Nested Logit Model Analysis of Aggregate Sales Response

by

Weizhong Xiao

M. S. Chemistry
Nanjing University, 1989

SUBMITTED TO THE SLOAN SCHOOL OF MANAGEMENT IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN MANAGEMENT OF TECHNOLOGY

AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 1997

Signature of Author: ______________________________
MIT Sloan School of Management
May 16, 1997

Certified by: ______________________________
John D C Little
Institute Professor
Thesis Supervisor

Accepted by: ______________________________
Rochelle Weichman
Director, Management of Technology Program

Nested Logit Model Analysis of Aggregate Sales Response

by

Weizhong Xiao

Submitted to the Alfred P. Sloan School of Management
on May 16, 1997 in Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Management of Technology

ABSTRACT

A nested logit model (NLM) analysis has been performed to study individual household response to given merchandising conditions. The model assumes that the customer makes purchase decisions on two levels, i.e., the product category level (when a category purchase will occur) and individual brandsize level (what brandsize will be chosen). On the brandsize level, the probability that a customer purchases a particular product is modeled as a function of brandsize loyalty, presence/absence of store advertisement, presence/absence of store display, regular shelf price, and other possible variables. On the category level, the probability that a customer makes the purchase decision on a particular trip is modeled a function of various variables including household inventory, category price, and the attractiveness of buying on that shopping trip as affected by the product choice variables at the time.

Using the marketing data containing purchasing information of bottled juice by more than 2,000 households in Marion, Indiana over a four-year period, we have demonstrated that the nested logit is an excellent model to explain consumer choice behavior in packaged consumer goods market. Model parameters obtained are statistically significant and stable over the entire purchasing period. Although the model is parsimonious in that the major parameters are the same across all the brandsizes and customers, the predicted data track the actual data remarkably well. Combining product purchase and category purchase in a nested fashion allows us to calculate not only the brandsize share but also the actual brandsize sales. On the basis of the model, marketing responses on both individual customer level and aggregate market level have been evaluated.

Thesis Supervisor: John D C Little
Title: Institute Professor

ACKNOWLEDGMENTS

I would like to express my sincere thanks to my thesis advisor, Professor John Little, for his inspiration and guidance throughout this thesis effort. I would also like to extend my appreciation to the Management of Technology Program and the Sloan community for providing excellent services and resources. Finally, I am deeply indebted to my parents and my wife, Min, for their encouragement, deep love and constant support.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

The battle for markets among manufacturers and merchants is extremely intense as a result of the increase in number of global competitors and the decline of sales. According to McCarthy (1993), over $70 billion was spent on marketing in 1991, 50% of which was on trade promotions. Within this competitive environment, marketers not only need to understand how marketing activities affect the sales and shares of the products they sell, but they also need to understand how products interact with each other within the same product category. To better understand these issues, many marketing models have been constructed within the past decade (Lilien, Kotler, and Moorthy, 1992; Hanssens, Parsons, and Schultz, 1990).

In 1983, Guadagni and Little used the Multinomial Logit Model (MNL) to study the determinants of brand choice for the eight major brand-size combinations of coffee on the market assuming that customers make category purchases. Their analysis showed high statistical significance for the explanatory variables of brand loyalty, size loyalty, presence/absence of store promotion, regular shelf price and promotional price cut. The model was later extended to include the decision to make a purchase in the category on a shopping trip (Guadagni and Little, 1987). This extension required generalizing the MNL to the Nested Logit Model (NLM). The addition of the category choice step not only provided a more complete description of the purchasing process, but it also made possible the better evaluation of sales response by including the effect of marketing action on category sales as well as brand share.

The calibration and validation of these models usually require the availability of large amounts of information about customers and merchandising conditions. The automatic recording of purchases by optical scanning of the Universal Product Code (UPC) has made the collection of such massive information possible. The data recorded by a UPC scanner usually come in two forms: store data and panel data. Store data provide sales information for individual UPCs by store and by week. This may also include information on other store activities, such as display, coupon redemption, retail advertising, etc. Panel data record histories of purchases for a panel of households. Members of the panel identify themselves at checkouts so that their purchase records could be stored and accumulated over time.

This thesis analyzes UPC panel and store data according to the nested logit model (NLM) analysis for the bottled drinks market. This analysis assumes that customers make purchase decisions on two levels, the product category level (when a category purchase will occur) and individual brandsize level (what brandsize will be chosen). On the brandsize level, the probability that a customer purchases a particular product is modeled as a function of brand-size loyalty, presence/absence of store promotion, regular shelf price, presence/absence of store display and other possible variables. On the category level, the probability that a customer makes the purchase decision on a particular trip is modeled a function of various variables including household inventory, category price, and the attractiveness of buying on that shopping trip as affected by the product choice variables at the time.

The marketing data used in this research come from four years of purchasing information of bottled juice by more than 2,000 households in Marion, Indiana. Due to computational limitations, only a subset of households are chosen in this research. The model is initialized on the first year and calibrated on the second and third years. The last year is held out for model testing and evaluation. Finally, the model is further evaluated by comparing the forecast and actual sales data by individual stores.

Chapter 2

DISCRETE CHOICE MODELS

Although we are primarily interested in the collective response of a large number of individuals, the modeling of the individual behavior should be at the core of all predictive models of aggregate behavior. This chapter starts with an overview of the steps of the individual decision-making process and the basic elements of a choice problem. We then briefly discuss the probabilistic choice theory and its two distinct approaches, the constant utility approach and the random utility approach. Based on the latter approach, we then give a brief description of binary logit and multinomial logit models. We then give a brief discussion of the maximum likelihood method used to calculate the model parameters. The chapter closes with a brief introduction to various approaches to assessing model quality.

2.1. The Individual Decision-making Process and the Elements of A Choice Problem

In order to understand individual purchasing behavior, it is helpful to first understand the individual decision-making process. When an individual makes a choice, she usually goes through a sequential process that includes the following steps (Ben-Akiva and Lerman, 1985):

1. defining the choice problem,
2. generating alternatives,
3. evaluating attributes of the alternatives,
4. making choice,
5. implementing decision.

The decision-making process starts with the definition of the choice problem. For a shopper, a problem could be that of deciding what to buy in a product category. Depending on her environment and personal experiences, the shopper generates a list of alternatives available to her, that is, a list of the products in a certain category available in a supermarket. In the next step, the shopper seeks information on the attributes of the alternative products. The shopper then choose a product based on certain decision rule and finally she actually makes the purchase.

Thus a process-oriented model of the consumer choice process is a collection of procedures that defines the following elements:

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

The decision maker can be an individual person, or a group of people, or an organization. When we consider a group or an organization as the decision-maker, we assume that it is possible to abstract partially the complex interactions within the group or the organization. When a customer makes a choice, her environment defines a so-called universal set of alternatives. However, the customer only considers a subset of this universal set, called a *choice set*. The latter set consists of only those alternatives that are both known and attainable to the customer. It is worth noting that there are two types of choice sets. In the first type, the choice is continuous such as the quantities of various homogeneous consumption commodities (e.g. gasoline). In the second type, the alternatives are discrete such as the choice between buying or not buying a product category on a shopping trip.

Facing various alternatives, the customer needs a decision rule to make a choice. Various decision rules have been documented in the literature. The most widely used class of decision rules is the hypothesis of *utility maximization*. This class of decision rules assumes commensurability of attributes. In practice, the utility of an alternative for a consumer is often expressed as a linear function of observed attributes. Mathematically, it can be expressed as follows:

$$V_{in} = \beta_1 x_{in1} + \beta_2 x_{in2} + \ldots + \beta_k x_{ink} = \beta' \mathbf{X}_{in} \quad (1)$$

where V_{in} is the so-called deterministic part of utility of alternative i for consumer n (see further discussion below), $\mathbf{X}_{in}$ is the vector of the observed attributes (or predictors) and β is the vector of utility weights for the attributes. The assumption of commensurability is based on the notion of trade-offs that a decision-maker uses in comparing different attributes. One reason for wide application of utility maximization decision rule in many choice theories, both deterministic and probabilistic, is that it results in formulations of choice processes that are amenable to mathematical analysis and statistical applications. In addition, it has been shown to work very well in many discrete choice situations (Ben-Akiva, 1973).

2.2. Probabilistic Choice Theory

In a deterministic choice theory, one expects that an individual would make the same choice for identical choice situations. However, in reality, violations of this theory were often observed. In an attempt to explain these inconsistencies, two different probabilistic choice approaches, the constant utility approach and the random utility approach, were developed. In the former approach, the decision rule is subject to randomness, but individuals' evaluations of the alternatives are assumed constant (Luce, 1959). According to this approach, choice probabilities are defined by the utility functions of the product alternatives that form the individual's choice set. Whereas the random utility approach hypothesizes that the product utility values undergo random fluctuations, while the choice mechanism is deterministic. In this context, the consumer utility can be expressed as a combination of a systematic (deterministic) component and a random component:

$$U_{in} = V_{in} + \varepsilon_{in} \qquad (2)$$

where U_{in} is the total utility of alternative i for consumer n, V_{in} is the deterministic component and ε_{in} is the random component. Therefore, from the perspectives of the random utility approach and the maximum utility decision-making rule, the choice probability of alternative i is equal to the probability that the utility of alternative i, U_{in}, is greater than or equal to the utilities of all other alternatives in the choice set. That is,

$$P_n(i) = Pr(U_{in} \geq U_{jn}) \qquad \text{for all } j \in C_n \qquad (3)$$

where C_n represents the choice set for the consumer n. Combining equations (2) and (3), we have,

$$P_n(i) = Pr[(\varepsilon_{jn} - \varepsilon_{in}) \geq (V_{in} - V_{jn})] \quad \text{for all } j \in C_n \text{ and } j \neq i \qquad (4)$$

This general formula forms the basis for various random utility models including binary logit, multinomial logit and nested logit models as we shall describe in the following sections.

2.3. The Logit Models

In this section, we consider a special case where the probability in eqn. (4) takes the *logistic* form. If we order the alternatives so that the one chosen $i = 1$, then eqn. (4) can be rewritten as follows:

$$P_n(1) = Pr[(\varepsilon_{jn}-\varepsilon_{1n}) \geq (V_{1n}-V_{jn})] \quad \text{for all } j \in C_n \tag{5}$$

$$= Pr[(\varepsilon_{2n}-\varepsilon_{1n}) \geq (V_{1n}-V_{2n}); (\varepsilon_{3n}-\varepsilon_{1n}) \geq (V_{1n}-V_{3n});\ldots$$

$$(\varepsilon_{jn}-\varepsilon_{1n}) \geq (V_{1n}-V_{jn}); \ldots(\varepsilon_{Jn}-\varepsilon_{1n}) \geq (V_{1n}-V_{jn});] \tag{6}$$

The logit model assumes that the random components of the utility are independent and double-exponentially distributed,

$$Pr(\varepsilon_j) = \exp(-e^{-\mu(\varepsilon_j - \eta)}) \qquad \mu > 0 \tag{7}$$

where η is a location parameter and μ is a positive scale parameter. Since the scaling of utility is not identifiable in advance, we can set $\mu = 1$. Also we can assume that all alternatives have a constant η or $\eta = 0$ since doing so is not in any sense restrictive as long as each deterministic utility has a constant term. Then the probability of choosing alternative *i* by customer *n* can be expressed in the following simple form (Ben-Akiva and Lerman, 1985):

$$P_n(1) = \frac{e^{V_{1n}}}{\sum_{j=1}^{J_n} e^{V_{jn}}} \tag{8}$$

$$= \frac{e^{\beta' X_{1n}}}{\sum_{j=1}^{J_n} e^{\beta' X_{jn}}} \tag{9}$$

Eqn. (9) is known as the *multinomial logit.* A special case arises when the choice set C_n only contains two alternatives, say, *1* and *2*. Thus, the probability that decision-maker n chooses alternative *1* becomes,

$$P_n(1) = \frac{e^{V_{1n}}}{e^{V_{1n}} + e^{V_{2n}}} \tag{10}$$

Eqn.(10) is known as *binary logit.* When V_{2n} is held constant, $P_n(1)$ is S-shaped. Therefore, $P_n(1)$ is insensitive to either very large or very small values of V_{1n}. One can also straightforwardly verify the following properties of the multinomial logit:

$$0 \leq P_n(i) \leq 1 \tag{11}$$

$$\sum_{i=1}^{J_n} P_n(i) = 1 \tag{12}$$

$$\frac{P_n(i)}{P_n(j)} = e^{V_{in}-V_{jn}} \tag{13}$$

There are three main criticisms about the logit model. The first arises from the double exponential distribution of ε's. Although Domencich and McFadden (1975) provided some support for the soundness of this distribution, sometimes some other distributions may be more appropriate. The second criticism has to do with the constraint that all ε's have the same scale factor μ. Although the choice of μ is arbitrary since it simply sets the scale of the utilities, the fact that each ε shares the same scaling factor μ implies that the variances of the random components of the utilities are equal. However, the most serious criticism derives from the so-called *Independence from Irrelevant Alternative* (IIA) property of logit models, as shown in eqn. (13). This property states that for a specific individual the relative odds of choosing between any two alternatives is entirely unaffected by the features of any other alternatives. This seemingly simple property sometimes gives rise to erroneous results. The most widely mentioned example of abnormality is the *red bus-blue bus paradox* (Ben-Akiva and Lerman, 1985). Suppose, currently there exists a transportation system consisting of two transit modes, a red bus and a train and commuters choose among them with equal probabilities. Now a blue bus is introduced that is identical in all attributes to the red bus except it is painted in different color. According to the above logit model, the blue bus will have the same choice probability as the red bus and the train. However, in reality, the commuter will most likely treat the two bus modes as a single alternative and the blue bus will draw more commuters from the red bus than from the train.

Many models have been developed to overcome the IIA problem (Ben-Akiva and Lerman, 1985). They can be broadly categorized into the following three approaches: introducing correlation between the ε's explicitly, segmenting decision-makers into homogeneous groups, and assuming a choice hierarchy with different choice mechanisms applying to the different stages (or dimensions) of the hierarchy. In the following chapter, we describe one of the most widely applied hierarchical models — the nested logit.

