 hhtyp_d345_3
hhtyp_d1_4 hhtyp_d2_4 hhtyp_d345_4
hhtyp_d1_5 hhtyp_d2_5 hhtyp_d345_5
hhtyp_d1_6 hhtyp_d2_6 hhtyp_d345_6
occtyp_d1246_1 occtyp_d35_1 occtyp_d8_1 occtyp_d710_1
occtyp_d1246_2 occtyp_d35_2 occtyp_d8_2 occtyp_d710_2
occtyp_d1246_3 occtyp_d35_3 occtyp_d8_3 occtyp_d710_3
occtyp_d1246_4 occtyp_d35_4 occtyp_d8_4 occtyp_d710_4
occtyp_d1246_5 occtyp_d35_5 occtyp_d8_5 occtyp_d710_5
occtyp_d1246_6 occtyp_d35_6 occtyp_d8_6 occtyp_d710_6
educ_d12_1
educ_d12_2
educ_d12_3
educ_d12_4
educ_d12_5
educ_d12_6
income_d1234_1 income_d56_1
income_d1234_2 income_d56_2
income_d1234_3 income_d56_3
income_d1234_4 income_d56_4
income_d1234_5 income_d56_5
income_d1234_6 income_d56_6
purpose_d1_1 purpose_d2_1 purpose_d3_1
purpose_d1_2 purpose_d2_2 purpose_d3_2
purpose_d1_3 purpose_d2_3 purpose_d3_3
purpose_d1_4 purpose_d2_4 purpose_d3_4
purpose_d1_5 purpose_d2_5 purpose_d3_5
purpose_d1_6 purpose_d2_6 purpose_d3_6
pcomtoll_1 pcomtoll_3 pcomtoll_5
pwrktoll_1 pwrktoll_3 pwrktoll_5
pschtoll_1 pschtoll_3 pschtoll_5
prectoll_1 prectoll_3 prectoll_5
tollptoll_1 tollptoll_3 tollptoll_5
);