2.4. Method of Model Estimation — Maximum Likelihood

Maximum likelihood estimation is the standard for estimating logit model parameters. If the distribution of the data is fully specified up to a set of parameter $\beta=\{\beta_1, \beta_2, \ldots \beta_k\}$, and the distribution varies with β in a suitably smooth way, then maximum likelihood methods can always be applied, and resulting estimators have attractive efficiency properties. Consider the likelihood of a sample of N observations. We first define an indicator variable y_{in} so that

$$y_{in} = \begin{cases} 1 & \text{if alternative } i \text{ is chosen by customer } n \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

The likelihood function for a general multinomial choice model is

$$\mathcal{L}^*(\beta_1, \beta_2, \ldots \beta_k) = \prod_{n}^{N} \prod_{i \in C_n} P_n(i)^{y_{in}} \tag{15}$$

where $P_n(i)$ are the choice probabilities as expressed in eqn. (9). Taking the logarithm of eqn.(15), we find the log likelihood function

$$\mathcal{L} = \sum_{n}^{N} \sum_{i} y_{in} \cdot \ln(P_n(i)) = \sum_{n}^{N} \sum_{i \in C_n} y_{in} \cdot (\beta' \mathbf{X}_{in} - \ln \sum_{j \in C_n} e^{\beta' \mathbf{X}_{jn}}) \tag{16}$$

It is noted here that the index n refers to observations rather than the customer. Each parameter in eqn. (16) varies from observation to observation except the coefficient vector β. Our task is to find the best coefficients β by maximizing $\mathcal{L}$. Setting the first derivatives of $\mathcal{L}$ with respect to the coefficients equal to zero, we obtain the first-order conditions:

$$\frac{\partial \mathcal{L}}{\partial \beta_k} = \sum_{n}^{N} \sum_{i \in C_n} y_{in} \cdot (x_{ink} - \frac{\sum_{j \in C_n} e^{\beta' x_{jn}} x_{jnk}}{\sum_{j \in C_n} e^{\beta' x_{jn}}}) \tag{17}$$

Or, in a more compact form,

$$\sum_{n}^{N} \sum_{i \in C_n} [y_{in} - P_n(i)] x_{ink} = 0, \qquad \forall\, k = 1,2,\ldots, K \tag{18}$$

This equation represents a system of K equations which can be solved using various methods. We applied the Newton-Raphson Method to obtain the maximum likelihood solution (Ben-Akiva and Lerman, 1985).

2.5. Model Evaluation

A number of methods have been used to evaluate the quality of the nested logit model in the literature (Guadagni and Little, 1983, 1987). We use the following two approaches. The first one is to generate the t-value for each parameter and the U^2 value for the overall model in the calibration period. The U^2 value provides a measure of uncertainty explained by the model and is defined as

$$U^2 = 1 - \rho^2 = 1 - \mathcal{L}(X) / \mathcal{L}_0 \tag{19}$$

where $\mathcal{L}_0$ is the log likelihood of the null model and $\mathcal{L}(X)$ is the log likelihood of the test model. The null model used in this study assumes that customers choose every product in the choice set with equal probability. A value of $U^2=1$ indicates that the model gives perfect prediction, whereas a value of $U^2=0$ indicates that the model explains nothing new. The second approach is to apply the model to a forecast period (i.e., holdout period). We assess model quality in the forecast period by calculating U^2 and comparing the plots of predicted vs. actual market share and sales.

Chapter 3

THE NESTED LOGIT AND ITS APPLICATION TO PACKAGED CONSUMER GOODS

As described in Chapter 2, the simple multinomial logit model only applies to the discrete choice situations in which the choice set is simple and uni-dimensional. There are, however, many situations where the members of the set of feasible alternatives are combinations of underlying hierarchic choice dimensions. In this chapter, we begin with an overview of the nested logit model which has been widely applied to many hierarchic decision situations. We then describe its application to packaged consumer goods in detail, first on the product choice (or conditional probability) level and then on the category choice (or the marginal probability) level. Finally, we give a brief discussion on the estimation of the nested logit model.

3.1 The Nested Logit

The nested logit is a decision model structured in such a way that logit model is applied at each hierarchic decision stage. Let us begin by considering a product purchase as a two-stage decision-making process (see Figure 1). On a particular shopping trip, the customer first decides whether to buy a particular category or not (that is, when to buy). Having decided to buy a category, she then decides which product to buy (that is, what to buy). Thus the choice set for the shopper consists of category and product two

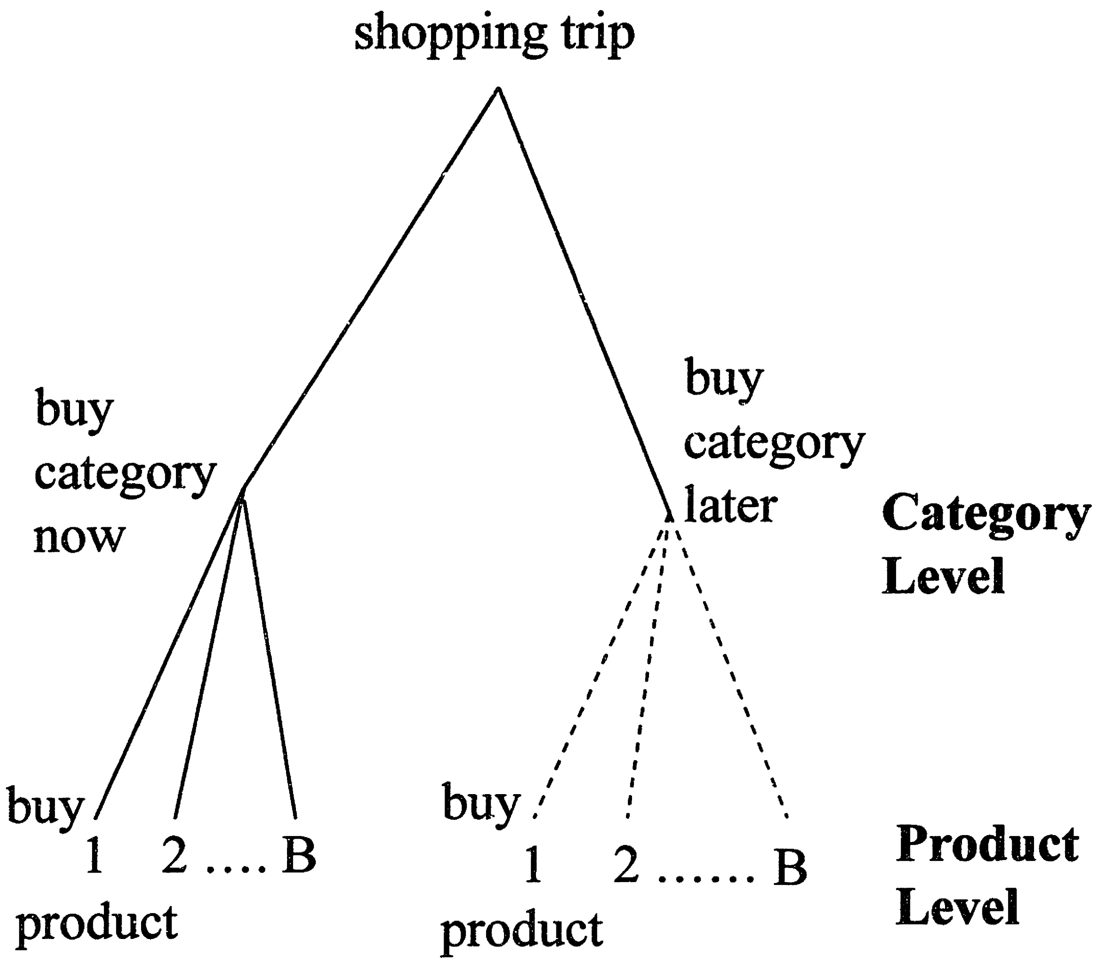


Figure 1. The customer's decision tree on a shopping trip.

dimensions. Let $\{a_1, a_2\}$ represent the category choices buy-now and buy-later and $\{b_1, b_2, \ldots b_J\}$ the products available. With this representation, the customer choice set can be expressed as:

$$C_n = \{ (a_1, b_1), (a_1, b_2), \ldots, (a_1, b_J), (a_2, b_1), (a_2, b_2), \ldots (a_2, b_J) \} \qquad (20)$$

Based on the random utility approach and assuming that the utility associated with categories and products is separable, the utility of a particular choice *(a,b)* to customer *n* can be represented as follows (omitting the customer index *n*):

$$U_{ab} = V_a + V_b + V_{ab} + \varepsilon_a + \varepsilon_b + \varepsilon_{ab} \qquad (21)$$

where V_a, V_b and V_{ab} are the deterministic components of utility related to choosing category *a*, product *b* and category-product combination (*a, b*), respectively. Similarly, ε_a, ε_b and ε_{ab} are the random components of utility for category *a*, product *b* and category-product combination (*a, b*), respectively.

In order to obtain the nested logit form of choice probabilities, we assume the following (Ben-Akiva and Lerman, 1985, Chapter 10):

1. $var(\varepsilon_b)$ is negligible compared to $var(\varepsilon_a)$, thus can be dropped from eqn.(21),
2. ε_a and ε_{ab} are independent for all *a* and *b* in the customer's choice set,
3. The terms ε_{ab} are independently and identically double-exponentially distributed with a scale factor μ_b,
4. ε_a is distributed so that $\max_b U_{ab}$ is double-exponentially distributed with scale factor μ_a.

As in the case of the multinomial logit, we can also set the scale factor μ_b to unity. With these assumptions, one can show that the probability of purchasing product b on the condition that the category choice is a is

$$P(b|a) = \frac{\exp(V_{ab} + V_b)}{\sum_{j=b1}^{b_J} (\exp(V_{aj} + V_j)} \qquad \text{(conditional probability)} \qquad (22)$$

and the probability of purchasing category a is

$$P(a) = \frac{\exp[(V_a + V_a^{'})\mu_a]}{\sum_{i=a1}^{a2} [(\exp(V_i + V_i^{'})\mu_a]} \qquad \text{(marginal probability)} \qquad (23)$$

where

$$V_a^{'} = \ln \sum_{j=b1}^{bJ} \exp(V_j + V_{aj}) \qquad \text{(category attractiveness)} \qquad (24)$$

and the total probability of product purchase is

$$P(b) = P(a) \cdot P(b|a) \qquad (25)$$

As we have seen in Chapter 2, Eqn.(22) is essentially the choice probability for an ordinary multinomial logit. It only involves variables on the product choice level. The probability of category choice would also be the same as the ordinary binomial logit except that it has an additional term from the product choice level $V_a^{'}$, the so-called *inclusive value*. The result is that product utilities also affect the decision at the category choice level. It can be shown that $V_a^{'}$ is the deterministic component of the maximum utility of the subset of product alternatives that involve a. The larger the $V_a^{'}$, the more likely the category is chosen. Therefore, $V_a^{'}$ is also called *category attractiveness.*

Like in Chapter 2, the deterministic components of utility here can also be expressed as linear combinations of various explanatory variables. For our purpose, it is sufficient to ignore the V_{ab} term. Therefore, we have

$$V_a = \delta_1 z_{1a} + \delta_2 z_{2a} + \ldots + \delta_k z_{ka} = \delta' \mathbf{Z}_{ma} \tag{26}$$

$$V_b = \beta_1 x_{1b} + \beta_2 x_{2b} + \ldots + \beta_k x_{kb} = \beta' \mathbf{X}_{jb} \tag{27}$$

where the $\mathbf{Z}_{ma}$ and $\mathbf{X}_{jb}$ are explanatory variables for category choice and product choice, respectively. δ' and β' are coefficient vectors for category and product variables, respectively. In the following two sections, we give a detail presentation of the application of the nested logit to packaged consumer goods, first on the conditional choice level and then on the marginal choice level.

The estimation of the parameters of the nested logit is carried out in sequential steps. First, estimate the parameters β at the conditional probability level (bottom level). Then, calculate the category attractiveness according to eqn.(24). Finally, treat the category attractiveness as a separate variable and estimate the coefficients δ for the category choice level. We note that, despite the notation, μ_a does not depend on *a* and can be absorbed into the coefficient for the category attractiveness.

3.2. Application of Nested Logit to Packaged Consumer Goods

The packaged consumer good markets have been excellent subjects for marketing model analysis for many reasons. First, they are frequently purchased by consumers, so massive data are available for analysis. Second, they are highly promoted by both manufacturers and merchants and price changes are relatively common, generating rich information suitable for model development and testing. Finally, there exist many alternative competing brandsizes.

Conditional Purchase Probability — Product Choice Level

As we have shown above, the conditional probability level model is essentially multinomial logit. The main problem for us to solve at this decision stage is to find out the probability of choosing a particular product by the customer in a product category. In the juice market studied in this thesis, for example, the problem is to find out the probability of purchasing a 48 Oz Ocean Spray Bottled Cranberry Juice or a 46 Oz V8 Canned Vegetable Juice, or some other brandsize combinations assuming that the customer decides to buy the juice category. Therefore, only those trips on which the customer made category purchases were included as observations on the conditional level. A brief discussion of model specifications for the juice market at this choice stage follows.

Observations

Each trip on which the customer buys one unit of an alternative juice brandsize combination represents an observation. In the case where the customer purchases multiple units of an alternative product on the same trip, the multiple purchases were treated as multiple single purchases. We define the dependent variable

$$y_n^i(k) = \begin{cases} 1 & \text{if alternative } i \text{ is chosen by customer } n \text{ on the kth choice occasion} \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

Alternatives

There are potentially many ways to define the alternatives for the juice market. Should different flavors or sizes of the same brand be treated as a different product or be lumped together? With the nested logit model, we can aggregate similarities by defining alternatives at the same level in a hierarchy. However, since the focus of our study is the brand competition, we treat different combinations of size and brand as a product alternatives. Therefore, the Ocean Spray 48Oz Bottled Cranberry Blend and the Ocean Spray 64Oz Bottled Cranberry Blend are considered as two different brandsizes. In the Marion City juice market used in this study, there are more than 300 brandsize combinations. For sake of computation, only the top 15 brandsizes were chosen.

Choice Set:

A choice set represents the set of alternatives that are available to a given household on a particular purchased trip. The number of alternatives included in this set can vary, depending on the time and the store at which the purchase occurred.

Observed Attribute Variables

The observed attribute variables can be divided into variables that are unique to the alternative (*alternative specific*) and those common across all alternatives (*generic*).

(A) Alternative Specific Variables

In the utility function expressed in eqn. (1), we included an additive constant term that is specific to a particular brandsize alternative. This is accomplished by introducing a set of dummy variables, one for each alternative. It is noted that one of the dummy variables has been omitted to avoid singularity in the estimation. These variables are defined as follows:

$$x_{0n}^{i}(k)=\begin{cases}1 & \text{if alternative } i \text{ is chosen}\\ 0 & \text{otherwise}\end{cases} \qquad (29)$$

It is noted that for J alternatives, only J-1 alternative specific constants are required by the model since all that matters are their differences. The omitted variable has an implicit value of zero. The coefficients of these variables (β_0^i) capture the difference in utilities between two alternatives when "all else is equal."

(B) Variables common across all alternatives

The common variables considered in this study are product price, presence or absence of display, presence or absence of product advertising in the store's weekly flier, and customer brandsize loyalty. The coefficients of these variables are the same for all alternatives.

Price. The first variable is the product price. Here the price is measured in dollars per ounce. It is a continuous variable.

$x_{1n}^{i}(k)$ = unit price of brand - size i at time of customer n's kth purchase

Display. The second generic variable is display. The display variable indicates the presence or absence of any type of in-store display for each alternative.

$$x_{2n}^{i}(k) = \begin{cases} 1 & \text{if brand - size } i \text{ is on display at time of customer } n\text{'s } k\text{th purchase} \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

Notice that this variable is a dummy variable and it takes on discrete values of 1 or 0.

Features. The next subset of generic variables includes four variations of advertising in the newspaper or store's weekly flier. Of the four variations,

Feature AA (or feataa): represents store coupons, such as low price, buy-one-get-one free;

Feature A (or feata): indicates big size advertising, probably having a picture of the product;

Feature B (or featb): indicates middle size advertising which may or may not have a picture of the product;

Feature C (or featc): represents small size advertising, usually just a line indicating the product and price.

Like display, all of these four variables, denoted as $x_{3n}^{i}(k)$, $x_{4n}^{i}(k)$, $x_{5n}^{i}(k)$, $x_{6n}^{i}(k)$, are dummy variables, taking a value of either 1 for presence of advertising or 0 for absence of advertising.

Loyalty. The last generic variable is the customer loyalty. This variable captures differences in purchase probabilities over the sampled population. It is defined as the individual's tendency to repurchase a product. It is defined as

$$x_{7n}^{i}(k) = \gamma x_{7n}^{i}(k-1) + (1-\gamma) \bullet \begin{cases} 1 & \text{if customer } n \text{ bought alternative } i \text{ at purchase occasion } k\text{–}1 \\ 0 & \text{otherwise} \end{cases} \quad (31)$$

where γ is termed the loyalty constant. It is noted that $x_{7n}^{i}(k)$ is an exponentially weighted average of past purchases of the same product, and it is non-linearly dependent on γ. We will estimate γ using an iterative method suggested by Fader, Lattin, and Little (1992).

Marginal Purchase Probability — Category Choice Level

Alternatives

The model at the marginal purchase probability stage is essentially a binomial logit. The two alternative choices for the customer is buy-now or buy-later, denoted to be: a=1 and a=2, respectively.

Observations

One observation at the category choice level corresponds to one purchase opportunity. In most cases where a category is not purchased, one observation simply means a single shopping trip. However, in order to handle multiple purchases, we view the purchase of each unit of a product as a separate decision. For example, if a customer purchases one unit of product A on a trip, we would say the trip represents two purchase opportunities: the first when the customer walks into the store, the second immediately after the first purchase. Likewise, if the customer purchases N units of a single product, we would say the trip represents $N+1$ opportunities. When different products are purchased, we treat them as they were purchased on separate trips. The dependent variable is defined as follows:

$$w_n^1(m) = \begin{cases} 1 & \text{if customer } n \text{ makes a category purchase on } m\text{th purchase opportunity} \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

$$w_n^2(m) = \begin{cases} 1 & \text{if customer } n \text{ makes no category purchase on } m\text{th purchase opportunity} \\ 0 & \text{otherwise} \end{cases} \tag{33}$$

Category Purchase Attribute Variables

As we have shown above, the utilities of category purchase can be expressed as a linear function of attribute variables. However, an issue arises that the two alternatives seem quite different: buy-now has hard data associated with it, buy-later is vague and uncertain. Any variable defined for one alternative must have a value for the other (unless the variable is alternative specific, in which case its value for the other is defined as zero). The natural tendency is to make most variables unique to one or the other alternative. However, the power of the model is likely to reside in variables that provide relevant comparisons between buy-now and buy-later. In the rest of the section, we provide a brief description of each variable, with special attention paid to household inventory, category attractiveness, and category price.

Buy-now dummy. This dummy variable is devised to capture the uniqueness of the two alternative buy-now. It takes the following form:

$$z_{0n}^{1}(k)=\begin{cases}1 & \text{buy-now}\\ 0 & \text{otherwise}\end{cases} \tag{34}$$

$$z_{0n}^{2}(m)=0 \qquad \text{buy-later.} \tag{35}$$

As was the case at the conditional level, the alternative specific constant for the buy-later is implicitly zero.

First purchase opportunity dummy. This dummy variable is devised to capture the difference between the first purchase opportunity and later ones. It has the following definition:

$$z_{1n}^{1}(m)=\begin{cases}1 & \text{if customer } n\text{'s mth purchase opportunity is the first purchase opportunity of a shopping trip}\\ 0 & \text{otherwise}\end{cases} \tag{36}$$

$$z_{1n}^{2}(m)=0 \qquad \text{buy-later.} \tag{37}$$

It is noted that in the case of multiple product purchases, each product is treated as if it were purchased on a separate shopping trip. In other words, each product purchase has a first purchase opportunity.

Household Inventory. Since the amount of drinks the household has on hand affects its decision of whether to buy now or buy later, it deserves careful treatment. We will estimate household inventory of drinks at the time of each purchase opportunity. To adjust for differences in consumption rates across households, inventory is measured in weeks of supply. Therefore, we define

$$z_{2n}^{1}(m)=z_{2n}^{1}(m-1)-(t_n(m)-t_n(m-1))+{q_n(m-1)}/{c_n} \qquad \text{(buy-now)} \tag{38}$$

$$z_{2n}^{2}(m)=0 \qquad \text{(buy-later)} \tag{39}$$

where:

$q_n(m)$ = quantity of juice drinks purchased by customer n on n's mth purchase opportunity (ounces)

c_n = n's average consumption rate (ounces / week)

$t_n(m)$ = point in time of customer n's mth purchase opportunity (weeks)

In words, the household's inventory at purchase opportunity m is its value at the previous opportunity increased by any purchases and depleted by estimated consumption between opportunities. Inventories are measured in weeks supply of product. The calculation of consumption rate is based on the household's purchase history. According to the definition, each purchase on a shopping trip increases inventory prior to the decision whether or not to make another purchase.

Category Attractiveness. A particularly important variable is the attractiveness of the category as a whole at the purchase opportunity. It is the expected maximum utility of a product choice as determined from the products available to the customer and their individual utilities in the product choice model. As discussed earlier, this variable equals the natural log of the denominator of the product choice probability.

$$z_{3n}^1(m) = \ln\left\{\sum_{j=b_1}^{b_J} \exp(V_j^n(m))\right\} \quad (40)$$

$$z_{3n}^2(m) = 0. \quad (41)$$

Marketing activities, such as display and advertising, that increase utilities for individual brands increase the value of z_3^1 and so the probability of buying now rather than later. It is noted that because of the multiplicative effect of adding terms to an exponent, a promotion on a brand for which a customer has high loyalty will produce a particularly strong push for buying the category.

Category Price. The next variable is the category price level at the time of the purchase opportunity:

$$z_{4n}^{1}(m) = \sum_{k} p_{nk}^{1}(m) \Big/ N(m) \tag{42}$$

$$z_{4n}^{2}(m) = 0\,. \tag{43}$$

where $p_{nk}^{1}(m)$ is the price of brandsize k in dollars/ounce at the time of the purchase opportunity m, and $N(m)$ is the number of brandsizes available. This variable captures the short-term effect of category price on the decision of category purchase.

Category Loyalty. The last explanatory variable is category loyalty. The variable is defined as

$$z_{5n}^{1}(m) = \sum_{i=1}^{N_{pur}(n)} (1) \Big/ N_{opp}(n) \tag{44}$$

$$z_{5n}^{2}(m) = 0 \tag{45}$$

where $N_{pur}(n)$ and $N_{opp}(n)$ are number of single unit purchase and purchase opportunities, respectively. The category loyalty variable simply reflects the share of buy-now observations over an initialization period.

Basket. This variable is devised to capture the differences between smaller category-specific shopping trips and larger weekly purchases where the household is more likely to make a category purchase. The variable, is defined as the size (in dollar terms) of the bundle of products purchased by the household on a particular shopping trip.

$$z_{6n}^{1}(m) = \text{dollars spent on current shopping trip} \qquad \text{buy-now} \tag{46}$$

$$z_{6n}^{2}(m) = 0 \qquad \text{buy-later} \tag{47}$$

First Category Purchase Dummy. This dummy variable is meant to differentiate the first category purchase from later category purchases on the same purchasing trip. It is defined as

$$z_{7n}^{1}(m) = \begin{cases} 1 & \text{first category purchase, buy-now} \\ 0 & \text{otherwise} \end{cases} \tag{48}$$

$$z_{7n}^{2}(m) = 0 \qquad \text{later category purchases} \tag{49}$$

Chapter 4

MODEL COMPUTATIONS

The computer programs to implement the nested logit model analysis are written in Matlab. The programs are executed in a modular fashion and are very generic in that they can handle arbitrary number of households, brandsizes, displays, features, and any other variables one wants to incorporate in the model. This chapter describes these modules and the relationship between them.

4.1. Organization of Computational Modules

There are a total of six modules used in the whole computational process. They include data preparation, data cleaning, data expansion, nested logit, data reporting, and data plotting (see Chart 1). The modules are executed sequentially. The code of Matlab routines are displayed in Appendix 1.

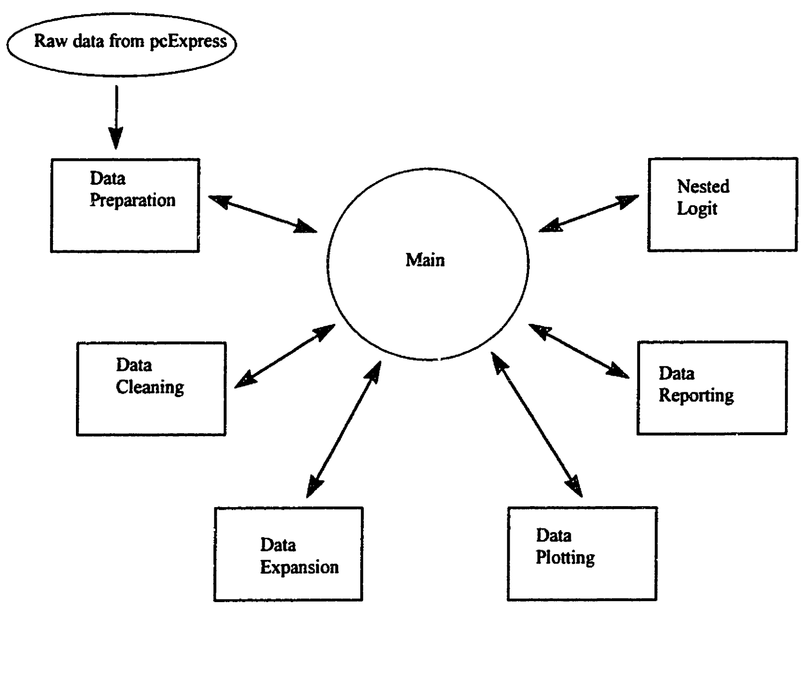


Chart 1. Organization of computational modules

The nested logit module contains many subroutines which can be grouped into two categories, the conditional probability level and the marginal probability level (see Chart 2 and 3).

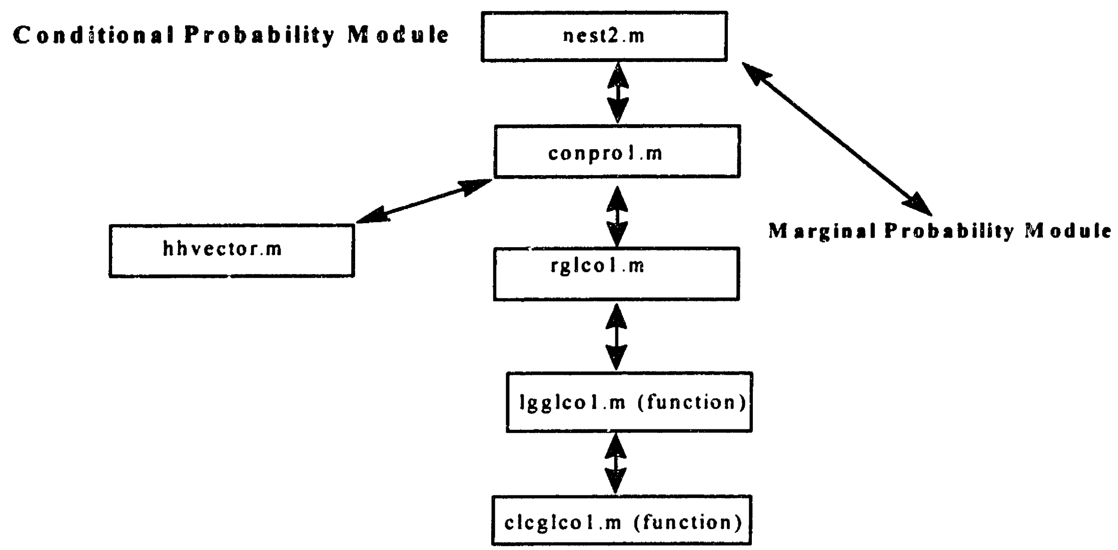


Chart 2. Matlab routines for determining the conditional probability of product purchase.