set dfw135;

```
*       set q_combined;
        array tvec{6} tt1 - tt6;
        array tollvec{6} toll1 - toll6;

        retain pid 0;
        pid + 1;
        do i = 1 to 6 by 2;
                id = surveyid;
                if estim=1 then decision = ( SPCh = i );
                if estim=0 then decision = .;
                if estim=0 then decision1 = ( SPCh = i );
                mode_route = i;
                if (i=1 or i=2) then mode=1;
                if (i=3 or i=4) then mode=2;
                if (i=5 or i=6) then mode=3;
                if (i=1 or i=3 or i=5) then route=1;
                if (i=2 or i=4 or i=6) then route=2;

                ttime = tvec{i};
                toll = tollvec{i};
                if (i=2 or i=4 or i=6) then toll=0;
*******************************;
*INTERACTIONS;
*******************************;
                pcomtoll=0;    pwrktoll=0;    pschtoll=0;     prectoll=0;
                if purpagg=1 then pcomtoll = toll;
                if purpagg=3 then pwrktoll = toll;
                if purpagg=4 then pschtoll = toll;
                if purpagg=2 then prectoll = toll;

                tollptoll=0;
                if tollpay=1 then tollptoll = toll;
*******************************;
*DUMMIES;
*******************************;
                age_d1=0; age_d234=0;
                if age=1 then age_d1=1;
                if (age=2 or age=3 or age=4) then age_d234=1;
                ethni_d1=0;    ethni_d2=0;    ethni_d3=0;
                if ethnicity=1 then ethni_d1=1;
                if ethnicity=2 then ethni_d2=1;
                if ethnicity=3 then ethni_d3=1;
                hhtyp_d1=0;    hhtyp_d2=0;    hhtyp_d345=0;
                if hhtype=1 then hhtyp_d1=1;
```

```
                if hhtype=2 then hhtyp_d2=1;
                if (hhtype=3 or hhtype=4 or hhtype=5) then hhtyp_d345=1;
                occtyp_d1246=0; occtyp_d35=0; occtyp_d8=0; occtyp_d710=0;
                if (occtype=1 or occtype=2 or occtype=4 or occtype=6) then
occtyp_d1246=1;
                if (occtype=3 or occtype=5) then occtyp_d35=1;
                if occtype=8 then occtyp_d8=1;
                if (occtype=7 or occtype=10) then occtyp_d710=1;
                educ_d12=0;
                if (educ=1 or educ=2) then educ_d12=1;
                income_d1234=0; income_d56=0;
                if (income=1 or income=2 or income=3 or income=4) then
income_d1234=1;
                if (income=5 or income=6) then income_d56=1;
                purpose_d1=0; purpose_d2=0;         purpose_d3=0;
                if purpagg=1 then purpose_d1 = 1;
                if purpagg=3 then purpose_d2 = 1;
                if purpagg=4 then purpose_d3 = 1;


*****************************;
*MODE SPECIFICS;
*****************************;
                ttime_1=0;              toll_1=0;
                ttime_2=0;
                ttime_3=0;              toll_3=0;
                ttime_4=0;
                ttime_5=0;              toll_5=0;
                ttime_6=0;
                if (i EQ 1) then ttime_1=ttime;             if (i EQ 1) then toll_1=toll;
                if (i EQ 2) then ttime_2=ttime;
                if (i EQ 3) then ttime_3=ttime;             if (i EQ 3) then toll_3=toll;
                if (i EQ 4) then ttime_4=ttime;
                if (i EQ 5) then ttime_5=ttime;             if (i EQ 5) then toll_5=toll;
                if (i EQ 6) then ttime_6=ttime;
                unit = 1; unit_r=1; unit_m=1;
                unit_1 = 0;
                unit_2 = 0;
                unit_3 = 0;
                unit_4 = 0;
                unit_5 = 0;
                unit_6 = 0;

                if (i EQ 1) then unit_1=1;
```

```
                if (i EQ 2) then unit_2=1;
                if (i EQ 3) then unit_3=1;
                if (i EQ 4) then unit_4=1;
                if (i EQ 5) then unit_5=1;
                if (i EQ 6) then unit_6=1;
*********************************;
                toll_1_3_5 = tollvec{i};
                if (i=2 or i=4 or i=6) then toll_1_3_5=0;
                ttime_1_3_5=0;
                ttime_2_4_6=0;
                if (i=1 or i=3 or i=5) then ttime_1_3_5=ttime;
                if (i=2 or i=4 or i=6) then ttime_2_4_6=ttime;
                unit_1_3_5 = 0;
                unit_2_4_6 = 0;
                if (i=1 or i=3 or i=5) then unit_1_3_5=1;
                if (i=2 or i=4 or i=6) then unit_2_4_6=1;
*********************************;
                if (i EQ 1) then unit_1=1;
                if (i EQ 2) then unit_2=1;
                if (i EQ 3) then unit_3=1;
                if (i EQ 4) then unit_4=1;
                if (i EQ 5) then unit_5=1;
                if (i EQ 6) then unit_6=1;
*********************************;
                gender_1=0; gender_2=0; gender_3=0; gender_4=0; gender_5=0;
gender_6=0;
                if i=1 then gender_1=gender;
                if i=2 then gender_2=gender;
                if i=3 then gender_3=gender;
                if i=4 then gender_4=gender;
                if i=5 then gender_5=gender;
                if i=6 then gender_6=gender;
                hhnum_1=0; hhnum_2=0; hhnum_3=0; hhnum_4=0; hhnum_5=0;
hhnum_6=0;
                if i=1 then hhnum_1=hhnum;
                if i=2 then hhnum_2=hhnum;
                if i=3 then hhnum_3=hhnum;
                if i=4 then hhnum_4=hhnum;
                if i=5 then hhnum_5=hhnum;
                if i=6 then hhnum_6=hhnum;
                motorveh_1=0; motorveh_2=0; motorveh_3=0; motorveh_4=0;
motorveh_5=0; motorveh_6=0;
                if i=1 then motorveh_1=motorveh;
                if i=2 then motorveh_2=motorveh;
```

```
                if i=3 then motorveh_3=motorveh;
                if i=4 then motorveh_4=motorveh;
                if i=5 then motorveh_5=motorveh;
                if i=6 then motorveh_6=motorveh;
                mlbin_1=0; mlbin_2=0; mlbin_3=0; mlbin_4=0; mlbin_5=0; mlbin_6=0;
                if i=1 then mlbin_1=mlbin;
                if i=2 then mlbin_2=mlbin;
                if i=3 then mlbin_3=mlbin;
                if i=4 then mlbin_4=mlbin;
                if i=5 then mlbin_5=mlbin;
                if i=6 then mlbin_6=mlbin;
                fampool_1=0; fampool_2=0; fampool_3=0; fampool_4=0; fampool_5=0;
fampool_6=0;
                if i=1 then fampool_1=fampool;
                if i=2 then fampool_2=fampool;
                if i=3 then fampool_3=fampool;
                if i=4 then fampool_4=fampool;
                if i=5 then fampool_5=fampool;
                if i=6 then fampool_6=fampool;
                tollpay_1=0; tollpay_2=0; tollpay_3=0; tollpay_4=0; tollpay_5=0;
tollpay_6=0;
                if i=1 then tollpay_1=tollpay;
                if i=2 then tollpay_2=tollpay;
                if i=3 then tollpay_3=tollpay;
                if i=4 then tollpay_4=tollpay;
                if i=5 then tollpay_5=tollpay;
                if i=6 then tollpay_6=tollpay;
                age_d1_1=0; age_d1_2=0; age_d1_3=0; age_d1_4=0; age_d1_5=0;
age_d1_6=0;
                age_d234_1=0; age_d234_2=0; age_d234_3=0; age_d234_4=0;
age_d234_5=0; age_d234_6=0;
                if i=1 then age_d1_1=age_d1;
                if i=1 then age_d234_1=age_d234;
                if i=2 then age_d1_2=age_d1;
                if i=2 then age_d234_2=age_d234;
                if i=3 then age_d1_3=age_d1;
                if i=3 then age_d234_3=age_d234;
                if i=4 then age_d1_4=age_d1;
                if i=4 then age_d234_4=age_d234;
                if i=5 then age_d1_5=age_d1;
                if i=5 then age_d234_5=age_d234;
                if i=6 then age_d1_6=age_d1;
                if i=6 then age_d234_6=age_d234;
```

```
                ethni_d1_1=0; ethni_d1_2=0; ethni_d1_3=0; ethni_d1_4=0;
ethni_d1_5=0; ethni_d1_6=0;
                ethni_d2_1=0; ethni_d2_2=0; ethni_d2_3=0; ethni_d2_4=0;
ethni_d2_5=0; ethni_d2_6=0;
                ethni_d3_1=0; ethni_d3_2=0; ethni_d3_3=0; ethni_d3_4=0;
ethni_d3_5=0; ethni_d3_6=0;
                if i=1 then ethni_d1_1=ethni_d1;
                if i=1 then ethni_d2_1=ethni_d2;
                if i=1 then ethni_d3_1=ethni_d3;
                if i=2 then ethni_d1_2=ethni_d1;
                if i=2 then ethni_d2_2=ethni_d2;
                if i=2 then ethni_d3_2=ethni_d3;
                if i=3 then ethni_d1_3=ethni_d1;
                if i=3 then ethni_d2_3=ethni_d2;
                if i=3 then ethni_d3_3=ethni_d3;
                if i=4 then ethni_d1_4=ethni_d1;
                if i=4 then ethni_d2_4=ethni_d2;
                if i=4 then ethni_d3_4=ethni_d3;
                if i=5 then ethni_d1_5=ethni_d1;
                if i=5 then ethni_d2_5=ethni_d2;
                if i=5 then ethni_d3_5=ethni_d3;
                if i=6 then ethni_d1_6=ethni_d1;
                if i=6 then ethni_d2_6=ethni_d2;
                if i=6 then ethni_d3_6=ethni_d3;
                hhtyp_d1_1=0; hhtyp_d1_2=0; hhtyp_d1_3=0; hhtyp_d1_4=0;
hhtyp_d1_5=0; hhtyp_d1_6=0;
                hhtyp_d2_1=0; hhtyp_d2_2=0; hhtyp_d2_3=0; hhtyp_d2_4=0;
hhtyp_d2_5=0; hhtyp_d2_6=0;
                hhtyp_d345_1=0; hhtyp_d345_2=0; hhtyp_d345_3=0; hhtyp_d345_4=0;
hhtyp_d345_5=0; hhtyp_d345_6=0;
                if i=1 then hhtyp_d1_1=hhtyp_d1;
                if i=1 then hhtyp_d2_1=hhtyp_d2;
                if i=1 then hhtyp_d345_1=hhtyp_d345;
                if i=2 then hhtyp_d1_2=hhtyp_d1;
                if i=2 then hhtyp_d2_2=hhtyp_d2;
                if i=2 then hhtyp_d345_2=hhtyp_d345;
                if i=3 then hhtyp_d1_3=hhtyp_d1;
                if i=3 then hhtyp_d2_3=hhtyp_d2;
                if i=3 then hhtyp_d345_3=hhtyp_d345;
                if i=4 then hhtyp_d1_4=hhtyp_d1;
                if i=4 then hhtyp_d2_4=hhtyp_d2;
                if i=4 then hhtyp_d345_4=hhtyp_d345;
                if i=5 then hhtyp_d1_5=hhtyp_d1;
                if i=5 then hhtyp_d2_5=hhtyp_d2;
```

```
                if i=5 then hhtyp_d345_5=hhtyp_d345;
                if i=6 then hhtyp_d1_6=hhtyp_d1;
                if i=6 then hhtyp_d2_6=hhtyp_d2;
                if i=6 then hhtyp_d345_6=hhtyp_d345;
                occtyp_d1246_1=0; occtyp_d1246_2=0; occtyp_d1246_3=0;
occtyp_d1246_4=0; occtyp_d1246_5=0; occtyp_d1246_6=0;
                occtyp_d35_1=0; occtyp_d35_2=0; occtyp_d35_3=0; occtyp_d35_4=0;
occtyp_d35_5=0; occtyp_d35_6=0;
                occtyp_d8_1=0; occtyp_d8_2=0; occtyp_d8_3=0; occtyp_d8_4=0;
occtyp_d8_5=0; occtyp_d8_6=0;
                occtyp_d710_1=0; occtyp_d710_2=0; occtyp_d710_3=0;
occtyp_d710_4=0; occtyp_d710_5=0; occtyp_d710_6=0;
                if i=1 then occtyp_d1246_1=occtyp_d1246;
                if i=1 then occtyp_d35_1=occtyp_d35;
                if i=1 then occtyp_d8_1=occtyp_d8;
                if i=1 then occtyp_d710_1=occtyp_d710;
                if i=2 then occtyp_d1246_2=occtyp_d1246;
                if i=2 then occtyp_d35_2=occtyp_d35;
                if i=2 then occtyp_d8_2=occtyp_d8;
                if i=2 then occtyp_d710_2=occtyp_d710;
                if i=3 then occtyp_d1246_3=occtyp_d1246;
                if i=3 then occtyp_d35_3=occtyp_d35;
                if i=3 then occtyp_d8_3=occtyp_d8;
                if i=3 then occtyp_d710_3=occtyp_d710;
                if i=4 then occtyp_d1246_4=occtyp_d1246;
                if i=4 then occtyp_d35_4=occtyp_d35;
                if i=4 then occtyp_d8_4=occtyp_d8;
                if i=4 then occtyp_d710_4=occtyp_d710;
                if i=5 then occtyp_d1246_5=occtyp_d1246;
                if i=5 then occtyp_d35_5=occtyp_d35;
                if i=5 then occtyp_d8_5=occtyp_d8;
                if i=5 then occtyp_d710_5=occtyp_d710;
                if i=6 then occtyp_d1246_6=occtyp_d1246;
                if i=6 then occtyp_d35_6=occtyp_d35;
                if i=6 then occtyp_d8_6=occtyp_d8;
                if i=6 then occtyp_d710_6=occtyp_d710;
                educ_d12_1=0; educ_d12_2=0; educ_d12_3=0; educ_d12_4=0;
educ_d12_5=0; educ_d12_6=0;
                if i=1 then educ_d12_1=educ_d12;
                if i=2 then educ_d12_2=educ_d12;
                if i=3 then educ_d12_3=educ_d12;
                if i=4 then educ_d12_4=educ_d12;
                if i=5 then educ_d12_5=educ_d12;
                if i=6 then educ_d12_6=educ_d12;
```

```
                income_d1234_1=0; income_d1234_2=0; income_d1234_3=0;
income_d1234_4=0; income_d1234_5=0; income_d1234_6=0;
                income_d56_1=0; income_d56_2=0; income_d56_3=0;
income_d56_4=0; income_d56_5=0; income_d56_6=0;
                if i=1 then income_d1234_1=income_d1234;
                if i=1 then income_d56_1=income_d56;
                if i=2 then income_d1234_2=income_d1234;
                if i=2 then income_d56_2=income_d56;
                if i=3 then income_d1234_3=income_d1234;
                if i=3 then income_d56_3=income_d56;
                if i=4 then income_d1234_4=income_d1234;
                if i=4 then income_d56_4=income_d56;
                if i=5 then income_d1234_5=income_d1234;
                if i=5 then income_d56_5=income_d56;
                if i=6 then income_d1234_6=income_d1234;
                if i=6 then income_d56_6=income_d56;
                purpose_d1_1=0; purpose_d1_2=0; purpose_d1_3=0; purpose_d1_4=0;
purpose_d1_5=0; purpose_d1_6=0;
                purpose_d2_1=0; purpose_d2_2=0; purpose_d2_3=0; purpose_d2_4=0;
purpose_d2_5=0; purpose_d2_6=0;
                purpose_d3_1=0; purpose_d3_2=0; purpose_d3_3=0; purpose_d3_4=0;
purpose_d3_5=0; purpose_d3_6=0;
                if i=1 then purpose_d1_1=purpose_d1;
                if i=1 then purpose_d2_1=purpose_d2;
                if i=1 then purpose_d3_1=purpose_d3;
                if i=2 then purpose_d1_2=purpose_d1;
                if i=2 then purpose_d2_2=purpose_d2;
                if i=2 then purpose_d3_2=purpose_d3;
                if i=3 then purpose_d1_3=purpose_d1;
                if i=3 then purpose_d2_3=purpose_d2;
                if i=3 then purpose_d3_3=purpose_d3;
                if i=4 then purpose_d1_4=purpose_d1;
                if i=4 then purpose_d2_4=purpose_d2;
                if i=4 then purpose_d3_4=purpose_d3;
                if i=5 then purpose_d1_5=purpose_d1;
                if i=5 then purpose_d2_5=purpose_d2;
                if i=5 then purpose_d3_5=purpose_d3;
                if i=6 then purpose_d1_6=purpose_d1;
                if i=6 then purpose_d2_6=purpose_d2;
                if i=6 then purpose_d3_6=purpose_d3;
*****************************;
                pcomtoll_1=0; pcomtoll_3=0; pcomtoll_5=0;
                if i=1 then pcomtoll_1=pcomtoll;
                if i=3 then pcomtoll_3=pcomtoll;
```

```
                if i=5 then pcomtoll_5=pcomtoll;
                pwrktoll_1=0; pwrktoll_3=0; pwrktoll_5=0;
                if i=1 then pwrktoll_1=pwrktoll;
                if i=3 then pwrktoll_3=pwrktoll;
                if i=5 then pwrktoll_5=pwrktoll;
                pschtoll_1=0; pschtoll_3=0; pschtoll_5=0;
                if i=1 then pschtoll_1=pschtoll;
                if i=3 then pschtoll_3=pschtoll;
                if i=5 then pschtoll_5=pschtoll;
                prectoll_1=0; prectoll_3=0; prectoll_5=0;
                if i=1 then prectoll_1=prectoll;
                if i=3 then prectoll_3=prectoll;
                if i=5 then prectoll_5=prectoll;
                tollptoll_1=0; tollptoll_3=0; tollptoll_5=0;
                if i=1 then tollptoll_1=tollptoll;
                if i=3 then tollptoll_3=tollptoll;
                if i=5 then tollptoll_5=tollptoll;
**************************************************************************
*;
                output;
        end;
*proc print data=newdata135;
run;
************************************************************;
* MULTINOMIAL LOGIT ML alts---GENERIC---PARAMETERS (ALL VARS)
proc mdc data=newdata135 maxit=300;
        model decision = unit ttime toll
gender hhnum motorveh mlbin fampool tollpay
age_d1 age_d234
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d345
occtyp_d1246 occtyp_d35 occtyp_d8 occtyp_d710
educ_d12
income_d1234 income_d56
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll
/ type=clogit choice=(mode_route 1 3 5) optmethod=qn covest=hess;
        id pid;
        output out=probdata pred=p;
*proc print data=probdata;
*run;
************************************************************;
data dfw246;
set dfw;
```

```
if (SPCh=2 or SPCh=4 or SPCh=6) then output;
run;
data newdata246(keep= pid id decision decision1 dec_mode dec_route
mode_route mode route
ttime
ttime_1 ttime_2 ttime_3 ttime_4 ttime_5 ttime_6
ttime_1_3_5 ttime_2_4_6
toll tollptoll
toll_1 toll_3 toll_5
toll_1_3_5
age gender ethnicity hhtype hhnum motorveh occtype educ income mlbin fampool
tollpay
unit
unit_r unit_m
unit_1 unit_2 unit_3 unit_4 unit_5 unit_6
unit_1_3_5 unit_2_4_6
age_d1 age_d234
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d345
occtyp_d1246 occtyp_d35 occtyp_d8 occtyp_d710
educ_d12
income_d1234 income_d56
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll

gender_1 gender_2 gender_3 gender_4 gender_5 gender_6
hhnum_1 hhnum_2 hhnum_3 hhnum_4 hhnum_5 hhnum_6
motorveh_1 motorveh_2 motorveh_3 motorveh_4 motorveh_5 motorveh_6
mlbin_1 mlbin_2 mlbin_3 mlbin_4 mlbin_5 mlbin_6
fampool_1 fampool_2 fampool_3 fampool_4 fampool_5 fampool_6
tollpay_1 tollpay_2 tollpay_3 tollpay_4 tollpay_5 tollpay_6
age_d1_1 age_d234_1
age_d1_2 age_d234_2
age_d1_3 age_d234_3
age_d1_4 age_d234_4
age_d1_5 age_d234_5
age_d1_6 age_d234_6
ethni_d1_1 ethni_d2_1 ethni_d3_1
ethni_d1_2 ethni_d2_2 ethni_d3_2
ethni_d1_3 ethni_d2_3 ethni_d3_3
ethni_d1_4 ethni_d2_4 ethni_d3_4
ethni_d1_5 ethni_d2_5 ethni_d3_5
ethni_d1_6 ethni_d2_6 ethni_d3_6
hhtyp_d1_1 hhtyp_d2_1 hhtyp_d345_1
```

```
hhtyp_d1_2 hhtyp_d2_2 hhtyp_d345_2
hhtyp_d1_3 hhtyp_d2_3 hhtyp_d345_3
hhtyp_d1_4 hhtyp_d2_4 hhtyp_d345_4
hhtyp_d1_5 hhtyp_d2_5 hhtyp_d345_5
hhtyp_d1_6 hhtyp_d2_6 hhtyp_d345_6
occtyp_d1246_1 occtyp_d35_1 occtyp_d8_1 occtyp_d710_1
occtyp_d1246_2 occtyp_d35_2 occtyp_d8_2 occtyp_d710_2
occtyp_d1246_3 occtyp_d35_3 occtyp_d8_3 occtyp_d710_3
occtyp_d1246_4 occtyp_d35_4 occtyp_d8_4 occtyp_d710_4
occtyp_d1246_5 occtyp_d35_5 occtyp_d8_5 occtyp_d710_5
occtyp_d1246_6 occtyp_d35_6 occtyp_d8_6 occtyp_d710_6
educ_d12_1
educ_d12_2
educ_d12_3
educ_d12_4
educ_d12_5
educ_d12_6
income_d1234_1 income_d56_1
income_d1234_2 income_d56_2
income_d1234_3 income_d56_3
income_d1234_4 income_d56_4
income_d1234_5 income_d56_5
income_d1234_6 income_d56_6
purpose_d1_1 purpose_d2_1 purpose_d3_1
purpose_d1_2 purpose_d2_2 purpose_d3_2
purpose_d1_3 purpose_d2_3 purpose_d3_3
purpose_d1_4 purpose_d2_4 purpose_d3_4
purpose_d1_5 purpose_d2_5 purpose_d3_5
purpose_d1_6 purpose_d2_6 purpose_d3_6
pcomtoll_1 pcomtoll_3 pcomtoll_5
pwrktoll_1 pwrktoll_3 pwrktoll_5
pschtoll_1 pschtoll_3 pschtoll_5
prectoll_1 prectoll_3 prectoll_5
tollptoll_1 tollptoll_3 tollptoll_5
);

        set dfw246;
*       set q_combined;
        array tvec{6} tt1 - tt6;
        array tollvec{6} toll1 - toll6;

        retain pid 0;
        pid + 1;
        do i = 2 to 6 by 2;
```

```
                id = surveyid;
                if estim=1 then decision = ( SPCh = i );
                if estim=0 then decision = .;
                if estim=0 then decision1 = ( SPCh = i );
                mode_route = i;
                if (i=1 or i=2) then mode=1;
                if (i=3 or i=4) then mode=2;
                if (i=5 or i=6) then mode=3;
                if (i=1 or i=3 or i=5) then route=1;
                if (i=2 or i=4 or i=6) then route=2;

                ttime = tvec{i};
                toll = tollvec{i};
                if (i=2 or i=4 or i=6) then toll=0;
*****************************;
*INTERACTIONS;
*****************************;
                pcomtoll=0;    pwrktoll=0;    pschtoll=0;    prectoll=0;
                if purpagg=1 then pcomtoll = toll;
                if purpagg=3 then pwrktoll = toll;
                if purpagg=4 then pschtoll = toll;
                if purpagg=2 then prectoll = toll;

                tollptoll=0;
                if tollpay=1 then tollptoll = toll;
*****************************;
*DUMMIES;
*****************************;
                age_d1=0; age_d234=0;
                if age=1 then age_d1=1;
                if (age=2 or age=3 or age=4) then age_d234=1;
                ethni_d1=0;    ethni_d2=0;    ethni_d3=0;
                if ethnicity=1 then ethni_d1=1;
                if ethnicity=2 then ethni_d2=1;
                if ethnicity=3 then ethni_d3=1;
                hhtyp_d1=0;   hhtyp_d2=0;   hhtyp_d345=0;
                if hhtype=1 then hhtyp_d1=1;
                if hhtype=2 then hhtyp_d2=1;
                if (hhtype=3 or hhtype=4 or hhtype=5) then hhtyp_d345=1;
                occtyp_d1246=0; occtyp_d35=0; occtyp_d8=0; occtyp_d710=0;
                if (occtype=1 or occtype=2 or occtype=4 or occtype=6) then
occtyp_d1246=1;
                if (occtype=3 or occtype=5) then occtyp_d35=1;
                if occtype=8 then occtyp_d8=1;
```

```
                if (occtype=7 or occtype=10) then occtyp_d710=1;
                educ_d12=0;
                if (educ=1 or educ=2) then educ_d12=1;
                income_d1234=0; income_d56=0;
                if (income=1 or income=2 or income=3 or income=4) then
income_d1234=1;
                if (income=5 or income=6) then income_d56=1;
                purpose_d1=0; purpose_d2=0;         purpose_d3=0;
                if purpagg=1 then purpose_d1 = 1;
                if purpagg=3 then purpose_d2 = 1;
                if purpagg=4 then purpose_d3 = 1;


*****************************;
*MODE SPECIFICS;
*****************************;
                ttime_1=0;              toll_1=0;
                ttime_2=0;
                ttime_3=0;              toll_3=0;
                ttime_4=0;
                ttime_5=0;              toll_5=0;
                ttime_6=0;
                if (i EQ 1) then ttime_1=ttime;                 if (i EQ 1) then toll_1=toll;
                if (i EQ 2) then ttime_2=ttime;
                if (i EQ 3) then ttime_3=ttime;                 if (i EQ 3) then toll_3=toll;
                if (i EQ 4) then ttime_4=ttime;
                if (i EQ 5) then ttime_5=ttime;                 if (i EQ 5) then toll_5=toll;
                if (i EQ 6) then ttime_6=ttime;
                unit = 1; unit_r=1; unit_m=1;
                unit_1 = 0;
                unit_2 = 0;
                unit_3 = 0;
                unit_4 = 0;
                unit_5 = 0;
                unit_6 = 0;

                if (i EQ 1) then unit_1=1;
                if (i EQ 2) then unit_2=1;
                if (i EQ 3) then unit_3=1;
                if (i EQ 4) then unit_4=1;
                if (i EQ 5) then unit_5=1;
                if (i EQ 6) then unit_6=1;
*******************************;
                toll_1_3_5 = tollvec{i};
```

```
                if (i=2 or i=4 or i=6) then toll_1_3_5=0;
                ttime_1_3_5=0;
                ttime_2_4_6=0;
                if (i=1 or i=3 or i=5) then ttime_1_3_5=ttime;
                if (i=2 or i=4 or i=6) then ttime_2_4_6=ttime;
                unit_1_3_5 = 0;
                unit_2_4_6 = 0;
                if (i=1 or i=3 or i=5) then unit_1_3_5=1;
                if (i=2 or i=4 or i=6) then unit_2_4_6=1;
********************************;
                if (i EQ 1) then unit_1=1;
                if (i EQ 2) then unit_2=1;
                if (i EQ 3) then unit_3=1;
                if (i EQ 4) then unit_4=1;
                if (i EQ 5) then unit_5=1;
                if (i EQ 6) then unit_6=1;
********************************;
                gender_1=0; gender_2=0; gender_3=0; gender_4=0; gender_5=0;
gender_6=0;
                if i=1 then gender_1=gender;
                if i=2 then gender_2=gender;
                if i=3 then gender_3=gender;
                if i=4 then gender_4=gender;
                if i=5 then gender_5=gender;
                if i=6 then gender_6=gender;
                hhnum_1=0; hhnum_2=0; hhnum_3=0; hhnum_4=0; hhnum_5=0;
hhnum_6=0;
                if i=1 then hhnum_1=hhnum;
                if i=2 then hhnum_2=hhnum;
                if i=3 then hhnum_3=hhnum;
                if i=4 then hhnum_4=hhnum;
                if i=5 then hhnum_5=hhnum;
                if i=6 then hhnum_6=hhnum;
                motorveh_1=0; motorveh_2=0; motorveh_3=0; motorveh_4=0;
motorveh_5=0; motorveh_6=0;
                if i=1 then motorveh_1=motorveh;
                if i=2 then motorveh_2=motorveh;
                if i=3 then motorveh_3=motorveh;
                if i=4 then motorveh_4=motorveh;
                if i=5 then motorveh_5=motorveh;
                if i=6 then motorveh_6=motorveh;
                mlbin_1=0; mlbin_2=0; mlbin_3=0; mlbin_4=0; mlbin_5=0; mlbin_6=0;
                if i=1 then mlbin_1=mlbin;
                if i=2 then mlbin_2=mlbin;
```

```
                if i=3 then mlbin_3=mlbin;
                if i=4 then mlbin_4=mlbin;
                if i=5 then mlbin_5=mlbin;
                if i=6 then mlbin_6=mlbin;
                fampool_1=0; fampool_2=0; fampool_3=0; fampool_4=0; fampool_5=0;
fampool_6=0;
                if i=1 then fampool_1=fampool;
                if i=2 then fampool_2=fampool;
                if i=3 then fampool_3=fampool;
                if i=4 then fampool_4=fampool;
                if i=5 then fampool_5=fampool;
                if i=6 then fampool_6=fampool;
                tollpay_1=0; tollpay_2=0; tollpay_3=0; tollpay_4=0; tollpay_5=0;