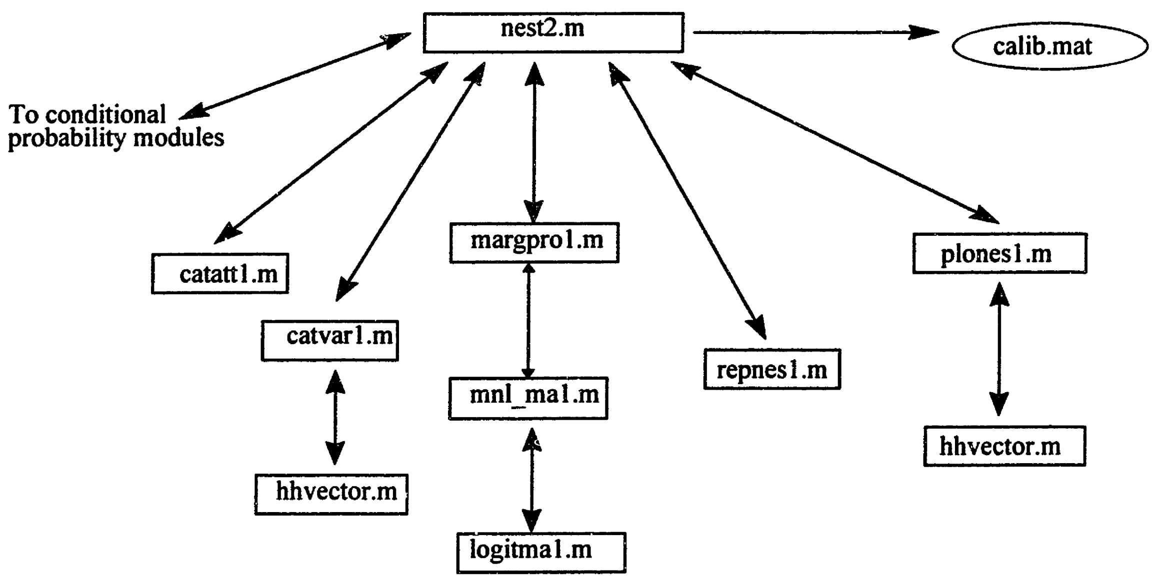


Chart 3. Matlab routines for determining the marginal probability of category purchase.

4.2. Description of Modules

Data Preparation The computational model starts with the data preparation module. The Matlab file for this module is "*dataprep1.m*". This program reads in ASCII data files and converts them into one Matlab data file, "*datprep.mat*". The *datprep.mat* file is used as the input file for the next module, data cleaning.

Data Cleaning The raw data we used lack information on the retail price of products for a significant number of purchase opportunities. The purpose of the data cleaning module is to correct this problem. The module uses "*datprep.mat*" as the input file and the outputs file is called "*clean.mat*". The module modifies the input data in the following ways: replacing the missing price data points with the overall mean category price in cases where the dataset lacks the price information on a non-purchasing brandsize; discarding purchase opportunities for which the price of no item is available. In addition, *clean1.m* eliminates those households who make no category purchases in either the initialization period or in the calibration period or in the forecasting period. This module also allows you to change the length of initialization, calibration, and forecasting period. It also counts the number of clean households and trips.

Data Expansion In order to deal with multiple purchases on a single trip, we expand the cleaned data in the way as discussed in chapter 3. About 90% of the overall computational time is spent in this module due to the mapping processes among several data files. Typically, it takes about 3 hours to expand a dataset containing one-hundred households using a 90 MHz Pentium desktop computer. The result is that the size of expanded dataset is about 1/3 larger than the starting dataset. This module also prepares the data files according to the format specified by the nested logit. For example, the module converts the original basket file which contains one column of data into a file that contains two columns of data. The output file is "*clean.mat*"

Nested Logit. The nested logit module is the most important module in the whole computational process. The module is further divided into two groups: the conditional probability level and the marginal probability level. At the conditional level, the module consists of five subroutines, *condprol.m*, *rglcol.m*, *lgglcol.m*, *clcglcol.m*, and *hhvector.m* (see Chart 2). *condprol.m* serves as the centerpiece where it calls other subroutines and links them together. In evaluating the customer loyalty smoothing constant, γ, we use three subroutines, *rglcol.m*, *lgglcol.m* and *clcglcol.m* following the method proposed by Fader, Lattin and Little (1992). The method provides a simple iterative algorithm for estimating nonlinear parameters at the same time as the usual linear coefficients. The procedure starts with an initial value of γ and calculating an initial loyalty matrix (one row per household, one column per brandsize) based on household purchasing history in the initialization period. The initial loyalty matrix is calculated as follows

$$Loy\_ini_n = \frac{1}{2}(\text{household mkt share}) + \frac{1}{2}(\text{overall mkt share}) \qquad (50)$$

where both household market share and overall market share are estimated in the initialization period. Based on these initial values, the program implements a maximum likelihood estimation using the Newton-Ralphson optimization, resulting in a new value of γ in an inner loop. A new matrix of customer loyalty is then calculated in the outer loop. The value of γ converges rapidly after several iterations.

Seven subroutines were used to carry out the calculation at the marginal level (see Chart 3). *catatt1.m* calculates category attractiveness. *catvar1.m* calculates other category variables including category loyalty, category price, and household inventory. Like the conditional probability level, we use maximum likelihood to estimate the coefficients for the category variables at the marginal probability level. The estimation is implemented using three subroutines, *margpro1.m*, *mn_ma1.m*, and *logitma1.m*. The subroutine *repnes1.m* outputs a report of coefficients for all variables at both the conditional and marginal levels and displays log likelihood values for nested logit. *plotnes2.m* plots nested logit results. At the conditional level, we compare predicted share of each product in a period of four weeks to the actual share calculated from aggregated panel data and store data.

Chapter 5

MODEL CALIBRATION AND TESTING

This chapter presents the results of estimating the parameters of the nested logit model discussed in Chapter 3. In section 5.1 we discuss the sales and merchandising data used in calibration and forecasting stages. This is followed by an examination of the calibration results at both the conditional and marginal choice levels. The chapter closes with a brief discussion of model testing.

5.1. Purchase Data

The data used in this research include store and panel information of bottled juice drinks purchased in Marion, Indiana over a 4 year period. The store data contains weekly store sales and merchandising activities for six supermarkets. The panel data contain the items purchased, the date of the purchase, the store the purchase was made from, and the price paid. In addition, the dataset includes panel information on the household identification number, week, store, and basket size for each purchase opportunity. The data are first extracted from flat files into a pcExpress[1] database. Then files are produced for Matlab by pcExpress programs. Depending on the number of variables included in the model, different number of files can be produced. All of the data files are maintained in matrix form.

Due to computational limitations, only a subset of brandsizes and households are chosen for this research. For most of the calculations, we choose the top 15 selling brandsizes and randomly select predetermined number of juice-purchasing households.

[1] pcExpress is a database management application from IRI.

However, in order for the model to be statistically stable, we used 200 households as our dataset for most of the analysis. Of the 4 years of data available, the first year (week 592 to week 643) is used for various initialization purposes such as obtaining initial loyalty. The second and third years (week 644 to week 747) are used for calibration. The last year (week 748 to 800) is held out for model testing and evaluation.

5.2. Model Calibration

This section presents the calibration results obtained on a dataset consisting of the top 15 selling brandsizes and 200 households. After data cleaning and expansion, only 120 households remain, and the final dataset includes 5974 purchasing events and 28305 purchasing opportunities. Below, we present the calibration results on both the product purchase level and category purchase level using this clean dataset.

Product Purchase Level. As we have described above, the choice on the product purchase level assumes that the customer makes the category purchase. Therefore, only 5974 purchasing events are included in the calibration of this stage. Table 1 presents the coefficient estimates for the observed attribute variables discussed in Chapter 3. For comparison purposes, a smaller dataset containing 3015 purchases and 14355 purchasing opportunities made by 60 households is also presented in Table 1. As may be seen, all the coefficients have the expected sign. It is noted that the coefficient magnitudes per se are not too instructive because of differing units. The better indicators of model quality are their t-statistics because the numerator of t is the coefficient itself, and so it increases as t increases. In addition, the denominator of t is the coefficient's standard error, which

Table 1. Maximum likelihood estimation results for nested logit model: product purchase level

Dataset 5974

Variable					**Coefficients**	**Standard Error**	**t-Value**
price					-59.1294	8.0039	-7.3876
disp					0.9037	0.0643	14.0552
feataa					1.5665	0.1549	10.114
feata					0.7801	0.1235	6.3151
featb					0.4137	0.1135	3.6447
featc					0.5012	0.1823	2.7493
loyalty					5.6028	0.0765	73.2404
PL/GEN	PL	64OZ	BTL	APLJC	0.0919	0.1033	0.8896
PL/GEN	PL	46OZ	CAN	TOM	0.0673	0.0932	0.7219
V8	V8	46OZ	CAN	VEG	0.1672	0.1102	1.5169
GATRD	GATRD	32OZ	BTL	ISO	0.5412	0.1693	3.1971
RDGLD	RDGLD	46OZ	CAN	TOM	-0.1365	0.1026	-1.3297
OSPRY	CRN	48OZ	BTL	CRNCBLD	1.8209	0.274	6.6456
OSPRY	CRN	64OZ	BTL	CRNCBLD	1.6186	0.2436	6.6457
HUNTS	HUNTS	46OZ	CAN	TOM	-0.0905	0.12	-0.7542
HIC	HIC	8.5OZ3CT	BOX	PNCHCHRY	0.7268	0.2239	3.2466
V8	V8	5.5OZ6CT	CAN	VEG	2.4014	0.365	6.5799
OSPRY	GFT	48OZ	BTL	GFTGBLD	0.9623	0.2442	3.9406
SQZIT	SQZIT	6.8OZ6CT	SQZB	PNCHCHRY	1.5009	0.2892	5.1891
GATRD	GATRD	64OZ	BTL	ISO	0.7719	0.1875	4.1158
JCYJC	JCYJC	46OZ	CAN	PNCHCHRY	0.7372	0.1953	3.7748

Loglikelihood value = -5521.34
Null Loglikelihood =-12321.6
U-square = 0.5519
N = 5974 purchasing events

will tend to decrease if the data for the attribute have large variance, assuming all else being equal.

Based on t-statistics, we find brand-size loyalty and display variables most important. The coefficient and t-statistic for brand-size loyalty are greater than those for display. This reflects the fact that establishing and maintaining customer loyalty is more important than arranging displays. The next most important attributes are price, store coupon and large store advertisements. The sign for price is negative reflecting the fact

Table 1. (continued)

Dataset 3015

Variable					Coefficients	Standard Error	t-Value
price					-81.0823	12.3084	-6.5876
disp					1.0729	0.0918	11.6865
feataa					1.1394	0.2360	4.8287
feata					0.5294	0.1823	2.9041
featb					0.2817	0.1589	1.7726
featc					0.4406	0.2503	1.7602
loyalty					5.6019	0.1137	49.2824
PL/GEN	PL	64OZ	BTL	APLJC	0.0693	0.1555	0.4456
PL/GEN	PL	46OZ	CAN	TOM	-0.1412	0.1373	-1.0285
V8	V8	46OZ	CAN	VEG	0.2903	0.1565	1.8310
GATRD	GATRD	32OZ	BTL	ISO	0.7166	0.2565	2.7943
RDGLD	RDGLD	46OZ	CAN	TOM	-0.3305	0.1513	-2.1851
OSPRY	CRN	48OZ	BTL	CRNCBLD	2.5175	0.4186	6.0145
OSPRY	CRN	64OZ	BTL	CRNCBLD	2.1770	0.3731	5.8352
HUNTS	HUNTS	46OZ	CAN	TOM	-0.0111	0.1728	-0.0642
HIC	HIC	8.5OZ3CT	BOX	PNCHCHRY	1.3290	0.3402	3.9060
V8	V8	5.5OZ6CT	CAN	VEG	2.6547	0.5755	4.6127
OSPRY	GFT	48OZ	BTL	GFTGBLD	1.6047	0.3615	4.4384
SQZIT	SQZIT	6.8OZ6CT	SQZB	PNCHCHRY	2.3150	0.4353	5.3188
GATRD	GATRD	64OZ	BTL	ISO	1.2191	0.2761	4.4150
JCYJC	JCYJC	46OZ	CAN	PNCHCHRY	1.1511	0.2842	4.0505

Loglikelihood value = -2587.36
Null Loglikelihood =-6206.85
U-square = 0.58315
N = 3015 purchasing events

that lower prices yield higher probabilities of being purchased. Similarly, the large t-statistics for store coupon and store advertisement indicate that the local store advertisement is an efficient way to sell products. The alternative specific constants (expressed in brandsize names) form a distinct group. They reflect unique product characteristics that can not be explained by common variables. If the other explanatory variables are doing a perfect job, these constants should be close to zero. Note that the constant for the first product (HIC, HIC, 46OZ, CAN, PNCHCHRY) is set to zero for the reason we discussed in Chapter 3. It is also interesting to note that products with the same brand name, such as two Ocean Spray drinks (OSPRY) and two private label drinks (PL/GEN), have similar coefficients and t-statistics. This may reflect the fact that products of the same brand name tend to have similar unique characteristics.

Comparing the results for the larger datasets with those for the smaller ones is revealing. We note that when comparing different datasets, we should understand that we are dealing with different samples. Therefore, marketing variables may have different effect on different samples, resulting in different coefficients. Nevertheless, we assume that both samples represent the overall population so that we should obtain similar results. In fact, the results obtained from both samples are very similar. However, we also notice that most of the alternative specific constants and their t-statistics decrease as the size of the dataset increases while the coefficients and the t-statistics for common variables increase with the size of the dataset. This is comforting to us since more uncertainty is explained by common attribute variables across households, a sign of a more powerful model. In addition, we notice that as sample size increases, the different kinds of store advertisement become more differentiated. This is what we expect since a buy-one-get-one-free store coupon (feature aa) has proved to be much more effective than a large advertisement (feature a) in the store newspaper. The results discussed above are confirmed by analysis performed on several other different datasets.

Category Purchase Level. At the category purchase level, the model studies the probability of purchasing a product if the customer is given a purchasing opportunity. The category choice model is initialized and calibrated on the same households and time periods as the product choice. The explanatory variables are exactly the same as we have described in Chapter 3. The model estimates are displayed in Table 2. As may be seen,

Table 2. Maximum likelihood estimation results for nested logit model: category purchase level

Dataset 5974

Variable	Coefficients	Standard Error	t-Value
category attractiveness	-0.0921	0.0493	-1.8674
category loyalty	3.1548	0.1677	18.8089
household inventory	-0.0024	0.0020	-1.1725
category price	-201.0713	49.8898	-4.0303
basket	0.2801	0.0362	7.7281
first_buy	24.5260	369.1585	0.0664
buy-now dummy	5.0723	1.6981	2.9871

Loglikelihood value = -261.07
Null Loglikelihood = -15225.67
U-square = 0.84493
N = 5974 purchasing events
N =28305 purchasing opportunities

Dataset 3015

Variable	Coefficients	Standard Error	t-Value
category attractiveness	-0.1186	0.0720	-1.6471
category loyalty	3.2487	0.2701	12.0260
household inventory	0.0018	0.0031	0.5906
category price	-275.3130	71.6660	-3.8416
basket	0.2097	0.0502	4.1756
first_buy	24.5726	516.3943	0.0476
buy-now dummy	7.4670	2.4138	3.0934

Loglikelihood value = -1157.11
Null Loglikelihood = -7727.9
U-square = 0.85027
N = 3015 purchasing events
N =14355 purchasing opportunities

the most important variable at the category purchase level is category loyalty, followed by basket size and category price. These results indicate that purchasing behavior tends to be fairly stable within household but differ from one households to another; large grocery purchases tend to include bottled juice drinks. Comparison of the results from different datasets reveals that only statistically significant ($t>2$) variables maintain their algebraic signs as size of dataset changes, as is displayed in Table 2 for household inventory. This is also true for category attractiveness. The fact that the category attractiveness and household inventory are statistically insignificant is a little unexpected.