tollpay_6=0;
                if i=1 then tollpay_1=tollpay;
                if i=2 then tollpay_2=tollpay;
                if i=3 then tollpay_3=tollpay;
                if i=4 then tollpay_4=tollpay;
                if i=5 then tollpay_5=tollpay;
                if i=6 then tollpay_6=tollpay;
                age_d1_1=0; age_d1_2=0; age_d1_3=0; age_d1_4=0; age_d1_5=0;
age_d1_6=0;
                age_d234_1=0; age_d234_2=0; age_d234_3=0; age_d234_4=0;
age_d234_5=0; age_d234_6=0;
                if i=1 then age_d1_1=age_d1;
                if i=1 then age_d234_1=age_d234;
                if i=2 then age_d1_2=age_d1;
                if i=2 then age_d234_2=age_d234;
                if i=3 then age_d1_3=age_d1;
                if i=3 then age_d234_3=age_d234;
                if i=4 then age_d1_4=age_d1;
                if i=4 then age_d234_4=age_d234;
                if i=5 then age_d1_5=age_d1;
                if i=5 then age_d234_5=age_d234;
                if i=6 then age_d1_6=age_d1;
                if i=6 then age_d234_6=age_d234;
                ethni_d1_1=0; ethni_d1_2=0; ethni_d1_3=0; ethni_d1_4=0;
ethni_d1_5=0; ethni_d1_6=0;
                ethni_d2_1=0; ethni_d2_2=0; ethni_d2_3=0; ethni_d2_4=0;
ethni_d2_5=0; ethni_d2_6=0;
                ethni_d3_1=0; ethni_d3_2=0; ethni_d3_3=0; ethni_d3_4=0;
ethni_d3_5=0; ethni_d3_6=0;
                if i=1 then ethni_d1_1=ethni_d1;
                if i=1 then ethni_d2_1=ethni_d2;
```

```
                if i=1 then ethni_d3_1=ethni_d3;
                if i=2 then ethni_d1_2=ethni_d1;
                if i=2 then ethni_d2_2=ethni_d2;
                if i=2 then ethni_d3_2=ethni_d3;
                if i=3 then ethni_d1_3=ethni_d1;
                if i=3 then ethni_d2_3=ethni_d2;
                if i=3 then ethni_d3_3=ethni_d3;
                if i=4 then ethni_d1_4=ethni_d1;
                if i=4 then ethni_d2_4=ethni_d2;
                if i=4 then ethni_d3_4=ethni_d3;
                if i=5 then ethni_d1_5=ethni_d1;
                if i=5 then ethni_d2_5=ethni_d2;
                if i=5 then ethni_d3_5=ethni_d3;
                if i=6 then ethni_d1_6=ethni_d1;
                if i=6 then ethni_d2_6=ethni_d2;
                if i=6 then ethni_d3_6=ethni_d3;
                hhtyp_d1_1=0; hhtyp_d1_2=0; hhtyp_d1_3=0; hhtyp_d1_4=0;
hhtyp_d1_5=0; hhtyp_d1_6=0;
                hhtyp_d2_1=0; hhtyp_d2_2=0; hhtyp_d2_3=0; hhtyp_d2_4=0;
hhtyp_d2_5=0; hhtyp_d2_6=0;
                hhtyp_d345_1=0; hhtyp_d345_2=0; hhtyp_d345_3=0; hhtyp_d345_4=0;
hhtyp_d345_5=0; hhtyp_d345_6=0;
                if i=1 then hhtyp_d1_1=hhtyp_d1;
                if i=1 then hhtyp_d2_1=hhtyp_d2;
                if i=1 then hhtyp_d345_1=hhtyp_d345;
                if i=2 then hhtyp_d1_2=hhtyp_d1;
                if i=2 then hhtyp_d2_2=hhtyp_d2;
                if i=2 then hhtyp_d345_2=hhtyp_d345;
                if i=3 then hhtyp_d1_3=hhtyp_d1;
                if i=3 then hhtyp_d2_3=hhtyp_d2;
                if i=3 then hhtyp_d345_3=hhtyp_d345;
                if i=4 then hhtyp_d1_4=hhtyp_d1;
                if i=4 then hhtyp_d2_4=hhtyp_d2;
                if i=4 then hhtyp_d345_4=hhtyp_d345;
                if i=5 then hhtyp_d1_5=hhtyp_d1;
                if i=5 then hhtyp_d2_5=hhtyp_d2;
                if i=5 then hhtyp_d345_5=hhtyp_d345;
                if i=6 then hhtyp_d1_6=hhtyp_d1;
                if i=6 then hhtyp_d2_6=hhtyp_d2;
                if i=6 then hhtyp_d345_6=hhtyp_d345;
                occtyp_d1246_1=0; occtyp_d1246_2=0; occtyp_d1246_3=0;
occtyp_d1246_4=0; occtyp_d1246_5=0; occtyp_d1246_6=0;
                occtyp_d35_1=0; occtyp_d35_2=0; occtyp_d35_3=0; occtyp_d35_4=0;
occtyp_d35_5=0; occtyp_d35_6=0;
```

```
occtyp_d8_1=0; occtyp_d8_2=0; occtyp_d8_3=0; occtyp_d8_4=0;
occtyp_d8_5=0; occtyp_d8_6=0;
occtyp_d710_1=0; occtyp_d710_2=0; occtyp_d710_3=0;
occtyp_d710_4=0; occtyp_d710_5=0; occtyp_d710_6=0;
if i=1 then occtyp_d1246_1=occtyp_d1246;
if i=1 then occtyp_d35_1=occtyp_d35;
if i=1 then occtyp_d8_1=occtyp_d8;
if i=1 then occtyp_d710_1=occtyp_d710;
if i=2 then occtyp_d1246_2=occtyp_d1246;
if i=2 then occtyp_d35_2=occtyp_d35;
if i=2 then occtyp_d8_2=occtyp_d8;
if i=2 then occtyp_d710_2=occtyp_d710;
if i=3 then occtyp_d1246_3=occtyp_d1246;
if i=3 then occtyp_d35_3=occtyp_d35;
if i=3 then occtyp_d8_3=occtyp_d8;
if i=3 then occtyp_d710_3=occtyp_d710;
if i=4 then occtyp_d1246_4=occtyp_d1246;
if i=4 then occtyp_d35_4=occtyp_d35;
if i=4 then occtyp_d8_4=occtyp_d8;
if i=4 then occtyp_d710_4=occtyp_d710;
if i=5 then occtyp_d1246_5=occtyp_d1246;
if i=5 then occtyp_d35_5=occtyp_d35;
if i=5 then occtyp_d8_5=occtyp_d8;
if i=5 then occtyp_d710_5=occtyp_d710;
if i=6 then occtyp_d1246_6=occtyp_d1246;
if i=6 then occtyp_d35_6=occtyp_d35;
if i=6 then occtyp_d8_6=occtyp_d8;
if i=6 then occtyp_d710_6=occtyp_d710;
educ_d12_1=0; educ_d12_2=0; educ_d12_3=0; educ_d12_4=0;
educ_d12_5=0; educ_d12_6=0;
if i=1 then educ_d12_1=educ_d12;
if i=2 then educ_d12_2=educ_d12;
if i=3 then educ_d12_3=educ_d12;
if i=4 then educ_d12_4=educ_d12;
if i=5 then educ_d12_5=educ_d12;
if i=6 then educ_d12_6=educ_d12;
income_d1234_1=0; income_d1234_2=0; income_d1234_3=0;
income_d1234_4=0; income_d1234_5=0; income_d1234_6=0;
income_d56_1=0; income_d56_2=0; income_d56_3=0;
income_d56_4=0; income_d56_5=0; income_d56_6=0;
if i=1 then income_d1234_1=income_d1234;
if i=1 then income_d56_1=income_d56;
if i=2 then income_d1234_2=income_d1234;
if i=2 then income_d56_2=income_d56;
```

```
                if i=3 then income_d1234_3=income_d1234;
                if i=3 then income_d56_3=income_d56;
                if i=4 then income_d1234_4=income_d1234;
                if i=4 then income_d56_4=income_d56;
                if i=5 then income_d1234_5=income_d1234;
                if i=5 then income_d56_5=income_d56;
                if i=6 then income_d1234_6=income_d1234;
                if i=6 then income_d56_6=income_d56;
                purpose_d1_1=0; purpose_d1_2=0; purpose_d1_3=0; purpose_d1_4=0;
purpose_d1_5=0; purpose_d1_6=0;
                purpose_d2_1=0; purpose_d2_2=0; purpose_d2_3=0; purpose_d2_4=0;
purpose_d2_5=0; purpose_d2_6=0;
                purpose_d3_1=0; purpose_d3_2=0; purpose_d3_3=0; purpose_d3_4=0;
purpose_d3_5=0; purpose_d3_6=0;
                if i=1 then purpose_d1_1=purpose_d1;
                if i=1 then purpose_d2_1=purpose_d2;
                if i=1 then purpose_d3_1=purpose_d3;
                if i=2 then purpose_d1_2=purpose_d1;
                if i=2 then purpose_d2_2=purpose_d2;
                if i=2 then purpose_d3_2=purpose_d3;
                if i=3 then purpose_d1_3=purpose_d1;
                if i=3 then purpose_d2_3=purpose_d2;
                if i=3 then purpose_d3_3=purpose_d3;
                if i=4 then purpose_d1_4=purpose_d1;
                if i=4 then purpose_d2_4=purpose_d2;
                if i=4 then purpose_d3_4=purpose_d3;
                if i=5 then purpose_d1_5=purpose_d1;
                if i=5 then purpose_d2_5=purpose_d2;
                if i=5 then purpose_d3_5=purpose_d3;
                if i=6 then purpose_d1_6=purpose_d1;
                if i=6 then purpose_d2_6=purpose_d2;
                if i=6 then purpose_d3_6=purpose_d3;
*****************************;
                pcomtoll_1=0; pcomtoll_3=0; pcomtoll_5=0;
                if i=1 then pcomtoll_1=pcomtoll;
                if i=3 then pcomtoll_3=pcomtoll;
                if i=5 then pcomtoll_5=pcomtoll;
                pwrktoll_1=0; pwrktoll_3=0; pwrktoll_5=0;
                if i=1 then pwrktoll_1=pwrktoll;
                if i=3 then pwrktoll_3=pwrktoll;
                if i=5 then pwrktoll_5=pwrktoll;
                pschtoll_1=0; pschtoll_3=0; pschtoll_5=0;
                if i=1 then pschtoll_1=pschtoll;
                if i=3 then pschtoll_3=pschtoll;
```

```
                if i=5 then pschtoll_5=pschtoll;
                prectoll_1=0; prectoll_3=0; prectoll_5=0;
                if i=1 then prectoll_1=prectoll;
                if i=3 then prectoll_3=prectoll;
                if i=5 then prectoll_5=prectoll;
                tollptoll_1=0; tollptoll_3=0; tollptoll_5=0;
                if i=1 then tollptoll_1=tollptoll;
                if i=3 then tollptoll_3=tollptoll;
                if i=5 then tollptoll_5=tollptoll;
**************************************************************************
*;
                output;
        end;
*proc print data=newdata246;
run;
************************************************************;
* MULTINOMIAL LOGIT ML alts---GENERIC---PARAMETERS (ALL VARS)
proc mdc data=newdata246 maxit=300;
        model decision = unit ttime /*toll */
gender hhnum motorveh mlbin fampool tollpay
age_d1 age_d234
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d345
occtyp_d1246 occtyp_d35 occtyp_d8 occtyp_d710
educ_d12
income_d1234 income_d56
purpose_d1 purpose_d2 purpose_d3
/*pcomtoll pwrktoll pschtoll prectoll tollptoll*/
/ type=clogit choice=(mode_route 2 4 6) optmethod=qn covest=hess;
        id pid;
        output out=probdata pred=p;
*proc print data=probdata;
*run;
************************************************************;

data newdata1(keep= pid id decision decision1 dec_mode dec_route
mode_route mode route
ttime
ttime_1 ttime_2
toll tollptoll
age gender ethnicity hhtype hhnum motorveh occtype educ income mlbin fampool
tollpay
unit
unit_1 unit_2
```

```
age_d1 age_d2 age_d3 age_d4 age_d5
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d3 hhtyp_d4 hhtyp_d5
occtyp_d1 occtyp_d2 occtyp_d3 occtyp_d4 occtyp_d5 occtyp_d6 occtyp_d7 occtyp_d8
occtyp_d9 occtyp_d10 occtyp_d11
educ_d1 educ_d2 educ_d3 educ_d4
income_d123 income_d4 income_d5 income_d6 income_d7 income_d8
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll
tollptoll

gender_1 gender_2
hhnum_1 hhnum_2
motorveh_1 motorveh_2
mlbin_1 mlbin_2
fampool_1 fampool_2
tollpay_1 tollpay_2
age_d1_1 age_d2_1 age_d3_1 age_d4_1 age_d5_1
age_d1_2 age_d2_2 age_d3_2 age_d4_2 age_d5_2
ethni_d1_1 ethni_d2_1 ethni_d3_1
ethni_d1_2 ethni_d2_2 ethni_d3_2
hhtyp_d1_1 hhtyp_d2_1 hhtyp_d3_1 hhtyp_d4_1 hhtyp_d5_1
hhtyp_d1_2 hhtyp_d2_2 hhtyp_d3_2 hhtyp_d4_2 hhtyp_d5_2
occtyp_d1_1 occtyp_d2_1 occtyp_d3_1 occtyp_d4_1 occtyp_d5_1 occtyp_d6_1
occtyp_d7_1 occtyp_d8_1 occtyp_d9_1 occtyp_d10_1 occtyp_d11_1
occtyp_d1_2 occtyp_d2_2 occtyp_d3_2 occtyp_d4_2 occtyp_d5_2 occtyp_d6_2
occtyp_d7_2 occtyp_d8_2 occtyp_d9_2 occtyp_d10_2 occtyp_d11_2
educ_d1_1 educ_d2_1 educ_d3_1 educ_d4_1
educ_d1_2 educ_d2_2 educ_d3_2 educ_d4_2
income_d123_1 income_d4_1 income_d5_1 income_d6_1 income_d7_1 income_d8_1
income_d123_2 income_d4_2 income_d5_2 income_d6_2 income_d7_2 income_d8_2
purpose_d1_1 purpose_d2_1 purpose_d3_1
purpose_d1_2 purpose_d2_2 purpose_d3_2
);

        set dfw12;
        array tvec{2} tt1 - tt2;
        array tollvec{2} toll1 - toll2;

        retain pid 0;
        pid + 1;
        do i = 1 to 2;
                id = surveyid;
                if estim=1 then decision = ( SPCh = i );
```

```
                if estim=0 then decision = .;
                if estim=0 then decision1 = ( SPCh = i );
                mode_route = i;
                if (i=1 or i=2) then mode=1;
                if (i=3 or i=4) then mode=2;
                if (i=5 or i=6) then mode=3;
                if (i=1 or i=3 or i=5) then route=1;
                if (i=2 or i=4 or i=6) then route=2;
                ttime = tvec{i};
                toll = tollvec{i};
                if (i=2 or i=4 or i=6) then toll=0;
*****************************;
*INTERACTIONS;
*****************************;
                pcomtoll=0;    pwrktoll=0;    pschtoll=0;    prectoll=0;
                if purpagg=1 then pcomtoll = toll;
                if purpagg=3 then pwrktoll = toll;
                if purpagg=4 then pschtoll = toll;
                if purpagg=2 then prectoll = toll;

                tollptoll=0;
                if tollpay=1 then tollptoll = toll;
*****************************;
*DUMMIES;
*****************************;
                age_d1=0; age_d2=0; age_d3=0; age_d4=0; age_d5=0;
                if age=1 then age_d1=1;
                if age=2 then age_d2=1;
                if age=3 then age_d3=1;
                if age=4 then age_d4=1;
                if age=5 then age_d5=1;
                ethni_d1=0;    ethni_d2=0;    ethni_d3=0;
                if ethnicity=1 then ethni_d1=1;
                if ethnicity=2 then ethni_d2=1;
                if ethnicity=3 then ethni_d3=1;
                hhtyp_d1=0;    hhtyp_d2=0;    hhtyp_d3=0;    hhtyp_d4=0;    hhtyp_d5=0;
                if hhtype=1 then hhtyp_d1=1;
                if hhtype=2 then hhtyp_d2=1;
                if hhtype=3 then hhtyp_d3=1;
                if hhtype=4 then hhtyp_d4=1;
                if hhtype=5 then hhtyp_d5=1;
                occtyp_d1=0; occtyp_d2=0; occtyp_d3=0; occtyp_d4=0; occtyp_d5=0;
occtyp_d6=0; occtyp_d7=0; occtyp_d8=0;    occtyp_d9=0;
                occtyp_d10=0; occtyp_d11=0;
```

```
                if occtype=1 then occtyp_d1=1;
                if occtype=2 then occtyp_d2=1;
                if occtype=3 then occtyp_d3=1;
                if occtype=4 then occtyp_d4=1;
                if occtype=5 then occtyp_d5=1;
                if occtype=6 then occtyp_d6=1;
                if occtype=7 then occtyp_d7=1;
                if occtype=8 then occtyp_d8=1;
                if occtype=9 then occtyp_d9=1;
                if occtype=10 then occtyp_d10=1;
                if occtype=11 then occtyp_d11=1;
                educ_d1=0; educ_d2=0; educ_d3=0; educ_d4=0;
                if educ=1 then educ_d1=1;
                if educ=2 then educ_d2=1;
                if educ=3 then educ_d3=1;
                if educ=4 then educ_d4=1;
                income_d123=0; income_d4=0; income_d5=0; income_d6=0;
income_d7=0; income_d8=0;
                if (income=1 or income=2 or income=3) then income_d123=1;
                if income=4 then income_d4=1;
                if income=5 then income_d5=1;
                if income=6 then income_d6=1;
                if income=7 then income_d7=1;
                if income=8 then income_d8=1;
                purpose_d1=0; purpose_d2=0;          purpose_d3=0;
                if purpagg=1 then purpose_d1 = 1;
                if purpagg=3 then purpose_d2 = 1;
                if purpagg=4 then purpose_d3 = 1;


*****************************;
*MODE SPECIFICS;
*****************************;
                ttime_1=0;              toll_1=0;
                ttime_2=0;
                if (i EQ 1) then ttime_1=ttime;                  if (i EQ 1) then toll_1=toll;
                if (i EQ 2) then ttime_2=ttime;
                unit = 1;
                unit_1 = 0;
                unit_2 = 0;

                if (i EQ 1) then unit_1=1;
                if (i EQ 2) then unit_2=1;
*******************************;
```

```
*******************************;
*******************************;
                gender_1=0; gender_2=0;
                if i=1 then gender_1=gender;
                if i=2 then gender_2=gender;
                hhnum_1=0; hhnum_2=0;
                if i=1 then hhnum_1=hhnum;
                if i=2 then hhnum_2=hhnum;
                motorveh_1=0; motorveh_2=0;
                if i=1 then motorveh_1=motorveh;
                if i=2 then motorveh_2=motorveh;
                mlbin_1=0; mlbin_2=0;
                if i=1 then mlbin_1=mlbin;
                if i=2 then mlbin_2=mlbin;
                fampool_1=0; fampool_2=0;
                if i=1 then fampool_1=fampool;
                if i=2 then fampool_2=fampool;
                tollpay_1=0; tollpay_2=0;
                if i=1 then tollpay_1=tollpay;
                if i=2 then tollpay_2=tollpay;
                age_d1_1=0; age_d1_2=0;
                age_d2_1=0; age_d2_2=0;
                age_d3_1=0; age_d3_2=0;
                age_d4_1=0; age_d4_2=0;
                age_d5_1=0; age_d5_2=0;
                if i=1 then age_d1_1=age_d1;
                if i=1 then age_d2_1=age_d2;
                if i=1 then age_d3_1=age_d3;
                if i=1 then age_d4_1=age_d4;
                if i=1 then age_d5_1=age_d5;
                if i=2 then age_d1_2=age_d1;
                if i=2 then age_d2_2=age_d2;
                if i=2 then age_d3_2=age_d3;
                if i=2 then age_d4_2=age_d4;
                if i=2 then age_d5_2=age_d5;
                ethni_d1_1=0; ethni_d1_2=0;
                ethni_d2_1=0; ethni_d2_2=0;
                ethni_d3_1=0; ethni_d3_2=0;
                if i=1 then ethni_d1_1=ethni_d1;
                if i=1 then ethni_d2_1=ethni_d2;
                if i=1 then ethni_d3_1=ethni_d3;
                if i=2 then ethni_d1_2=ethni_d1;
                if i=2 then ethni_d2_2=ethni_d2;
                if i=2 then ethni_d3_2=ethni_d3;
```

```
hhtyp_d1_1=0; hhtyp_d1_2=0;
hhtyp_d2_1=0; hhtyp_d2_2=0;
hhtyp_d3_1=0; hhtyp_d3_2=0;
hhtyp_d4_1=0; hhtyp_d4_2=0;
hhtyp_d5_1=0; hhtyp_d5_2=0;
if i=1 then hhtyp_d1_1=hhtyp_d1;
if i=1 then hhtyp_d2_1=hhtyp_d2;
if i=1 then hhtyp_d3_1=hhtyp_d3;
if i=1 then hhtyp_d4_1=hhtyp_d4;
if i=1 then hhtyp_d5_1=hhtyp_d5;
if i=2 then hhtyp_d1_2=hhtyp_d1;
if i=2 then hhtyp_d2_2=hhtyp_d2;
if i=2 then hhtyp_d3_2=hhtyp_d3;
if i=2 then hhtyp_d4_2=hhtyp_d4;
if i=2 then hhtyp_d5_2=hhtyp_d5;
occtyp_d1_1=0; occtyp_d1_2=0;
occtyp_d2_1=0; occtyp_d2_2=0;
occtyp_d3_1=0; occtyp_d3_2=0;
occtyp_d4_1=0; occtyp_d4_2=0;
occtyp_d5_1=0; occtyp_d5_2=0;
occtyp_d6_1=0; occtyp_d6_2=0;
occtyp_d7_1=0; occtyp_d7_2=0;
occtyp_d8_1=0; occtyp_d8_2=0;
occtyp_d9_1=0; occtyp_d9_2=0;
occtyp_d10_1=0; occtyp_d10_2=0;
occtyp_d11_1=0; occtyp_d11_2=0;
if i=1 then occtyp_d1_1=occtyp_d1;
if i=1 then occtyp_d2_1=occtyp_d2;
if i=1 then occtyp_d3_1=occtyp_d3;
if i=1 then occtyp_d4_1=occtyp_d4;
if i=1 then occtyp_d5_1=occtyp_d5;
if i=1 then occtyp_d6_1=occtyp_d6;
if i=1 then occtyp_d7_1=occtyp_d7;
if i=1 then occtyp_d8_1=occtyp_d8;
if i=1 then occtyp_d9_1=occtyp_d9;
if i=1 then occtyp_d10_1=occtyp_d10;
if i=1 then occtyp_d11_1=occtyp_d11;
if i=2 then occtyp_d1_2=occtyp_d1;
if i=2 then occtyp_d2_2=occtyp_d2;
if i=2 then occtyp_d3_2=occtyp_d3;
if i=2 then occtyp_d4_2=occtyp_d4;
if i=2 then occtyp_d5_2=occtyp_d5;
if i=2 then occtyp_d6_2=occtyp_d6;
if i=2 then occtyp_d7_2=occtyp_d7;
```

```
                if i=2 then occtyp_d8_2=occtyp_d8;
                if i=2 then occtyp_d9_2=occtyp_d9;
                if i=2 then occtyp_d10_2=occtyp_d10;
                if i=2 then occtyp_d11_2=occtyp_d11;
                educ_d1_1=0; educ_d1_2=0;
                educ_d2_1=0; educ_d2_2=0;
                educ_d3_1=0; educ_d3_2=0;
                educ_d4_1=0; educ_d4_2=0;
                if i=1 then educ_d1_1=educ_d1;
                if i=1 then educ_d2_1=educ_d2;
                if i=1 then educ_d3_1=educ_d3;
                if i=1 then educ_d4_1=educ_d4;
                if i=2 then educ_d1_2=educ_d1;
                if i=2 then educ_d2_2=educ_d2;
                if i=2 then educ_d3_2=educ_d3;
                if i=2 then educ_d4_2=educ_d4;
                income_d123_1=0; income_d123_2=0;
                income_d4_1=0; income_d4_2=0;
                income_d5_1=0; income_d5_2=0;
                income_d6_1=0; income_d6_2=0;
                income_d7_1=0; income_d7_2=0;
                income_d8_1=0; income_d8_2=0;
                if i=1 then income_d123_1=income_d123;
                if i=1 then income_d4_1=income_d4;
                if i=1 then income_d5_1=income_d5;
                if i=1 then income_d6_1=income_d6;
                if i=1 then income_d7_1=income_d7;
                if i=1 then income_d8_1=income_d8;
                if i=2 then income_d123_2=income_d123;
                if i=2 then income_d4_2=income_d4;
                if i=2 then income_d5_2=income_d5;
                if i=2 then income_d6_2=income_d6;
                if i=2 then income_d7_2=income_d7;
                if i=2 then income_d8_2=income_d8;
                purpose_d1_1=0; purpose_d1_2=0;
                purpose_d2_1=0; purpose_d2_2=0;
                purpose_d3_1=0; purpose_d3_2=0;
                if i=1 then purpose_d1_1=purpose_d1;
                if i=1 then purpose_d2_1=purpose_d2;
                if i=1 then purpose_d3_1=purpose_d3;
                if i=2 then purpose_d1_2=purpose_d1;
                if i=2 then purpose_d2_2=purpose_d2;
                if i=2 then purpose_d3_2=purpose_d3;
**************************;
```

```
                pcomtoll_1=0;
                if i=1 then pcomtoll_1=pcomtoll;
                pwrktoll_1=0;
                if i=1 then pwrktoll_1=pwrktoll;
                pschtoll_1=0;
                if i=1 then pschtoll_1=pschtoll;
                prectoll_1=0;
                if i=1 then prectoll_1=prectoll;
                tollptoll_1=0;
                if i=1 then tollptoll_1=tollptoll;
**************************************************************************
*;
                output;
        end;
*proc print data=newdata1;
run;
*/************************************************/
* BINOMIAL LOGIT WITH---MODE1---GENERIC---PARAMETERS (ALL VARS)
proc mdc data=newdata1 maxit=300;
        model decision = unit ttime toll
gender hhnum motorveh mlbin fampool tollpay
age_d1 age_d2 age_d3 age_d4 age_d5
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d3 hhtyp_d4 hhtyp_d5
occtyp_d1 occtyp_d2 occtyp_d3 occtyp_d4 occtyp_d5 occtyp_d6 occtyp_d7 occtyp_d8
occtyp_d9 occtyp_d10 occtyp_d11
educ_d1 educ_d2 educ_d3 educ_d4
income_d123 income_d4 income_d5 income_d6 income_d7 income_d8
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll
/ type=clogit choice=(mode_route 1 2) optmethod=qn covest=hess;
        id pid;
        output out=probdata pred=p;
*proc print data=probdata;
*run;
*/************************************************/
* BINOMIAL LOGIT WITH---MODE1---GENERIC---PARAMETERS (unit ttime
toll)
proc mdc data=newdata1 maxit=300;
        model decision = unit ttime toll
/ type=clogit choice=(mode_route 1 2) optmethod=qn covest=hess;
        id pid;
        output out=probdata pred=p;
*proc print data=probdata;
```

```
*run;
data newdata2(keep= pid id decision decision1 dec_mode dec_route
mode_route mode route
ttime
ttime_3 ttime_4
toll tollptoll
age gender ethnicity hhtype hhnum motorveh occtype educ income mlbin fampool
tollpay
unit
unit_3 unit_4
age_d1 age_d2 age_d3 age_d4 age_d5
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d3 hhtyp_d4 hhtyp_d5
occtyp_d1 occtyp_d2 occtyp_d3 occtyp_d4 occtyp_d5 occtyp_d6 occtyp_d7 occtyp_d8
occtyp_d9 occtyp_d10 occtyp_d11
educ_d1 educ_d2 educ_d3 educ_d4
income_d123 income_d4 income_d5 income_d6 income_d7 income_d8
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll
tollptoll

gender_3 gender_4
hhnum_3 hhnum_4
motorveh_3 motorveh_4
mlbin_3 mlbin_4
fampool_3 fampool_4
tollpay_3 tollpay_4
age_d1_3 age_d2_3 age_d3_3 age_d4_3 age_d5_3
age_d1_4 age_d2_4 age_d3_4 age_d4_4 age_d5_4
ethni_d1_3 ethni_d2_3 ethni_d3_3
ethni_d1_4 ethni_d2_4 ethni_d3_4
hhtyp_d1_3 hhtyp_d2_3 hhtyp_d3_3 hhtyp_d4_3 hhtyp_d5_3
hhtyp_d1_4 hhtyp_d2_4 hhtyp_d3_4 hhtyp_d4_4 hhtyp_d5_4
occtyp_d1_3 occtyp_d2_3 occtyp_d3_3 occtyp_d4_3 occtyp_d5_3 occtyp_d6_3
occtyp_d7_3 occtyp_d8_3 occtyp_d9_3 occtyp_d10_3 occtyp_d11_3
occtyp_d1_4 occtyp_d2_4 occtyp_d3_4 occtyp_d4_4 occtyp_d5_4 occtyp_d6_4
occtyp_d7_4 occtyp_d8_4 occtyp_d9_4 occtyp_d10_4 occtyp_d11_4
educ_d1_3 educ_d2_3 educ_d3_3 educ_d4_3
educ_d1_4 educ_d2_4 educ_d3_4 educ_d4_4
income_d123_3 income_d4_3 income_d5_3 income_d6_3 income_d7_3 income_d8_3
income_d123_4 income_d4_4 income_d5_4 income_d6_4 income_d7_4 income_d8_4
purpose_d1_3 purpose_d2_3 purpose_d3_3
purpose_d1_4 purpose_d2_4 purpose_d3_4
);
```

```
        set dfw34;
        array tvec{2} tt3 - tt4;
        array tollvec{2} toll3 - toll4;

        retain pid 0;
        pid + 1;
        do i = 3 to 4;
                id = surveyid;
                if estim=1 then decision = ( SPCh = i );
                if estim=0 then decision = .;
                if estim=0 then decision1 = ( SPCh = i );
                mode_route = i;
                if (i=1 or i=2) then mode=1;
                if (i=3 or i=4) then mode=2;
                if (i=5 or i=6) then mode=3;
                if (i=1 or i=3 or i=5) then route=1;
                if (i=2 or i=4 or i=6) then route=2;
*               ttime = tvec{i};
*               toll = tollvec{i};
                if i=3 then ttime=tt3;
                if i=4 then ttime=tt4;
                if i=3 then toll=toll3;
                if i=4 then toll=toll4;
                if (i=2 or i=4 or i=6) then toll=0;
*****************************;
*INTERACTIONS;
*****************************;
                pcomtoll=0;   pwrktoll=0;   pschtoll=0;   prectoll=0;
                if purpagg=1 then pcomtoll = toll;
                if purpagg=3 then pwrktoll = toll;
                if purpagg=4 then pschtoll = toll;
                if purpagg=2 then prectoll = toll;

                tollptoll=0;
                if tollpay=1 then tollptoll = toll;
*****************************;
*DUMMIES;
*****************************;
                age_d1=0; age_d2=0; age_d3=0; age_d4=0; age_d5=0;
                if age=1 then age_d1=1;
                if age=2 then age_d2=1;
                if age=3 then age_d3=1;
                if age=4 then age_d4=1;
```

```
                if age=5 then age_d5=1;
                ethni_d1=0;    ethni_d2=0;    ethni_d3=0;
                if ethnicity=1 then ethni_d1=1;
                if ethnicity=2 then ethni_d2=1;
                if ethnicity=3 then ethni_d3=1;