5.3. Model Testing

As we have discussed in Chapter 2, we shall use two criteria, U^2 and the comparison between actual vs. predicted purchases, to evaluate the model quality. U^2 measures the amount of uncertainty explained relative to a null model. It measures the goodness-of-fit of the model to the actual data. Comparing actual vs. predicted purchases in both calibration and holdout periods poses the ultimate challenge to the model. Like model calibration, model testing for nested logit is also carried out on both the product purchase level and category purchase level.

<u>Product Purchase Level</u> At the product purchase level, the null model against which the model is compared assumes that customers purchase all products with equal probability no matter how product attractiveness varies. The U^2 values for both the smaller and larger datasets are shown in Table 1. It is noted that U^2 improves dramatically as the sample size increases from 50 households to 100 households, and beyond 100 households it tends to stabilize around 0.55. This indicates that the model improvement beyond 100 households is marginal.

The comparison between actual vs. predicted purchases at the product choice level is performed by tracking market share of purchases by 4-week periods. Figure 2 presents some sample plots of the actual and predicted shares. The plots for rest of the brandsizes are included in Appendix 2. In order to account for sampling variations due to the relatively small number of purchases in each week, we have plotted 90% confidence intervals. These confidence intervals are computed based on the assumption that the actual purchase is binomially distributed with the probability given by the model (Guadagni & Little, 1983). Mathematically, the standard error is calculated as follows:

$$s = \sum_{i=1}^{n} p_i / n \tag{50}$$

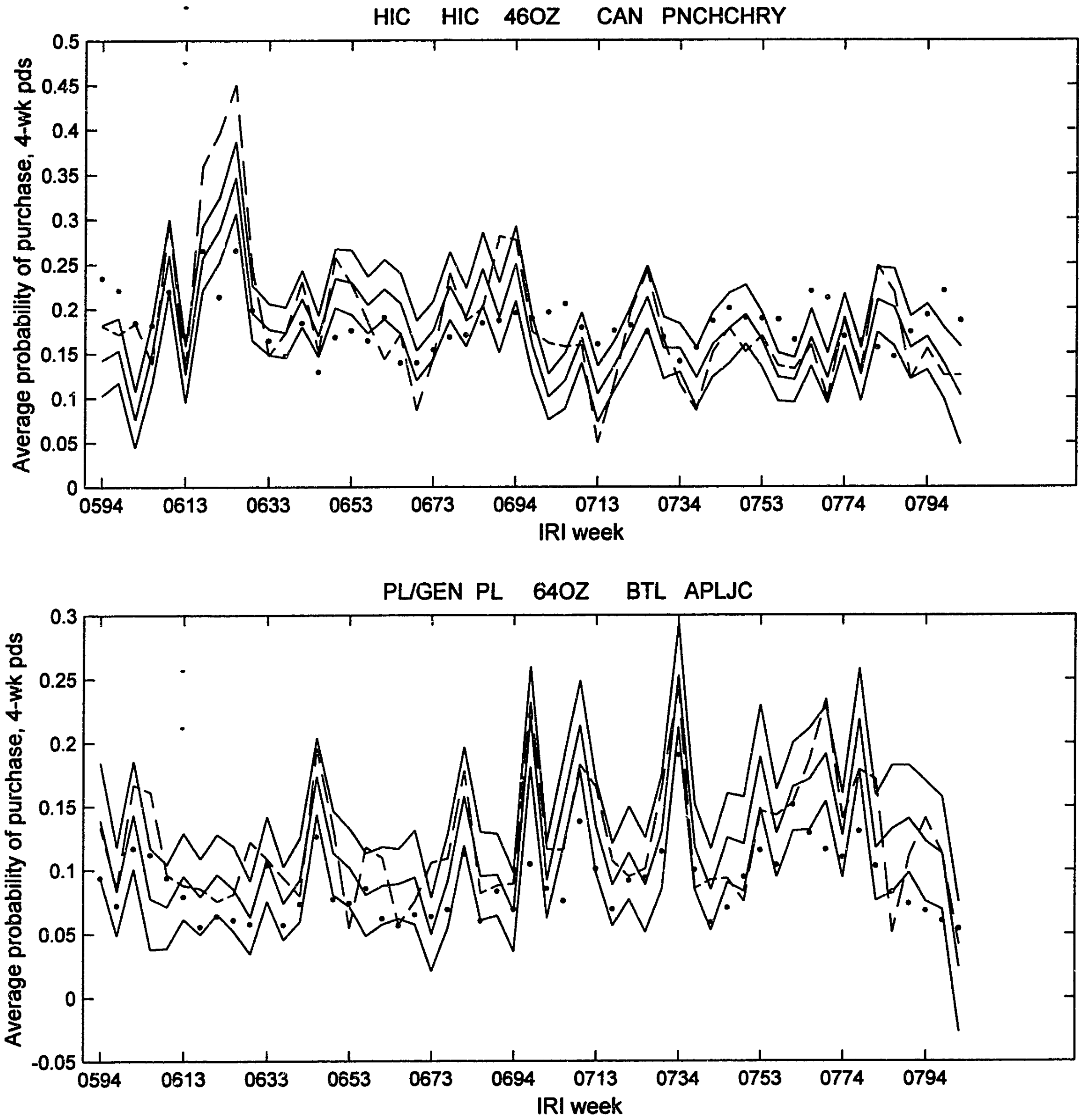
$$SE(s) = \left[\sum_{i=1}^{n} p_i (1 - p_i) \right]^{1/2} / n \tag{51}$$

where *SE(s)* is the standard error of share and p_i is the probability of purchase.

As may be seen, the tracking quality of the model is remarkably good, even into the forecast period. The model can capture most ups and downs in market shares for all products. In most cases, the share changes are followed within the 90% confidence interval. It is noted that in the forecast period, we continue to use the actual prices and other marketing variables of all brandsizes. However, a dilemma arises when we calculate the loyalty variable. We use purchases to calculate loyalty whereas the purpose of the holdout sample is to predict purchases. Here we simply use the actual purchases just for the construction of loyalty. In Figure 2, we have also plotted actual store shares to look at the representativeness of the panel data. In most of the cases, the actual panel share matches the actual store share very well.

Category Purchase Level At the category purchase level, the null hypothesis assumes that the customer chooses the buy-now or the buy-later option with equal

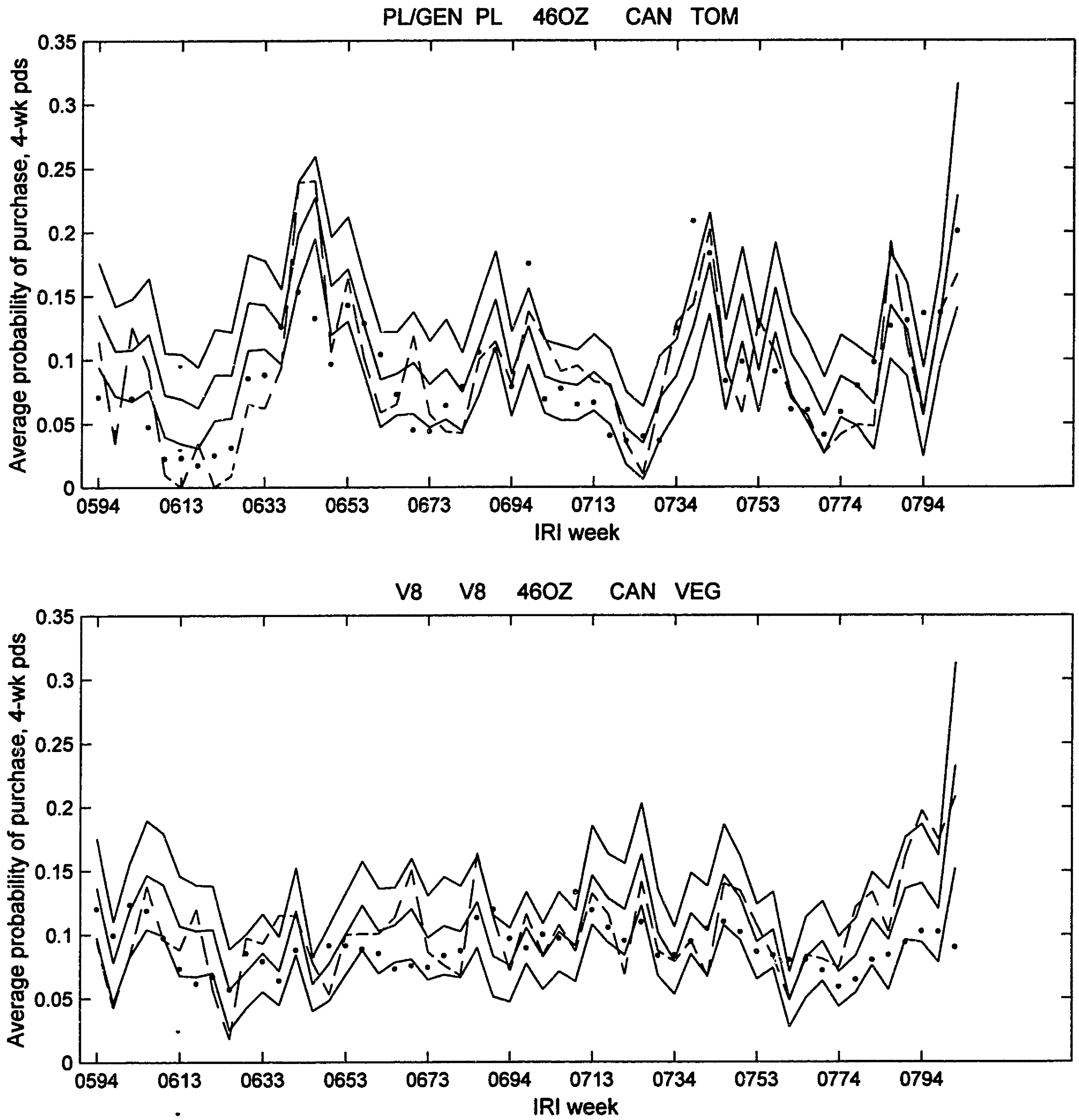
Results of the Nested Model at the Conditional Level
5974 purchases



"-" is predicted share and 90% CI "--" is actual panel share
".." is actual store share

Figure 2. Tracking of purchase share of two sample brandsizes.

Results of the Nested Model at the Conditional Level
5974 purchases



"-" is predicted share and 90% CI "--" is actual panel share

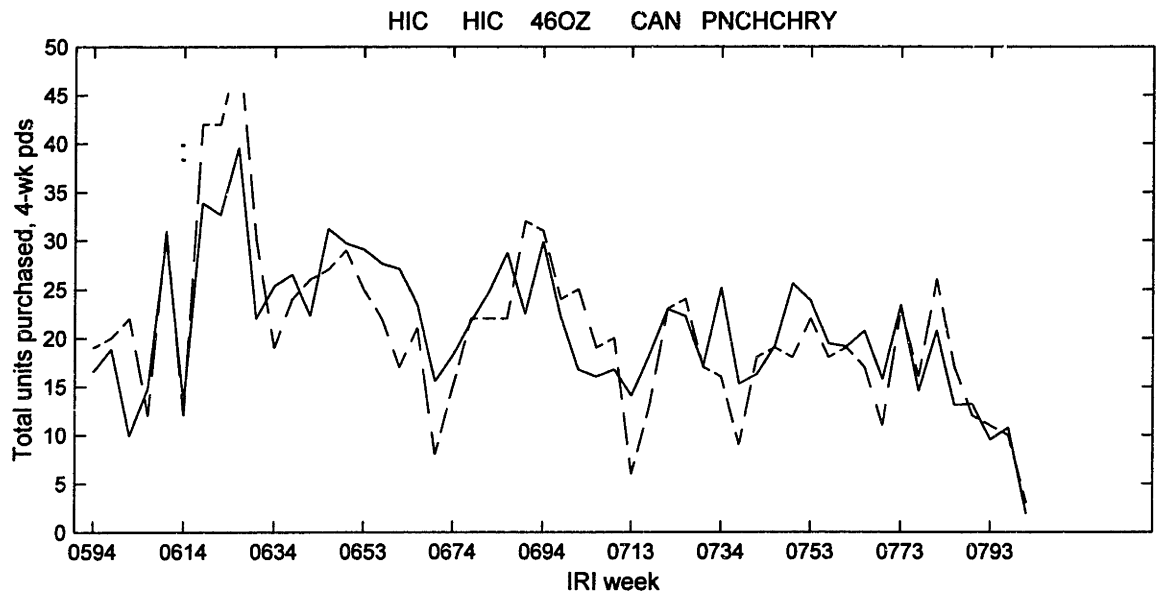
".." is actual store share

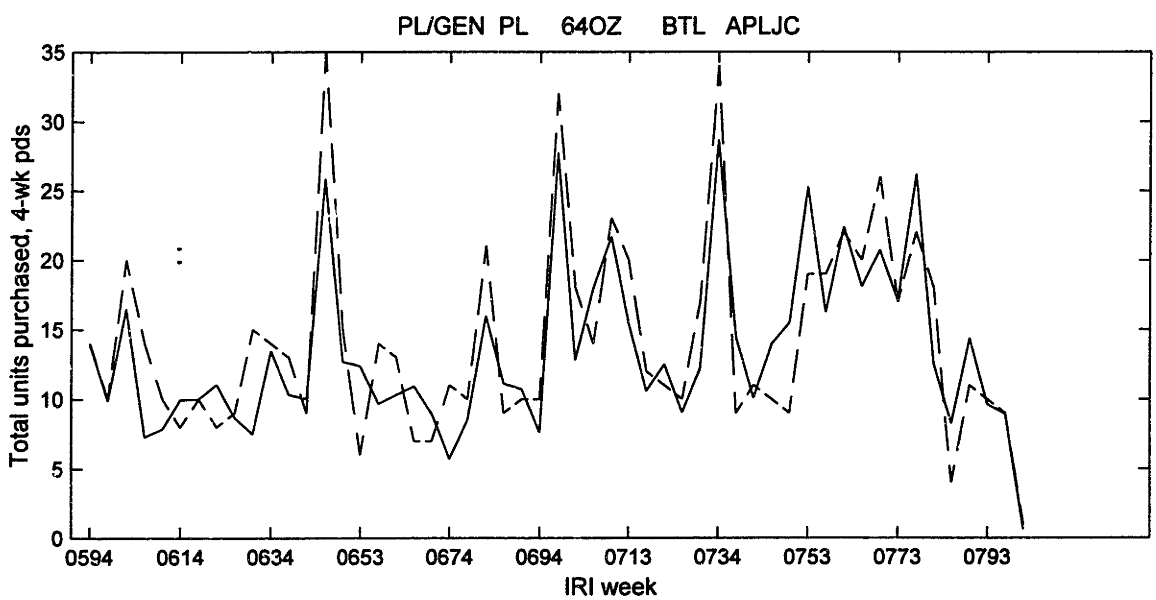
Figure 2. (continued)

probability given a purchase opportunity. The U^2 values at the category purchase level are listed in Table 2. As is in the case of product purchase, U^2 improves remarkably when the number of households increases from 50 to 100, but it stabilizes around 0.85 when the number goes beyond 100.

The tracking quality at category purchase level is assessed by comparing the actual vs. predicted sales (number of units in a 4-week period). Plots of actual vs. predicted sales for four brandsizes are presented in Figure 3, with the rest of plots shown in Appendix 3. Overall, the predicted sales match the actual sales quite well in both calibration and forecasting periods. Most of the twists and turns in sales are captured by the model. However, the fit is generally not quite as good as at the product choice level. This is not surprising since the model is dealing with much more variables than in the former case. The shape of the plots are also noticeably different from what we have seen at the product purchase level since variables at the category level are now playing roles in affecting brand-size sales.

Results of the Overall Nested Model
2.83e+004 purchases opportunities

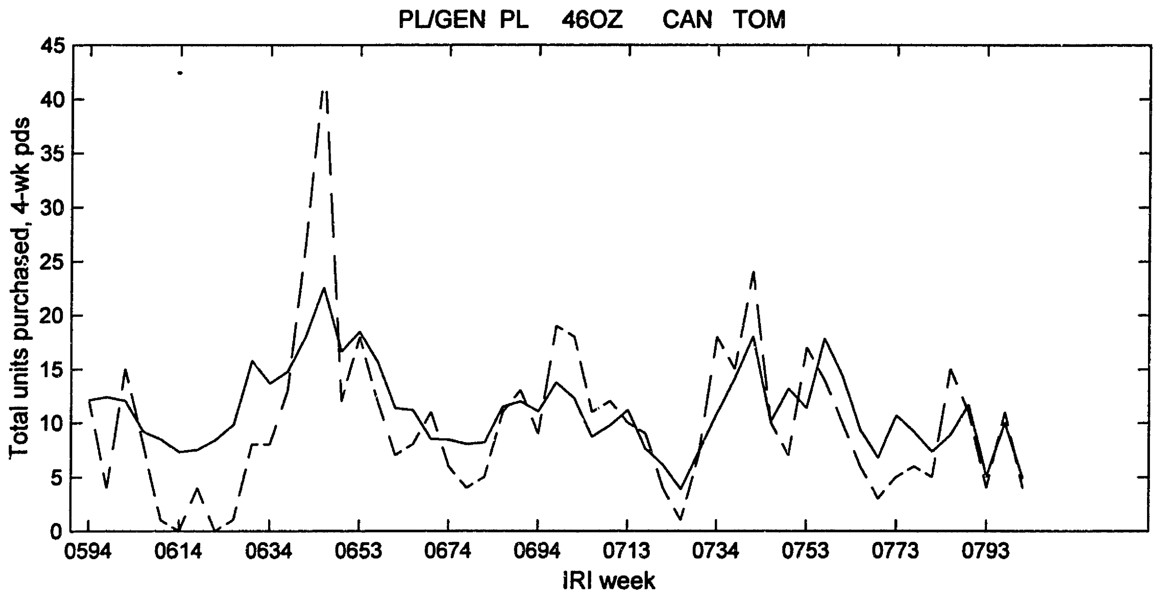


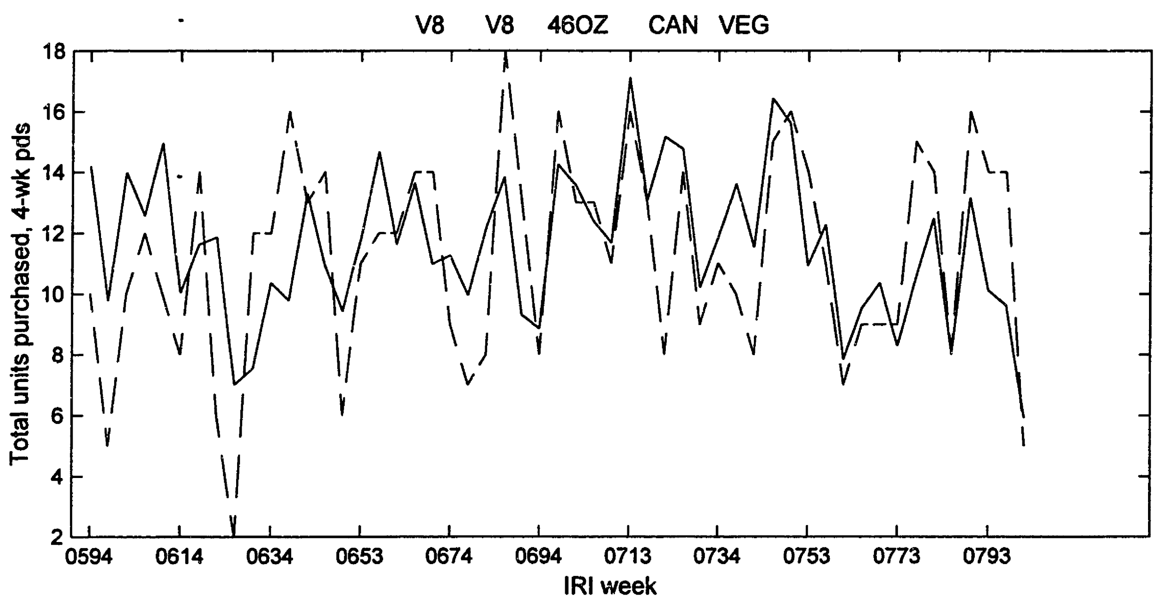


"-" is predicted sales level "--" is actual sales level

Figure 3. Tracking of sales of two sample brandsizes.

Results of the Overall Nested Model
2.83e+004 purchases opportunities





"-" is predicted sales level "--" is actual sales level

Figure 3 (continued).

Chapter 6

MARKET RESPONSE ANALYSIS AND CONCLUSIONS

Consumer product manufacturers wish to understand how their marketing activities will affect customer purchases and consequently the company's bottom line. The nested logit model provides an excellent tool to fulfill this task since it considers the effect of marketing activities not only on market share but also on total sales. We begin this chapter by demonstrating individual customer response to a store coupon in a hypothetical market with only two products. We then discuss short-term market response, both in terms of price elasticities of sales change and share change for 7 of the 15 brandsizes studied in this research. Wc closes the chapter with conclusions derived from this research.

6.1. Individual Customer Response

Although the coefficients obtain from nested logit are the same for all customers, individuals respond differently to marketing activities depending on prior loyalties and store environment. We illustrate this by examining customer's response to a store coupon in a hypothetical market with two products. The results are shown in Table 3. As may be seen, if we assume all else being equal, placing a store coupon has much larger effects on a customer with equal loyalties than on a customer with distinct loyalties. This is consistent with experience in marketing practice. Understanding the individual customer responses are at the core of building the aggregate market response.

Table 3. Individual response to store coupons in a hypothetical market with two Products

Assumptions
Two products: A & B
Loyalty coefficient: 5.0
Coefficient for strore coupons (feataa): 1.0
No other attributes

Case 1: Store coupon response of customers with equal loyalties
Loyalty to A = loyalty to B = 0.5
Without store promotion, coupon A = coupon B = 0
Probability of choosing A = $e^{2.5}/(e^{2.5}+e^{2.5}) = 0.5$
With store promotion, coupon A = 1, coupon B = 0
Probability of choosing A $= e^{3.5}/(e^{3.5}+e^{2.5}) = 0.7310$
Δ Probability of choosing = 0.2310

Case 2: Store coupon response of customers with distinct loyalties
Loyalty to A = 0,8, loyalty to B = 0.2
Without store promotion, coupon A = coupon B = 0
Probability of choosing A = $e^{4}/(e^{4}+e^{1}) = 0.9526$
With store promotion, coupon A = 1, coupon B = 0
Probability of choosing A $= e^{5}/(e^{5}+e^{1}) = 0.9820$
Δ Probability of choosing = 0.0294

6.2. Aggregate Market Response

Traditionally, aggregate market response is determined by integrating the customer response function over a joint distribution of customer loyalties, prices, displays, and other marketing variables (Simon, 1982; Mahajan and Muller, 1986). With the nested logit model, we can accomplish this task in a very straight forward way. For example, by changing the price of a brandsize by 1% over the entire time period and calculating the change in market share, we obtain an aggregate share response to the price cut. Moreover, with nested logit model, we can further calculate aggregate sales response to the price cut. Table 4 summarizes price elasticities of both share and sales. For sake of computational time, we performed the analysis on a 100-household dataset

Table 4. Short-term price elasticities of brandsize share and sales

Brandsizes	Purchase share (%)	Price elasticity of share	Price elasticity of sales
HIC, HIC, 46OZ, CAN, PNCHCHRY	0.2755	-0.1101	-0.1827
V8, V8, 46OZ, CAN, VEG	0.1054	-0.7299	-0.7612
PL/GEN, PL, 64OZ, BTL, APJLC	0.0940	-0.5957	-0.6571
PL/GEN, PL, 46OZ, CAN, TOM	0.0779	-0.7095	-0.7759
GATRD, GATRD, 32OZ, BTL, ISO	0.0601	-1.1106	-1.2291
JCYJC, JCYJC, 46OZ, CAN, PNCHCHRY	0.0238	-1.7859	-1.9718
V8, V8, 5.5OZ6CT, CAN, VEG	0.0081	-6.8750	-6.4583

and only choose 7 brandsizes. In calculating elasticities, we first assume a 1% price increase for the product in study and hold other variables constant. We then calculate the predicted purchase share and sales for the entire purchasing period using the model parameters obtained in the calibration period. The percent change in predicted purchase share or sales divided by the percent change in price is taken as the short-term price elasticities. As may be seen, the response varies very widely among brandsizes. This is not surprising since each brandsize has its own complete set of marketing variables and brand loyalty. Variations in price elasticity suggest that different pricing policies should be implemented across brandsizes.

Further examination of price elasticity reveals interesting response patterns by different brandsizes. Table 4 have been arranged so that the brandsizes are listed in order of decreasing share of purchases. It can be seen that the elasticity tends to increase as share decreases. This relationship between sensitivity and share is embedded in the structure of the logit model, as was discussed previously by Guadagni and Little (1983). As also may be seen, total brandsize sales shows a higher response than brandsize share. The reason for this is that promotion expands category sales in the short run (Guadagni & Little, 1987).

6.3. Conclusions

Using both store and panel data, we have demonstrated that the nested logit is an excellent model to explain consumer choice behavior. Model parameters obtained are statistically significant and stable over the entire purchasing period. Although the model is parsimonious in that the major model parameters are the same across all the brandsizes and customers, the predicted data track the actual data remarkably well over the whole purchasing period. Combining product purchase and category purchase in a nested fashion has its unique advantages. It allows us to forecast both brandsize share and total brandsize sales. The model can also allow us to analyze market response to various changes in marketing variables. Our analysis on market response to price cut has shown that larger share brandsizes tend to have weaker responses than smaller share brandsizes.

One of the major goals of this thesis is to develop and implement a generic marketing decision model. The model developed in this thesis can, in principle, incorporate any number of new marketing variables (except loyalty) and households in a fully automated fashion. It takes into account multiple purchases and allows users to perform market response analysis. However, the model can be improved in several areas. First, the program is not very efficient in handling large datasets due to many looped operations. Second, there are some occasions in which the model prediction deviates from the actual purchases. We believe that these deviations can be captured by introducing new variables such as manufacturer's coupons.

BIBLIOGRAPHY

Ben-Akiva, Moshe & Lerman, Steven. (1995). Discrete Choice Analysis. The MIT Press. Cambridge: MA.

Den-Akiva, Moshe. (1973). Structure of Passenger Travel Demand Models. Ph.D. dissertation. MIT. Cambridge: MA.

Domencich, T. A. & McFadden, Daniel. (1975). Urban Travel Demand: A Behavioral Analysis. Amsterdam: North Holland.

Fader, Hardie; Lattin, Abe; & Little, John. (1992). Estimating nonlinear parameters in the multinomial logit model. Marketing Science 11 (4). 372-385.

Guadagni, Peter & Little, John. (1983). A logit model of brand choice calibrated on scanner data. Marketing Science 2 (3). 203-238.

Guadagni, Peter & Little John. (1987). A nested logit model of coffee purchase. Working Paper. MIT Sloan School of Management. Cambridge: Massachusetts.

Hanssens, Dominique; Parsons, Leonard; & Schultz, Randall. (1990). Market Response Models: Econometric and Time Series Analysis. Kluwer. Boston: Massachusetts.

Lilien, Gary; Kotler, Philip; & Moorthy, K. Sridhar. (1992). Marketing Models. Prentice Hall, Inc.. Englewood: New Jersey.

Luce, Duncan. (1959). Individual Choice Behavior. Wiley Publishing Co.. New York: New York.

McCarthy, Michael. (1993). Getting tough on trade promotion. Adweek, April 13.

Simon, Hermann. (1982). ADPULS: An advertising model with wearout and pulsation. Journal of Marketing Research 19. 352-363.

Mahajan, Vijay & Eitan Muller. (1986). Advertising pulsing policies for generating awareness for new products. Marketing Science 5(2). 86-106.