                hhtyp_d1=0;   hhtyp_d2=0;   hhtyp_d3=0;   hhtyp_d4=0;   hhtyp_d5=0;
                if hhtype=1 then hhtyp_d1=1;
                if hhtype=2 then hhtyp_d2=1;
                if hhtype=3 then hhtyp_d3=1;
                if hhtype=4 then hhtyp_d4=1;
                if hhtype=5 then hhtyp_d5=1;
                occtyp_d1=0; occtyp_d2=0; occtyp_d3=0; occtyp_d4=0; occtyp_d5=0;
occtyp_d6=0; occtyp_d7=0; occtyp_d8=0;    occtyp_d9=0;
                occtyp_d10=0; occtyp_d11=0;
                if occtype=1 then occtyp_d1=1;
                if occtype=2 then occtyp_d2=1;
                if occtype=3 then occtyp_d3=1;
                if occtype=4 then occtyp_d4=1;
                if occtype=5 then occtyp_d5=1;
                if occtype=6 then occtyp_d6=1;
                if occtype=7 then occtyp_d7=1;
                if occtype=8 then occtyp_d8=1;
                if occtype=9 then occtyp_d9=1;
                if occtype=10 then occtyp_d10=1;
                if occtype=11 then occtyp_d11=1;
                educ_d1=0; educ_d2=0; educ_d3=0; educ_d4=0;
                if educ=1 then educ_d1=1;
                if educ=2 then educ_d2=1;
                if educ=3 then educ_d3=1;
                if educ=4 then educ_d4=1;
                income_d123=0; income_d4=0; income_d5=0; income_d6=0;
income_d7=0; income_d8=0;
                if (income=1 or income=2 or income=3) then income_d123=1;
                if income=4 then income_d4=1;
                if income=5 then income_d5=1;
                if income=6 then income_d6=1;
                if income=7 then income_d7=1;
                if income=8 then income_d8=1;
                purpose_d1=0; purpose_d2=0;          purpose_d3=0;
                if purpagg=1 then purpose_d1 = 1;
                if purpagg=3 then purpose_d2 = 1;
                if purpagg=4 then purpose_d3 = 1;
```

```
*****************************;
*MODE SPECIFICS;
*****************************;
                ttime_3=0;              toll_3=0;
                ttime_4=0;
                if (i EQ 3) then ttime_3=ttime;                 if (i EQ 3) then toll_3=toll;
                if (i EQ 4) then ttime_4=ttime;
                unit = 1;
                unit_3 = 0;
                unit_4 = 0;

                if (i EQ 3) then unit_3=1;
                if (i EQ 4) then unit_4=1;
*******************************;
*******************************;
*****************************;
                gender_3=0; gender_4=0;
                if i=3 then gender_3=gender;
                if i=4 then gender_4=gender;
                hhnum_3=0; hhnum_4=0;
                if i=3 then hhnum_3=hhnum;
                if i=4 then hhnum_4=hhnum;
                motorveh_3=0; motorveh_4=0;
                if i=3 then motorveh_3=motorveh;
                if i=4 then motorveh_4=motorveh;
                mlbin_3=0; mlbin_4=0;
                if i=3 then mlbin_3=mlbin;
                if i=4 then mlbin_4=mlbin;
                fampool_3=0; fampool_4=0;
                if i=3 then fampool_3=fampool;
                if i=4 then fampool_4=fampool;
                tollpay_3=0; tollpay_4=0;
                if i=3 then tollpay_3=tollpay;
                if i=4 then tollpay_4=tollpay;
                age_d1_3=0; age_d1_4=0;
                age_d2_3=0; age_d2_4=0;
                age_d3_3=0; age_d3_4=0;
                age_d4_3=0; age_d4_4=0;
                age_d5_3=0; age_d5_4=0;
                if i=3 then age_d1_3=age_d1;
                if i=3 then age_d2_3=age_d2;
                if i=3 then age_d3_3=age_d3;
                if i=3 then age_d4_3=age_d4;
                if i=3 then age_d5_3=age_d5;
```

```
if i=4 then age_d1_4=age_d1;
if i=4 then age_d2_4=age_d2;
if i=4 then age_d3_4=age_d3;
if i=4 then age_d4_4=age_d4;
if i=4 then age_d5_4=age_d5;
ethni_d1_3=0; ethni_d1_4=0;
ethni_d2_3=0; ethni_d2_4=0;
ethni_d3_3=0; ethni_d3_4=0;
if i=3 then ethni_d1_3=ethni_d1;
if i=3 then ethni_d2_3=ethni_d2;
if i=3 then ethni_d3_3=ethni_d3;
if i=4 then ethni_d1_4=ethni_d1;
if i=4 then ethni_d2_4=ethni_d2;
if i=4 then ethni_d3_4=ethni_d3;
hhtyp_d1_3=0; hhtyp_d1_4=0;
hhtyp_d2_3=0; hhtyp_d2_4=0;
hhtyp_d3_3=0; hhtyp_d3_4=0;
hhtyp_d4_3=0; hhtyp_d4_4=0;
hhtyp_d5_3=0; hhtyp_d5_4=0;
if i=3 then hhtyp_d1_3=hhtyp_d1;
if i=3 then hhtyp_d2_3=hhtyp_d2;
if i=3 then hhtyp_d3_3=hhtyp_d3;
if i=3 then hhtyp_d4_3=hhtyp_d4;
if i=3 then hhtyp_d5_3=hhtyp_d5;
if i=4 then hhtyp_d1_4=hhtyp_d1;
if i=4 then hhtyp_d2_4=hhtyp_d2;
if i=4 then hhtyp_d3_4=hhtyp_d3;
if i=4 then hhtyp_d4_4=hhtyp_d4;
if i=4 then hhtyp_d5_4=hhtyp_d5;
occtyp_d1_3=0; occtyp_d1_4=0;
occtyp_d2_3=0; occtyp_d2_4=0;
occtyp_d3_3=0; occtyp_d3_4=0;
occtyp_d4_3=0; occtyp_d4_4=0;
occtyp_d5_3=0; occtyp_d5_4=0;
occtyp_d6_3=0; occtyp_d6_4=0;
occtyp_d7_3=0; occtyp_d7_4=0;
occtyp_d8_3=0; occtyp_d8_4=0;
occtyp_d9_3=0; occtyp_d9_4=0;
occtyp_d10_3=0; occtyp_d10_4=0;
occtyp_d11_3=0; occtyp_d11_4=0;
if i=3 then occtyp_d1_3=occtyp_d1;
if i=3 then occtyp_d2_3=occtyp_d2;
if i=3 then occtyp_d3_3=occtyp_d3;
if i=3 then occtyp_d4_3=occtyp_d4;
```

```
if i=3 then occtyp_d5_3=occtyp_d5;
if i=3 then occtyp_d6_3=occtyp_d6;
if i=3 then occtyp_d7_3=occtyp_d7;
if i=3 then occtyp_d8_3=occtyp_d8;
if i=3 then occtyp_d9_3=occtyp_d9;
if i=3 then occtyp_d10_3=occtyp_d10;
if i=3 then occtyp_d11_3=occtyp_d11;
if i=4 then occtyp_d1_4=occtyp_d1;
if i=4 then occtyp_d2_4=occtyp_d2;
if i=4 then occtyp_d3_4=occtyp_d3;
if i=4 then occtyp_d4_4=occtyp_d4;
if i=4 then occtyp_d5_4=occtyp_d5;
if i=4 then occtyp_d6_4=occtyp_d6;
if i=4 then occtyp_d7_4=occtyp_d7;
if i=4 then occtyp_d8_4=occtyp_d8;
if i=4 then occtyp_d9_4=occtyp_d9;
if i=4 then occtyp_d10_4=occtyp_d10;
if i=4 then occtyp_d11_4=occtyp_d11;
educ_d1_3=0; educ_d1_4=0;
educ_d2_3=0; educ_d2_4=0;
educ_d3_3=0; educ_d3_4=0;
educ_d4_3=0; educ_d4_4=0;
if i=3 then educ_d1_3=educ_d1;
if i=3 then educ_d2_3=educ_d2;
if i=3 then educ_d3_3=educ_d3;
if i=3 then educ_d4_3=educ_d4;
if i=4 then educ_d1_4=educ_d1;
if i=4 then educ_d2_4=educ_d2;
if i=4 then educ_d3_4=educ_d3;
if i=4 then educ_d4_4=educ_d4;
income_d123_3=0; income_d123_4=0;
income_d4_3=0; income_d4_4=0;
income_d5_3=0; income_d5_4=0;
income_d6_3=0; income_d6_4=0;
income_d7_3=0; income_d7_4=0;
income_d8_3=0; income_d8_4=0;
if i=3 then income_d123_3=income_d123;
if i=3 then income_d4_3=income_d4;
if i=3 then income_d5_3=income_d5;
if i=3 then income_d6_3=income_d6;
if i=3 then income_d7_3=income_d7;
if i=3 then income_d8_3=income_d8;
if i=4 then income_d123_4=income_d123;
if i=4 then income_d4_4=income_d4;
```

```
                if i=4 then income_d5_4=income_d5;
                if i=4 then income_d6_4=income_d6;
                if i=4 then income_d7_4=income_d7;
                if i=4 then income_d8_4=income_d8;
                purpose_d1_3=0; purpose_d1_4=0;
                purpose_d2_3=0; purpose_d2_4=0;
                purpose_d3_3=0; purpose_d3_4=0;
                if i=3 then purpose_d1_3=purpose_d1;
                if i=3 then purpose_d2_3=purpose_d2;
                if i=3 then purpose_d3_3=purpose_d3;
                if i=4 then purpose_d1_4=purpose_d1;
                if i=4 then purpose_d2_4=purpose_d2;
                if i=4 then purpose_d3_4=purpose_d3;
*****************************;
                pcomtoll_3=0;
                if i=3 then pcomtoll_3=pcomtoll;
                pwrktoll_3=0;
                if i=3 then pwrktoll_3=pwrktoll;
                pschtoll_3=0;
                if i=3 then pschtoll_3=pschtoll;
                prectoll_3=0;
                if i=3 then prectoll_3=prectoll;
                tollptoll_3=0;
                if i=3 then tollptoll_3=tollptoll;
*************************************************************************
*;
                output;
        end;
*proc print data=newdata2;
run;
*/************************************************/
* BINOMIAL LOGIT WITH---MODE2---GENERIC---PARAMETERS (ALL VARS)
proc mdc data=newdata2 maxit=300;
        model decision = unit ttime toll
gender hhnum motorveh mlbin fampool tollpay
age_d1 age_d2 age_d3 age_d4 age_d5
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d3 hhtyp_d4 hhtyp_d5
occtyp_d1 occtyp_d2 occtyp_d3 occtyp_d4 occtyp_d5 occtyp_d6 occtyp_d7 occtyp_d8
occtyp_d9 occtyp_d10 occtyp_d11
educ_d1 educ_d2 educ_d3 educ_d4
income_d123 income_d4 income_d5 income_d6 income_d7 income_d8
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll
```

```
/ type=clogit choice=(mode_route 3 4) optmethod=qn covest=hess;
        id pid;
        output out=probdata pred=p;
*proc print data=probdata;
*run;
*/************************************************/
* BINOMIAL LOGIT WITH---MODE2---GENERIC---PARAMETERS (unit ttime
toll)
proc mdc data=newdata2 maxit=300;
        model decision = unit ttime toll
/ type=clogit choice=(mode_route 3 4) optmethod=qn covest=hess;
        id pid;
        output out=probdata pred=p;
*proc print data=probdata;
*run;
data newdata3(keep= pid id decision decision1 dec_mode dec_route
mode_route mode route
ttime
ttime_5 ttime_6
toll tollptoll
age gender ethnicity hhtype hhnum motorveh occtype educ income mlbin fampool
tollpay
unit
unit_5 unit_6
age_d1 age_d2 age_d3 age_d4 age_d5
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d3 hhtyp_d4 hhtyp_d5
occtyp_d1 occtyp_d2 occtyp_d3 occtyp_d4 occtyp_d5 occtyp_d6 occtyp_d7 occtyp_d8
occtyp_d9 occtyp_d10 occtyp_d11
educ_d1 educ_d2 educ_d3 educ_d4
income_d123 income_d4 income_d5 income_d6 income_d7 income_d8
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll
tollptoll

gender_5 gender_6
hhnum_5 hhnum_6
motorveh_5 motorveh_6
mlbin_5 mlbin_6
fampool_5 fampool_6
tollpay_5 tollpay_6
age_d1_5 age_d2_5 age_d3_5 age_d4_5 age_d5_5
age_d1_6 age_d2_6 age_d3_6 age_d4_6 age_d5_6
ethni_d1_5 ethni_d2_5 ethni_d3_5
```

```
ethni_d1_6 ethni_d2_6 ethni_d3_6
hhtyp_d1_5 hhtyp_d2_5 hhtyp_d3_5 hhtyp_d4_5 hhtyp_d5_5
hhtyp_d1_6 hhtyp_d2_6 hhtyp_d3_6 hhtyp_d4_6 hhtyp_d5_6
occtyp_d1_5 occtyp_d2_5 occtyp_d3_5 occtyp_d4_5 occtyp_d5_5 occtyp_d6_5
occtyp_d7_5 occtyp_d8_5 occtyp_d9_5 occtyp_d10_5 occtyp_d11_5
occtyp_d1_6 occtyp_d2_6 occtyp_d3_6 occtyp_d4_6 occtyp_d5_6 occtyp_d6_6
occtyp_d7_6 occtyp_d8_6 occtyp_d9_6 occtyp_d10_6 occtyp_d11_6
educ_d1_5 educ_d2_5 educ_d3_5 educ_d4_5
educ_d1_6 educ_d2_6 educ_d3_6 educ_d4_6
income_d123_5 income_d4_5 income_d5_5 income_d6_5 income_d7_5 income_d8_5
income_d123_6 income_d4_6 income_d5_6 income_d6_6 income_d7_6 income_d8_6
purpose_d1_5 purpose_d2_5 purpose_d3_5
purpose_d1_6 purpose_d2_6 purpose_d3_6
);

        set dfw56;
        array tvec{2} tt5 - tt6;
        array tollvec{2} toll5 - toll6;

        retain pid 0;
        pid + 1;
        do i = 5 to 6;
                id = surveyid;
                if estim=1 then decision = ( SPCh = i );
                if estim=0 then decision = .;
                if estim=0 then decision1 = ( SPCh = i );
                mode_route = i;
                if (i=1 or i=2) then mode=1;
                if (i=3 or i=4) then mode=2;
                if (i=5 or i=6) then mode=3;
                if (i=1 or i=3 or i=5) then route=1;
                if (i=2 or i=4 or i=6) then route=2;
*               ttime = tvec{i};
*               toll = tollvec{i};
                if i=5 then ttime=tt5;
                if i=6 then ttime=tt6;
                if i=5 then toll=toll5;
                if i=6 then toll=toll6;
                if (i=2 or i=4 or i=6) then toll=0;
*****************************;
*INTERACTIONS;
*****************************;
                pcomtoll=0;   pwrktoll=0;   pschtoll=0;   prectoll=0;
                if purpagg=1 then pcomtoll = toll;
```

```
                if purpagg=3 then pwrktoll = toll;
                if purpagg=4 then pschtoll = toll;
                if purpagg=2 then prectoll = toll;

                tollptoll=0;
                if tollpay=1 then tollptoll = toll;
*****************************;
*DUMMIES;
*****************************;
                age_d1=0; age_d2=0; age_d3=0; age_d4=0; age_d5=0;
                if age=1 then age_d1=1;
                if age=2 then age_d2=1;
                if age=3 then age_d3=1;
                if age=4 then age_d4=1;
                if age=5 then age_d5=1;
                ethni_d1=0;    ethni_d2=0;    ethni_d3=0;
                if ethnicity=1 then ethni_d1=1;
                if ethnicity=2 then ethni_d2=1;
                if ethnicity=3 then ethni_d3=1;
                hhtyp_d1=0;   hhtyp_d2=0;   hhtyp_d3=0;   hhtyp_d4=0;   hhtyp_d5=0;
                if hhtype=1 then hhtyp_d1=1;
                if hhtype=2 then hhtyp_d2=1;
                if hhtype=3 then hhtyp_d3=1;
                if hhtype=4 then hhtyp_d4=1;
                if hhtype=5 then hhtyp_d5=1;
                occtyp_d1=0; occtyp_d2=0; occtyp_d3=0; occtyp_d4=0; occtyp_d5=0;
occtyp_d6=0; occtyp_d7=0; occtyp_d8=0;    occtyp_d9=0;