APPENDIX 1. Matlab routines for nested logit calculations

```
%dataprep1.m-- Prepares dataset before running clean.m
% last updated on 5/3/97
outfile='c:\wayne\nest\marion2\datprep.mat';
fprintf('\nPreparing the data...\n\n');

infile1='c:\wayne\nest\marion2\basket.dat';
eval(['load ' infile1 ]);

infile2='c:\wayne\nest\marion2\cs.dat';
eval(['load ' infile2 ]);

infile3='c:\wayne\nest\marion2\disp.dat';
eval(['load ' infile3 ]);

infile4='c:\wayne\nest\marion2\feata.dat';
eval(['load ' infile4 ]);

infile5='c:\wayne\nest\marion2\feata.dat';
eval(['load ' infile5 ]);

infile6='c:\wayne\nest\marion2\feataa.dat';
eval(['load ' infile6 ]);

infile7='c:\wayne\nest\marion2\featb.dat';
eval(['load ' infile7 ]);

infile8='c:\wayne\nest\marion2\featc.dat';
eval(['load ' infile8 ]);

infile9='c:\wayne\nest\marion2\hhwk.dat';
eval(['load ' infile9 ]);

infile10='c:\wayne\nest\marion2\store.dat';
eval(['load ' infile10 ]);

infile11='c:\wayne\nest\marion2\price.dat';
eval(['load ' infile11 ]);

%infile12='c:\wayne\nest\marion2\tcs.dat';
%eval(['load ' infile12 ]);

infile13='c:\wayne\nest\marion2\tdisp.dat';
eval(['load ' infile13 ]);

infile14='c:\wayne\nest\marion2\tfeata.dat';
eval(['load ' infile14 ]);

infile15='c:\wayne\nest\marion2\tfeataa.dat';
eval(['load ' infile15 ]);

infile16='c:\wayne\nest\marion2\tfeatb.dat';
eval(['load ' infile16 ]);

infile17='c:\wayne\nest\marion2\tfeatc.dat';
eval(['load ' infile17 ]);

infile18='c:\wayne\nest\marion2\tprice.dat';
eval(['load ' infile18 ]);

infile19='c:\wayne\nest\marion2\trip.dat';
eval(['load ' infile19 ]);

infile20='c:\wayne\nest\marion2\tyy.dat';
eval(['load ' infile20 ]);

infile20='c:\wayne\nest\marion2\unitsvol.dat';
eval(['load ' infile20 ]);

infile21='c:\wayne\nest\marion2\yy.dat';
eval(['load ' infile21 ]);


clear infile1;
clear infile2;
clear infile3;
clear infile4;
clear infile5;
clear infile6;
clear infile7;
clear infile8;
clear infile9;
clear infile10;
clear infile11;
%clear infile12;
clear infile13;
clear infile14;
clear infile15;
clear infile16;
clear infile17;
clear infile18;
clear infile19;
clear infile20;
clear infile21;

eval(['save ',outfile]);
eval(['clear ']);


%clean1.m - Clears data of NaN. Use before running nested.m
% last updated on 5/3/97

infile='c:\wayne\nest\marion2\datprep.mat';
eval(['load ', infile]);
outfile='c:\wayne\nest\marion2\clean1.mat';

fprintf('\nCleaning the data...\n\n');

namgen;
volume=unitsvol;
[s1 s11]=size(namvar);
namvar(1,:)=[];
namvar(s1-1,:)=[];
[s1 ss1]=size(namvar);
[s2 ss2]=size(nampvar);
[s3 ss3]=size(namtvar);


% transform tyy first
s=size(tyy,1);
k=0;
for i=1:s
  if tyy(i+k,1)>1
    for j=1:tyy(i+k,1)-1
          for m=1:s3
            temp=eval(namtvar(m,:));
       temp=[temp(1:i+k,:);temp(i+k,:);temp(i+k+1:s+k,:)];
            eval([namtvar(m,:) '=temp;']);
          end;
          trip=[trip(1:i+k,:);trip(i+k,:);trip(i+k+1:s+k,:)];
```

```
end;
Idx1=find(tyy>1);
tyy(Idx1)=ones(size(Idx1));

row_ini=size(tyy,1);
numhh_ini=size(hhvector(trip(:,1)),1);

fprintf('1) Checking entries in price and tprice...\n');

% If NaN in price, but UPC not purchased, then set the
corresponding
%   cs entry to zero, and set corresponding price and tprice to
0.0251
% If NaN in price, and UPC is purchased, then delete the
corresponding
%   row from all purchase and trip matrices.

[Ipr Ipc]=find(isnan(price)); %Row and col index for NaNs in
price
Idx=find(tyy(:,1)); %Index of trip purchases
Irem=[];

for i=1:size(Ipr,1)
 t=yy(Ipr(i),Ipc(i)); %Test for purchase

 if ~t                %NaN wasn't purchased
  cs(Ipr(i),Ipc(i))=0;             %set cs to zero
  price(Ipr(i),Ipc(i))=0.0251;          %set price to 0.0251
  tprice(Idx(Ipr(i),:),Ipc(i))=0.0251;       %set tprice to
0.0251

 else                %NaN was purchsed
  Irem=[Irem;Ipr(i)];
 end;
end;

%Delete corresponding rows


for m=1:s2
temp=eval(nampvar(m,:));
temp(Irem,:)=[];
eval([nampvar(m,:) '=temp;']);
end;
hhwk(Irem,:)=[];
volume(Irem,:)=[];

for m=1:s3
temp=eval(namtvar(m,:));
temp(Idx(Irem),:)=[];
eval([namtvar(m,:) '=temp;']);
end;
trip(Idx(Irem),:)=[];


Itpr=find(sum(isnan(tprice'))==s1); %NaN for whole category
for m=1:s3
temp=eval(namtvar(m,:));
temp(Itpr,:)=[];
eval([namtvar(m,:) '=temp;']);
end;
trip(Itpr,:)=[];

Itpc=find(isnan(tprice)); %remaining NaN
tprice(Itpc)=0.0251*ones(size(Itpc),1);
          k=k+1;
    end;
  end;
row_1=size(tyy,1);
numhh_1=size(hhvector(trip(:,1)),1);

fprintf('\t Got rid of %g HHs, %g rows...\n',numhh_ini-
numhh_1,row_ini-row_1);

fprintf('2) Checking purchases during initialization...\n');


thh=trip(:,1);
hh=hhwk(:,1);
week=hhwk(:,2);

init=min(week)+ 0.25*(max(week)-min(week));
I_init=find(init>week);

hh_ini=hhwk(I_init,1); %these are the HHs that purchased
something
                 % during the initialization period.

rem=[];          %list of rows to remove
trem=[];         %list of trows to remove

for i=1:size(hh)
 if ~(any(hh_ini==hh(i,:)))
   rem=[rem;i];
 end;
end;

for i=1:size(thh)
 if ~any(hh_ini==thh(i,:))
   trem=[trem;i];
 end;
end;

%rem=find(~any(hh_ini==hh));
%trem=find(~any(hh_ini==thh));

[s1 ss1]=size(namvar);
[s2 ss2]=size(nampvar);
[s3 ss3]=size(namtvar);

for m=1:s2
temp=eval(nampvar(m,:));
temp(rem,:)=[];
eval([nampvar(m,:) '=temp;']);
end;
hhwk(rem,:)=[];
volume(rem,:)=[];

for m=1:s3
temp=eval(namtvar(m,:));
temp(trem,:)=[];
eval([namtvar(m,:) '=temp;']);
end;
trip(trem,:)=[];

row_2=size(tyy,1);
numhh_2=size(hhvector(trip(:,1)),1);

fprintf('\t Got rid of %g HHs, %g rows...\n',(numhh_1-
numhh_2),(row_1-row_2));
```

```
clear Itpr Itpc Ipr Ipc Idx Irem t Idx I temp s1 ss1 s2 ss2 s3 ss3;
hh=hhwk(:,1);
week=hhwk(:,2);

I_init=find(init<=week);

hh_ini=hhwk(I_init,1);  %these are the HHs that purchased
something
                  % after the initialization period.

rem=[];               %list of rows to remove
trem=[];              %list of trows to remove

for i=1:size(hh)
 if ~(any(hh_ini==hh(i,:)))
   rem=[rem;i];
 end;
end;

for i=1:size(thh)
 if ~any(hh_ini==thh(i))
   trem=[trem;i];
 end;
end;

%rem=find(~any(hh_ini==hh));
%trem=find(~any(hh_ini==thh));

[s1 ss1]=size(namvar);
[s2 ss2]=size(nampvar);
[s3 ss3]=size(namtvar);

for m=1:s2
temp=eval(nampvar(m,:));
temp(rem,:)=[];
eval([nampvar(m,:) '=temp;']);
end;
hhwk(rem,:)=[];
volume(rem,:)=[];

for m=1:s3
temp=eval(namtvar(m,:));
temp(trem,:)=[];
eval([namtvar(m,:) '=temp;']);
end;
trip(trem,:)=[];

row_3=size(tyy,1);
numhh_3=size(hhvector(trip(:,1)),1);

fprintf('\t Got rid of %g HHs, %g rows...\n',numhh_2-
numhh_3,row_2-row_3);

clear rem trem week hh thh init;
pack;

fprintf('\n In total, got rid of %g HHs, %g rows...\n',...
   numhh_ini-numhh_3,row_ini-row_3);
fprintf('\t%g Original HHs, %g Clean
HHs\n',numhh_ini,numhh_3);
fprintf('\t%g Original Purchase Opps, %g Clean Purchase
Opps\n',row_ini,row_3);

clear numhh_ini numhh_1 numhh_2 numhh_3 row_ini row_1
row_2 row_3 temp;
clear hh_ini I_init i Idx1 j k m s s1 s11 s2 s3 ss1 ss2 ss3;
eval(['save ' outfile]);
```

```
fprintf('3) Checking purchases after initialization...\n');

thh=trip(:,1);

%gltreat1.m
% last updated on 5/3/97
%GL treatment of multiple purchases:
%1: Multiple purchases in one event are translated into
multiple different
%  purchase occasions;
%2: Multiple purchases are turned into purchase opportunities:
%          1 purchase becomes 2 purchase opp;
%          2 purchases  "   3     ", etc.

infile='c:\wayne\nest\marion2\clean1.mat';
eval(['load ' infile]);
outfile='c:\wayne\nest\marion2\clean.mat';
namgen;
[s1 s11]=size(namvar);
namvar(1,:)=[];
namvar(s1-1,:)=[];
[s1 ss1]=size(namvar);
[s2 ss2]=size(nampvar);
[s3 ss3]=size(namtvar);
unitsvol=volume;

                           % purchases modification

[p q]=size(volume);
i=1;k=0;j=1;m=1;
for i=1:p
  if volume((k+i),1)>1  %multiple purchase
    v=volume((k+i),2)/volume((k+i),1); %vol per purchase
           for j=1:(volume((k+i),1)-1)
                   for m=1:s2
                   temp=eval(nampvar(m,:));
                   temp=[temp(1:(k+i),:); temp((k+i),:);...
                 temp(((k+i+1):(p+k),:)];
                   eval([nampvar(m,:) '=temp;']);
                   end; %end m loop
                   hhwk=[hhwk(1:(k+i),:); hhwk((k+i),:);...
                 hhwk(((k+i+1):(p+k),:)];
              store=[store(1:(k+i),:); store((k+i),:);...
                 store(((k+i+1):(p+k),:)];
              volume((k+i),1)=1;
              volume((k+i),2)=v;
              volume=[volume(1:(k+i),:);volume((k+i),:);...
                 volume(((k+i+1):(p+k),:)];
              k=k+1;
           end; %end j loop
  end; %end if
end;
fprintf('\nI just finished purchases modification...\n\n');

         %purchase opportunity modification

Idx=find(tyy(:,1));
[pp qq]=size(Idx);
[p q]=size(tyy);
i=1;k=0;j=1;kk=1;
for i=1:pp
        for j=1:tyy(Idx(i))
           if j==1 %first purchase
                          for l=1:unitsvol(kk,1)
                             for m=2:s3  %exclude tyy
                      temp=eval(namtvar(m,:));
                      temp=[temp(1:(k+Idx(i)),:);));
```

```
eval(['clear ']);
eval([namtvar(m,:) '=temp;']);
                                        end; %end m loop
                                        trip=[trip(1:(Idx(i)+k),:);
trip((Idx(i)+k),:);...

trip((Idx(i)+k+1):(p+k),:)];
                                  k=k+1;
                          end;%end l loop
                                  kk=kk+1;
               else
                                        for l=1:unitsvol(kk,1)+1
                                          for m=2:s3  %exclude tyy
                                    temp=eval(namtvar(m,:));
                                    temp=[temp(1:(k+Idx(i)),:);
temp((k+Idx(i)),:);...
                              temp((k+Idx(i)+1):(p+k),:)];
                                          eval([namtvar(m,:)
'=temp;']);
                                          end; %end m loop
                                          trip=[trip(1:(Idx(i)+k),:);
trip((Idx(i)+k),:);...

trip((Idx(i)+k+1):(p+k),:)];
                                  k=k+1;
                          end;%end l loop
                                  kk=kk+1;
               end; %end if
            end;%end j loop
end; %end i loop
fprintf('\nI just finished purchases opptunities
modification...\n\n');



                          % special treatment for tyy

i=1;k=0;j=1;kk=1;
tyy_back=tyy;
for i=1:pp
        for j=1:tyy_back(Idx(i))
                          if j==1
                          temp=eval(namtvar(1,:));
                          temp=[temp(1:(Idx(i)+k),:);
ones((unitsvol(kk,1)-1),1);0;...
                                        temp((Idx(i)+k+1):(p+k),:)];
                          eval([namtvar(1,:) '=temp;']);
                          k=k+unitsvol(kk,1);
                          kk=kk+1;%finish one same product
multiple units purchase conversion
                          else
                          temp=eval(namtvar(1,:));
                          temp=[temp(1:(Idx(i)+k),:);
ones(unitsvol(kk,1),1);0;...
                                        temp((Idx(i)+k+1):(p+k),:)];
                          eval([namtvar(1,:) '=temp;']);
                          k=k+unitsvol(kk,1)+1;
                          kk=kk+1;
                          end; %end if
        end;%end j loop
end; %end i loop

Idx1=find(tyy>1);%Replace greater than 1 numbers with 1
tyy(Idx1)=ones(size(Idx1));

volume=volume(:,2);
basket=[basket,zeros(size(basket))];
temp((k+Idx(i)),:);...
                                  temp((k+Idx(i)+1):(p+k),:)];

tyy=[tyy,1-tyy];

clear ans p q pp qq tyy_back i j k kk l unitsvol Idx Idx1 v...
tyy_back m s1 s2 s3 ss1 ss2 ss3 temp s11;
eval(['save ' outfile]);
eval(['clear ']);


% nest2.m
% Upper level driver for multistage logit model
%
% updated on 5/3/97

infile='c:\wayne\nest\marion2\clean.mat';
eval(['load ' infile]);  %path of nested.mat

outfile='c:\wayne\nest\marion2\calib.mat';

hh=hhwk(:,1);
week=hhwk(:,2);  % Purchase Household and Week identifiers

thh=trip(:,1);  % Trip Household identifier
tweek=trip(:,3); % Trip Week identifier


% Calculate conditional probabilities given a category
purchase
%[beta, gamma, SEbeta,L, L0, loyff,probba,tsum]=
condprol(yy,cs,tcs, ...
%
price,tprice,display,tdisplay,feature,tfeature,hh,thh,week);

conprol;

% Calculate category attractiveness for buy now alternative
[vpa]= catatt1(tsum);

% Build category decision varibles
%[catloy, inventory, catprice,first_buy,ini_inv,crate,catloyff]=
catvar1(yy,tyy,thh,tweek,tprice,volume,week,hh);
catvar1;

basket=basket/mean(basket(:,1));

% Calculate marginal probability of buying from the category
%[delta,SEdelta,Ld,Ld0,probma]=margprol(tyy,tcs,vpa,catloy,
inventory,catprice,basket,first_buy,thh,tweek);
margprol;

% Print out a report to a file
repnes1;

% Plotting stage
plotnes2;

eval(['save ' outfile ' beta delta gamma loyff ini_inv crate
catloyff;']);


% conprol.m -- Determines conditional probabilities given a
purchase
%   by Wayne Xiao 4/10/97
```

```
tcs=ones(size(basket
%
tcs,price,tprice,display,tdisplay,feature,tfeature,hh,thh,week);


% Set Up: Divide data into 3 periods
% Init = 1/2 data, Calib = 1/4 data, Hold Out = 1/4 data

fprintf('\n1) Determining conditional probabilities... \n');
[p q]=size(yy);
diff=max(week)-min(week);
init=min(week)+0.25*diff;
cal=init+0.5*diff;

Ivector=find(week<init);
yy_ini=yy(Ivector,:);
hh_ini=hh(Ivector,:);
purch_ini=hhvector(hh_ini);

[s1 ss1]=size(namvar);
[s2 ss2]=size(nampvår);
[s3 ss3]=size(namtvar);
Cvector=find(week<=cal);
pvar_cal=[]; %store purchase variable in calibration period

for m=1:s2
temp=eval(nampvar(m,:));
temp=temp(Cvector,:);
pvar_cal=[pvar_cal;'pvar_cal' int2str(10+m)]; %reason using
'10' here is
                    %due to string length limitation
eval([pvar_cal(m,:) '=temp;']); %assign temp to pvar_cal1,and
2 etc.
end;
hh_cal=hh(Cvector,:);
purch_cal=hhvector(hh_cal);


% Initialization period
%  Used to determine each household's initial loyalty row
%  loy0 == 0.5(overall mkt share) + 0.5(household's mkt
share)
% A row is included for each household in the sample period.
% For households not in the init period, loy0=tot_mktshr


tot_mktshr(1,:)=mean(yy_ini);

purch=hhvector(hh);
num_hh=size(purch,1);

hh_mktshr=zeros(num_hh,q);
temp=zeros(num_hh,1);
j=1;k=1;l=1;ptr=1;

for i=1:num_hh

 if ~any(hh_ini==hh(j))    %hh not in init period
   hh_mktshr(i,:)=tot_mktshr;
 elseif purch_ini(k)>1
   hh_mktshr(i,:)=mean(yy_ini(ptr:ptr+purch_ini(k)   ))
   ptr=ptr+purch_ini(k);
   k=k+1;
 else
   hh_mktshr(i,:)=yy_ini(ptr,:);
   ptr=ptr+1;
   k=k+1;
```

```
%function [beta, gamma, SEbeta,L,L0,
loyff,probba,tsum]=condprob(yy,cs, ...

if any(hh_cal==hh(j))
   temp(i)=purch_cal(l);
   l=l+1;
 end;

 j=j+purch(i);      %update to the first purch of next hh

end; % for loop

loy0=0.5*(ones(num_hh,1)*tot_mktshr) + 0.5*hh_mktshr;


% Calibration period
%  Used to determine the beta coefficients and the gamma
constant

% Call run_gl with the merchandising matrices corresponding
to the
% calibration period. Rename overall merchandising matrices,
and
% rename calibration merchandising matrices.

nampv_bk=[];
for m=1:s2
ptemp=eval(nampvar(m,:));
nampv_bk=[nampv_bk;'pvar_bk' int2str(10+m)]; %reason
using '10' here is...
                    %that string length is limited between 10-
99
eval([nampv_bk(m,:) '=ptemp;']); %assign ptemp to
pvar_bk1,and 2 etc.
ptemp=eval([pvar_cal(m,:)]); %assign pvar_cal to nampvar
eval([nampvar(m,:) '=ptemp;']);
end;
purch_bk=purch;

purch=temp;

% Run MNL regression to find maximum likelihood
coefficients

rglcol;

beta=beta(:,1)';

% This step determines the initial loyalty for each HH.
% Use the loyalty row obtained from rungl. If a household is
not
% on the calibration period, then its loyalty is assumed to be
% the same as its loyalty at the end of the initialization period.

for m=1:s2
ptemp=eval(nampv_bk(m,:));
eval([nampvar(m,:) '=ptemp;']); %assign ptemp to nampvar.
eval(['clear ' nampv_bk(m,:)]);
end;
purch=purch_bk;

j=1; l=1; ptr=1;
[p q]=size(yy);

for i=1:size(purch,1)

  if any(hh_cal==hh(j)) %hh in calib pd: use loyalty from
```

```
  end; % if statement
 l=l+1;
    end %of while loop

    loyff(i,:)=loyalty(ptr+purch_cal(l)-1,:);
            %take row corresponding to the HH's last purchase
  else      %hh not in calib pd: use loy0 from init period
    loyff(i,:)=loy0(i,:);
  end;  %if statement

 j=j+purch(i);
 end; % for loop


% Now, calculate each HH's loyalty for each purchase in the
sample.
% Use initial loyalties obtained in loyff. Hopefully, this will
converge
% to the real HH's loyalty.

j=1;k=1;
[T J]=size(yy);
loy=zeros(size(price,1),q);

for i=1:size(purch,1)
  loy(j,:)=loyff(i,:);  %Take the initial loyalty from loyff
  j=j+1;

  while j<k+purch(i)     %Update loy for all of HH's
purchases
    loy(j,:)=gamma*loy(j-1,:) + (1-gamma)*yy(j-1,:);
    j=j+1;
  end;

  k=j;
end;

% Finally, take the loyalty row corresponding to each HH's last
purchase as
% the convergent loyalty for that particular HH.

ptr=1; loyff=zeros(size(purch,1),q);

for i=1:size(purch,1)
  ptr=ptr+purch(i);
  %Go to the row corresponding to the following HH's first
purchase

  loyff(i,:)=loy(ptr-1,:); %take the previous row (previous HH's
last purch)
e .d; % for loop

clear loy;


% Determine Conditional Probability Matrix

tpurch=hhvector(thh);

tloy=zeros(size(tprice,1),q);
j=1;

for i=1:size(tpurch,1)
 k=tpurch(i);
 ttemp=ones(k,1) * loyff(i,:);
 tloy(j:j+k-1,:)=ttemp;
 j=j+k;
rungl
    while ~(hh_cal(ptr)==hh(j)),
      ptr=ptr+purch_cal(l);

aa=[ 'tloy                    '];
TVAR=[namtvar(2:s3-1,:);aa];


[T J]=size(tprice);
t=T*J;
[K a]=size(TVAR);
fprintf('\nK is %g\n',K);
tcs1=ones(T,J);
f=find(tprice==0.0251);
tcs1(f)=zeros(size(f),1);
tcs1=reshape(tcs1',t,1);

xtt=[];
for k=1:K
  xtt=[xtt;reshape(eval(TVAR(k,:))',t,1)'];
end;
evb=exp(beta(1:s2-1)*xtt + reshape([0
beta(s2:bb)]'*ones(1,T),t,1)');
%here s2-1 is equal # of price,feat,displ and loyalty
evb=reshape(evb .* tcs1', J, T);

tsum=sum(evb); % sum over all UPC in a given trip

tsum= ~tsum + tsum; % get rid of zeros

probba=(evb ./ (ones(J,1) * tsum))'; % TxJ conditional
probability


% rglco1.m, Last revision: 4/10/97

% Multinomial Logit Guadangi and Little with ASC


VAR2= [ ' DloyDgamma                    ';
      ' loy                  '];
[s1 ss1]=size(namvar);
[s2 ss2]=size(nampvar);
[s3 ss3]=size(namtvar);
VAR=[nampvar;namvar(2:s1,:)];

[K S] = size(VAR);   % K = No. of variables
[Q,r] = size(VAR2);
[T J] = size(eval(VAR(1,:)));
t = T * J;
xt = [];
for k = 3 : K - J + 1
 xt = [ xt; reshape(eval(VAR(k,:))', t, 1)' ];
 eval(['clear ',VAR(k,:)])
end;

VAR = [ VAR(1:2,:); VAR2; VAR(3:K,:) ];

tau=loy0; %total no.of households x J
[q,r] = size(xt);

gamma = [.71];
clear K s r k t

[gamma1, loy, beta, ASC, SEcoeff,xt2,xt,theta,L,L0,s4] = ...
 lgglco1(eval('yy'), eval('cs'), xt, VAR, ones(1,s2-2),
tau,gamma,purch,Q,s1,s2);
```

```
end;
loyalty=xt2;
clear VAR Q xt xt2 tau NHT;

gamma = gamma1(1,1);
%beta=[beta(:,1);theta(:,2);ASC(:,1)]';
beta=[beta;loy;ASC];
SEbeta=[SEcoeff(3:s4-s1+1) SEcoeff(2) SEcoeff(s4-s1+2:s4)];
bb=size(beta);
```