                occtyp_d10=0; occtyp_d11=0;
                if occtype=1 then occtyp_d1=1;
                if occtype=2 then occtyp_d2=1;
                if occtype=3 then occtyp_d3=1;
                if occtype=4 then occtyp_d4=1;
                if occtype=5 then occtyp_d5=1;
                if occtype=6 then occtyp_d6=1;
                if occtype=7 then occtyp_d7=1;
                if occtype=8 then occtyp_d8=1;
                if occtype=9 then occtyp_d9=1;
                if occtype=10 then occtyp_d10=1;
                if occtype=11 then occtyp_d11=1;
                educ_d1=0; educ_d2=0; educ_d3=0; educ_d4=0;
                if educ=1 then educ_d1=1;
                if educ=2 then educ_d2=1;
                if educ=3 then educ_d3=1;
                if educ=4 then educ_d4=1;
```

```
                income_d123=0; income_d4=0; income_d5=0; income_d6=0;
income_d7=0; income_d8=0;
                if (income=1 or income=2 or income=3) then income_d123=1;
                if income=4 then income_d4=1;
                if income=5 then income_d5=1;
                if income=6 then income_d6=1;
                if income=7 then income_d7=1;
                if income=8 then income_d8=1;
                purpose_d1=0; purpose_d2=0;          purpose_d3=0;
                if purpagg=1 then purpose_d1 = 1;
                if purpagg=3 then purpose_d2 = 1;
                if purpagg=4 then purpose_d3 = 1;


*******************************;
*MODE SPECIFICS;
*******************************;
                ttime_5=0;              toll_5=0;
                ttime_6=0;
                if (i EQ 5) then ttime_5=ttime;                  if (i EQ 5) then toll_5=toll;
                if (i EQ 6) then ttime_6=ttime;
                unit = 1;
                unit_5 = 0;
                unit_6 = 0;

                if (i EQ 5) then unit_5=1;
                if (i EQ 6) then unit_6=1;
*******************************;
*******************************;
*******************************;
                gender_5=0; gender_6=0;
                if i=5 then gender_5=gender;
                if i=6 then gender_6=gender;
                hhnum_5=0; hhnum_6=0;
                if i=5 then hhnum_5=hhnum;
                if i=6 then hhnum_6=hhnum;
                motorveh_5=0; motorveh_6=0;
                if i=5 then motorveh_5=motorveh;
                if i=6 then motorveh_6=motorveh;
                mlbin_5=0; mlbin_6=0;
                if i=5 then mlbin_5=mlbin;
                if i=6 then mlbin_6=mlbin;
                fampool_5=0; fampool_6=0;
                if i=5 then fampool_5=fampool;
```

```
if i=6 then fampool_6=fampool;
tollpay_5=0; tollpay_6=0;
if i=5 then tollpay_5=tollpay;
if i=6 then tollpay_6=tollpay;
age_d1_5=0; age_d1_6=0;
age_d2_5=0; age_d2_6=0;
age_d3_5=0; age_d3_6=0;
age_d4_5=0; age_d4_6=0;
age_d5_5=0; age_d5_6=0;
if i=5 then age_d1_5=age_d1;
if i=5 then age_d2_5=age_d2;
if i=5 then age_d3_5=age_d3;
if i=5 then age_d4_5=age_d4;
if i=5 then age_d5_5=age_d5;
if i=6 then age_d1_6=age_d1;
if i=6 then age_d2_6=age_d2;
if i=6 then age_d3_6=age_d3;
if i=6 then age_d4_6=age_d4;
if i=6 then age_d5_6=age_d5;
ethni_d1_5=0; ethni_d1_6=0;
ethni_d2_5=0; ethni_d2_6=0;
ethni_d3_5=0; ethni_d3_6=0;
if i=5 then ethni_d1_5=ethni_d1;
if i=5 then ethni_d2_5=ethni_d2;
if i=5 then ethni_d3_5=ethni_d3;
if i=6 then ethni_d1_6=ethni_d1;
if i=6 then ethni_d2_6=ethni_d2;
if i=6 then ethni_d3_6=ethni_d3;
hhtyp_d1_5=0; hhtyp_d1_6=0;
hhtyp_d2_5=0; hhtyp_d2_6=0;
hhtyp_d3_5=0; hhtyp_d3_6=0;
hhtyp_d4_5=0; hhtyp_d4_6=0;
hhtyp_d5_5=0; hhtyp_d5_6=0;
if i=5 then hhtyp_d1_5=hhtyp_d1;
if i=5 then hhtyp_d2_5=hhtyp_d2;
if i=5 then hhtyp_d3_5=hhtyp_d3;
if i=5 then hhtyp_d4_5=hhtyp_d4;
if i=5 then hhtyp_d5_5=hhtyp_d5;
if i=6 then hhtyp_d1_6=hhtyp_d1;
if i=6 then hhtyp_d2_6=hhtyp_d2;
if i=6 then hhtyp_d3_6=hhtyp_d3;
if i=6 then hhtyp_d4_6=hhtyp_d4;
if i=6 then hhtyp_d5_6=hhtyp_d5;
occtyp_d1_5=0; occtyp_d1_6=0;
```

```
occtyp_d2_5=0; occtyp_d2_6=0;
occtyp_d3_5=0; occtyp_d3_6=0;
occtyp_d4_5=0; occtyp_d4_6=0;
occtyp_d5_5=0; occtyp_d5_6=0;
occtyp_d6_5=0; occtyp_d6_6=0;
occtyp_d7_5=0; occtyp_d7_6=0;
occtyp_d8_5=0; occtyp_d8_6=0;
occtyp_d9_5=0; occtyp_d9_6=0;
occtyp_d10_5=0; occtyp_d10_6=0;
occtyp_d11_5=0; occtyp_d11_6=0;
if i=5 then occtyp_d1_5=occtyp_d1;
if i=5 then occtyp_d2_5=occtyp_d2;
if i=5 then occtyp_d3_5=occtyp_d3;
if i=5 then occtyp_d4_5=occtyp_d4;
if i=5 then occtyp_d5_5=occtyp_d5;
if i=5 then occtyp_d6_5=occtyp_d6;
if i=5 then occtyp_d7_5=occtyp_d7;
if i=5 then occtyp_d8_5=occtyp_d8;
if i=5 then occtyp_d9_5=occtyp_d9;
if i=5 then occtyp_d10_5=occtyp_d10;
if i=5 then occtyp_d11_5=occtyp_d11;
if i=6 then occtyp_d1_6=occtyp_d1;
if i=6 then occtyp_d2_6=occtyp_d2;
if i=6 then occtyp_d3_6=occtyp_d3;
if i=6 then occtyp_d4_6=occtyp_d4;
if i=6 then occtyp_d5_6=occtyp_d5;
if i=6 then occtyp_d6_6=occtyp_d6;
if i=6 then occtyp_d7_6=occtyp_d7;
if i=6 then occtyp_d8_6=occtyp_d8;
if i=6 then occtyp_d9_6=occtyp_d9;
if i=6 then occtyp_d10_6=occtyp_d10;
if i=6 then occtyp_d11_6=occtyp_d11;
educ_d1_5=0; educ_d1_6=0;
educ_d2_5=0; educ_d2_6=0;
educ_d3_5=0; educ_d3_6=0;
educ_d4_5=0; educ_d4_6=0;
if i=5 then educ_d1_5=educ_d1;
if i=5 then educ_d2_5=educ_d2;
if i=5 then educ_d3_5=educ_d3;
if i=5 then educ_d4_5=educ_d4;
if i=6 then educ_d1_6=educ_d1;
if i=6 then educ_d2_6=educ_d2;
if i=6 then educ_d3_6=educ_d3;
if i=6 then educ_d4_6=educ_d4;
```

```
                income_d123_5=0; income_d123_6=0;
                income_d4_5=0; income_d4_6=0;
                income_d5_5=0; income_d5_6=0;
                income_d6_5=0; income_d6_6=0;
                income_d7_5=0; income_d7_6=0;
                income_d8_5=0; income_d8_6=0;
                if i=5 then income_d123_5=income_d123;
                if i=5 then income_d4_5=income_d4;
                if i=5 then income_d5_5=income_d5;
                if i=5 then income_d6_5=income_d6;
                if i=5 then income_d7_5=income_d7;
                if i=5 then income_d8_5=income_d8;
                if i=6 then income_d123_6=income_d123;
                if i=6 then income_d4_6=income_d4;
                if i=6 then income_d5_6=income_d5;
                if i=6 then income_d6_6=income_d6;
                if i=6 then income_d7_6=income_d7;
                if i=6 then income_d8_6=income_d8;
                purpose_d1_5=0; purpose_d1_6=0;
                purpose_d2_5=0; purpose_d2_6=0;
                purpose_d3_5=0; purpose_d3_6=0;
                if i=5 then purpose_d1_5=purpose_d1;
                if i=5 then purpose_d2_5=purpose_d2;
                if i=5 then purpose_d3_5=purpose_d3;
                if i=6 then purpose_d1_6=purpose_d1;
                if i=6 then purpose_d2_6=purpose_d2;
                if i=6 then purpose_d3_6=purpose_d3;
*****************************;
                pcomtoll_5=0;
                if i=5 then pcomtoll_5=pcomtoll;
                pwrktoll_5=0;
                if i=5 then pwrktoll_5=pwrktoll;
                pschtoll_5=0;
                if i=5 then pschtoll_5=pschtoll;
                prectoll_5=0;
                if i=5 then prectoll_5=prectoll;
                tollptoll_5=0;
                if i=5 then tollptoll_5=tollptoll;
*************************************************************************
*;
                output;
        end;
*proc print data=newdata3;
run;
```

```
*/************************************************/
* BINOMIAL LOGIT WITH---MODE3---GENERIC---PARAMETERS (ALL VARS)
proc mdc data=newdata3 maxit=300;
        model decision = unit ttime toll
gender hhnum motorveh mlbin fampool tollpay
age_d1 age_d2 age_d3 age_d4 age_d5
ethni_d1 ethni_d2 ethni_d3
hhtyp_d1 hhtyp_d2 hhtyp_d3 hhtyp_d4 hhtyp_d5
occtyp_d1 occtyp_d2 occtyp_d3 occtyp_d4 occtyp_d5 occtyp_d6 occtyp_d7 occtyp_d8
occtyp_d9 occtyp_d10 occtyp_d11
educ_d1 educ_d2 educ_d3 educ_d4
income_d123 income_d4 income_d5 income_d6 income_d7 income_d8
purpose_d1 purpose_d2 purpose_d3
pcomtoll pwrktoll pschtoll prectoll tollptoll
/ type=clogit choice=(mode_route 5 6) optmethod=qn covest=hess;
        id pid;
        output out=probdata pred=p;
*proc print data=probdata;
*run;
*/************************************************/
* BINOMIAL LOGIT WITH---MODE2---GENERIC---PARAMETERS (unit ttime
toll)
proc mdc data=newdata3 maxit=300;
        model decision = unit ttime toll
/ type=clogit choice=(mode_route 5 6) optmethod=qn covest=hess;
        id pid;
        output out=probdata pred=p;
*proc print data=probdata;
*run;
/*********************************************/;
*VALIDATION QUERIES;
/*********************************************/;
PROC SQL;
CREATE TABLE obs_e AS
        SELECT pid, count(pid) as N, max(decision) as dec_obs, mode_route
                FROM probdata
                WHERE decision >= 0
                GROUP BY pid
                HAVING decision = calculated dec_obs;
QUIT;
PROC SQL;
CREATE TABLE pred_e AS
        SELECT pid, count(pid) as N, max(p) as dec_pred, mode_route
                FROM probdata
```

```
                WHERE (decision >= 0) AND (p ne .)
                GROUP BY pid
                HAVING p = calculated dec_pred;
QUIT;
PROC SQL;
CREATE TABLE obs_pred_e AS
SELECT A.pid, A.mode_route AS mode_obs, B.mode_route AS mode_pred
FROM pred_e B LEFT JOIN obs_e A
ON (A.pid = B.pid);
QUIT;
PROC SQL;
CREATE TABLE p1_e AS
        SELECT mode_obs, count(mode_pred) as pred1
                FROM obs_pred_e
                WHERE mode_pred=1
                GROUP BY mode_obs;
QUIT;
PROC SQL;
CREATE TABLE p2_e AS
        SELECT mode_obs, count(mode_pred) as pred2
                FROM obs_pred_e
                WHERE mode_pred=2
                GROUP BY mode_obs;
QUIT;
PROC SQL;
CREATE TABLE p3_e AS
        SELECT mode_obs, count(mode_pred) as pred3
                FROM obs_pred_e
                WHERE mode_pred=3
                GROUP BY mode_obs;
QUIT;
PROC SQL;
CREATE TABLE p4_e AS
        SELECT mode_obs, count(mode_pred) as pred4
                FROM obs_pred_e
                WHERE mode_pred=4
                GROUP BY mode_obs;
QUIT;
PROC SQL;
CREATE TABLE p5_e AS
        SELECT mode_obs, count(mode_pred) as pred5
                FROM obs_pred_e
                WHERE mode_pred=5
                GROUP BY mode_obs;
```

```
QUIT;
PROC SQL;
CREATE TABLE p6_e AS
        SELECT mode_obs, count(mode_pred) as pred6
                FROM obs_pred_e
                WHERE mode_pred=6
                GROUP BY mode_obs;
QUIT;
data final_e;
merge p1_e p2_e p3_e p4_e p5_e p6_e;
by mode_obs;
run;
PROC EXPORT DATA= WORK.Final_e
        OUTFILE= "C:\Documents and Settings\kxf7416\Desktop\11.xls"
        DBMS=EXCEL REPLACE;
    SHEET="11_e";
RUN;
**********************************************************************;
PROC SQL;
CREATE TABLE obs_v AS
        SELECT pid, count(pid) as N, max(decision1) as dec_obs, mode_route
                FROM probdata
                WHERE decision = .
                GROUP BY pid
                HAVING decision1 = calculated dec_obs;
QUIT;
PROC SQL;
CREATE TABLE pred_v AS
        SELECT pid, count(pid) as N, max(p) as dec_pred, mode_route
                FROM probdata
                WHERE (decision = .) AND (p ne .)
                GROUP BY pid
                HAVING p = calculated dec_pred;
QUIT;
PROC SQL;
CREATE TABLE obs_pred_v AS
SELECT A.pid, A.mode_route AS mode_obs, B.mode_route AS mode_pred
FROM pred_v B LEFT JOIN obs_v A
ON (A.pid = B.pid);
QUIT;
PROC SQL;
CREATE TABLE p1_v AS
        SELECT mode_obs, count(mode_pred) as pred1
                FROM obs_pred_v
```

```
                WHERE mode_pred=1
                GROUP BY mode_obs;
QUIT;
PROC SQL;
CREATE TABLE p2_v AS
        SELECT mode_obs, count(mode_pred) as pred2
                FROM obs_pred_v
                WHERE mode_pred=2
                GROUP BY mode_obs;
QUIT;
PROC SQL;
CREATE TABLE p3_v AS
        SELECT mode_obs, count(mode_pred) as pred3
                FROM obs_pred_v
                WHERE mode_pred=3
                GROUP BY mode_obs;
QUIT;
PROC SQL;
CREATE TABLE p4_v AS
        SELECT mode_obs, count(mode_pred) as pred4
                FROM obs_pred_v
                WHERE mode_pred=4
                GROUP BY mode_obs;
QUIT;
PROC SQL;
CREATE TABLE p5_v AS
        SELECT mode_obs, count(mode_pred) as pred5
                FROM obs_pred_v
                WHERE mode_pred=5
                GROUP BY mode_obs;
QUIT;
PROC SQL;
CREATE TABLE p6_v AS
        SELECT mode_obs, count(mode_pred) as pred6
                FROM obs_pred_v
                WHERE mode_pred=6
                GROUP BY mode_obs;
QUIT;
data final_v;
merge p1_v p2_v p3_v p4_v p5_v p6_v;
by mode_obs;
run;
PROC EXPORT DATA= WORK.Final_v
        OUTFILE= "C:\Documents and Settings\kxf7416\Desktop\11.xls"
```

```
        DBMS=EXCEL REPLACE;
    SHEET="11_v";
RUN;
```

# REFERENCES


Ben-Akiva, M., and Lerman, S. R.. (1985). Discrete Choice Analysis: Theory and Application to Travel Demand, The MIT Press, Cambridge, Massachusetts.

Bureau of Transportation Statistics (2005). National Transportation Statistics: 2005, U.S. Department of Transportation. http://www.bts.dot.gov/publications/national_transportation_statistics/2005/

Burris, M. W., Farokhi Sadabadi, K., Mattingly, S. P., Mahlawat, M., Li, J., Rasmidatta, I., and Khaleghi-Soroosh, A. (2007). Reaction to the Managed Lane Concept by Various Groups of Travelers. Paper Submitted for Publication and Presentation at the Transportation Research Board Annual Meeting.

Clemens, J. and Niels V. (2003). Replace harmful HOV lanes with HOT lanes. Vancouver Province, August 15.

Connolly, G. (2005). HOT lanes project should include transit component. Washington Post, August 11.

Cooke, J. A. (2005). Let's liberate the diamond lane. Logistics Management, November, 80.

Copeland, L. (2005). Solo in car-pool lane? That's HOT. USA Today, May 9.

Dahlgren, J. (2002). High-occupancy/toll lanes: Where should they be implemented? Transportation Research A, 36: 239-55.

DeCorla-Souza, P., Angela, J., Shannon, B., and Smith, T. (2003). Paying the value price. Public Roads, September-October, 43-47.

Eckenrode, V. (2005). Transportation experts look for ways to efficiently manage traffic in Atlanta. Florida Times Union, October 19.

Economist. (2000). Tolled you so. June 24, 355(8176): 32-33.

Ferguson, E. (1997). Rise and fall of the American carpool: 1970-1990. Transportation 24: 349-376.

Fuhs, C. and Obenberger, J. (2002). HOV Facility Development: A Review of National Trends. Transportation Research Board HOV Systems Committee. January 2002. http://www.hovworld.com/PDFs/Fuhs_Obenberger-final%20paper.pdf. Final Access: June 29, 2005.

Groat, G. (2004). Loosening the belt: A closer look at how I-495 officials sold the HOT lane concept. Roads and Bridges, April.

Hadley, J. (2004). HOT idea could ease traffic flow. Seattle Post Intelligencer, January 21.

Jaskevich, M. (2001). Mixed reviews for HOV lanes. American City and County, October, 60-68.

Lisco, T. (1967). The Value of Commuter's Travel Tim: A Study in Urban Transportation. Ph.D. dissertation. Department of Economics, University of Chicago, Chicago, Illinois.

Manheim, M. (1979). Fundamentals of Transportation Systems Analysis: Basic Concepts. Vol. 1. MIT Press, Cambridge, Massachusetts.

Manski, C. (1973). The Analysis of Qualitative Choice. Ph.D. dissertation. Department of Economics, MIT, Cambridge, Massachusetts.

Manski, C. (1977). The Structure of Random Utility Models. Theory and Decision 8: 229-254.

McFadden, D. (1974). Conditional Logit Analysis of Qualitative Choice Behavior. Frontiers in Econometrics. P. Zarembka, ed. Academic Press, New York, pp. 105-142.

Owens, E. and Lewis, C. (2002). Meeting Summary: 2002 Annual Project Monitoring Committee Workshop. Research Report 4160-3, Texas Southern University, Houston, TX, November.

Poole, R. and Orski, C. K. (1999). Building the case for HOT lanes: A new approach to reducing urban highway congestion. Policy study 257, April. Los Angeles: Reason Public Policy Institute.

Poole, R. and Balaker, T. (2005). Virtual exclusive busways: Improving urban transit while relieving congestion. Policy study 337, September. Los Angeles: Reason Public Policy Institute.

Poole, R. and Soucie, K. (2006). Adding FAST lanes to Milwaukee's freeways: Congestion relief, improved transit, and help funding reconstruction. Policy Summary of Study 342. Los Angeles: Reason Public Policy Institute.

Regan, E. (2004). Some like it HOT. American City and County, January, 28-32.

SAS Institute Inc. (2006). SAS/ETS Software Manual: Version 9.1.3, Chapter 15: The MDC Procedure, pp. 617-690.

Simon, H. A. (1957). Models of Man: Social and Rational, Mathematical Essays on Rational Human Behavior in a Social Setting, John Wiley and Sons, New York.

Slovic, P., Fischoff, and Lichtenstein, L. (1977). Behavioral Decision Theory, Annual Review of Psychology.

Small, K., Winston, C. and Yan, J. (2006). Differentiated Road Pricing, Express Lanes, and Carpools: Exploiting Heterogeneous Preferences in Policy Design. Working paper, March.

Stockton, W. R., Daniels (Goodin), G., Skowronek, D., and Fenno, D. (1999). The ABC's of HOV: The Texas Experience. TTI, Texas A&M University. September 1999. http://www.hovworld.com/PDFs/1353-I.pdf. Final Access: July 28, 2005.

Svenson, O. (1979). Process Description of Decision Making, Organizational Behavior and Human Performance, vol. 23, pp. 86-112.

Swait, J. (1984). Probabilistic Choice Set Formation in Transportation Demand Models, Ph.D. dissertation, Department of Civil Engineering, MIT, Cambridge, Massachusetts.

Swisher, M., Eisele, W., Ungemah, D., and Goodin, G. (2002). Life Cycle Graphical Representation of Managed HOV Lane Evolution, Committee on High Occupancy Vehicle Systems, 11th International HOV Conference, October 2002. http://www.hovworld.com/PDFs/TRB2003-002138.pdf. Final Access: June 29, 2005.

Toycen, C., et al. (2006) "How to Conduct a Robust Stated Preference Survey without Breaking the Bank: A Case Study of Outreach in Dallas and Houston, Paper Submitted to the Transportation Research Board for Consideration of Presentation and Publication.

Varaiya, P. (2005). What we've learned about highway congestion. Access, Fall, 27:2-9.

Washington, W. B. (2005). Tales from the Dark Side of Traffic Reform. Washington Post, March 6.

# BIOGRAPHICAL INFORMATION

The author holds a BS degree in Civil and Environmental Engineering from the Iran University of Science and Technology, Tehran, Iran, received in February 1999. He received his Master of Science degree in Transportation Planning and Engineering in February 2001 from the Sharif University of Technology, Tehran, Iran. In fall 2005, he started his studies at the Civil and Environmental Engineering Department of the University of Texas at Arlington in the Transportation Engineering field. This thesis is the result of author's position as Graduate Research Assistant during his stay at UTA. Mr. Farokhi Sadabadi's interests include demand analysis, network design, intelligent transportation systems, operations research and applications of artificial intelligence in the transportation field.