function

```
[gamma1,loy,beta,ASC,SEtheta,xt2,xt,theta,L,L0,s4]...
 = lgglcol(yy, cs, markxt,VAR,starting,
tau,gamma,purch,Q,s1,s2)

% Last revision: 4/10/97

converge = .0001;
maxit2 = 20;

epsilon = 10^(-100);

[T J] = size(yy);
t = T * J;

yy = yy';                          % JxT
cs = reshape(cs',t,1);

[K1 a] = size(markxt);              % K1 = No. of Marketing
vars
K = Q + K1;                        % K = Total # vars inc.
Deriv.loyalty

theta = [zeros(1,Q-1) 4 starting zeros(1,J-1)];     % preset
starting values
thetalist = theta;
testthetalist = [];
tolerance = [10^(-2)*ones(1,Q) 10^(-3)*ones(1,K1+J-1)];
%Set low tol. for der.

maxit = 20;

beta = starting;
L = [];

% OUTER LOOP
for it2 = 1: maxit2

  if it2 >= 1
    gamma = gamma + theta(1) ./ theta(2) ;
    theta(1:Q-1) = zeros(1,Q-1);      % Reset theta deriv. to 0
  end;

  yy = reshape(yy,J,T);
  cs = reshape(cs,J,T);

  [xt2,xt] = clcglcol(yy,cs,tau,gamma,purch,K,J,T);

  xt = [xt;markxt];

  cs = reshape(cs,t,1);

  % INNER LOOP
  for it = 1 : maxit

    num = exp( theta(1:K) * xt ...
%1.093 -0.04 0.1480


num=1xJT

    num = reshape(num .* cs',J,T);   % num=0 (JxT) if not in
choice set
    prob = num ./ (ones(J,1) * sum(num));    % JxT

      % Gradient
    num = yy(:) - prob(:);    % JTx1 residual vector
    delL = [ xt * num ; sum(reshape(num,J,T)')'];  % (K+J)x1

      % Hessian
    A = [];
    num = zeros(K.T);
    for j = 1 : J
      tmp = xt(:, j:J:J*T) .* ( ones(K,1) * prob(j,:) );  % KxT
      A = [ A; sum(tmp') ];                    % append 1xK
      num = num + tmp;
    end     % num = KxT

    H11 = - xt * ((prob(:) * ones(1,K)) .* xt' ) + num * num';
% kxk
    H12 = - A' + num * prob';  % KxJ and KxJ
    H22 = - diag(sum(prob')) + prob * prob';  % JxJ

    H = [H11 H12; H12' H22];    % merge block matrices

    idx = [1 : K K+2:K+J ];
    dtheta = -delL(idx)'/ H(idx,idx);

    theta = theta + dtheta;

    if (abs(dtheta ./ theta ) < tolerance)
        break
    end;

    if ~finite(sum(theta))
        break
    end;
  end

  format short
  L = log(prob(:)'+(1-cs')) * yy(:);  % to avoid log(0) for
nonchoice set
  share = mean(yy');
  L0 = T*log(1/J);
  covtheta = -eye(size(H(idx,idx)))/(H(idx,idx));
  SEtheta = sqrt(diag(covtheta))';


  num = sum(prob.* yy);   % 1xT prob of chosen alt
  [cs2] = max(prob);       % 1xT predicted choice

  if abs(theta(1,1:Q-1)) < converge
    break;
  end;
  if ~finite(sum(theta))
    break
  end;

end;

[s s4]=size(theta);
beta = [theta(3:s2)' SEtheta(3:s2)']; %s2 is #price +#disp+#feat
```

```
    + reshape([0 theta(K+1:K+J-1)]'* ones(1,T),t,1)'); %

gamma1 = [gamma SEtheta(1)];
gamma2 = [1 0];
loy = [theta(2) SEtheta(2)];
ASC = [theta(s2+1:s4)' Setheta(s2+1:s4)'];
```

```
function [xt2,xt] = clcglcol(yy,cs,tau,gamma,purch,K,J,T)

% Last revision: 4/10/97 Wayne Xiao

% Estimates Guadagni Little Model Multinomial
% DESCRIPTION OF MODEL
% Model = log(alpha) + BX

% DEFINITIONS
% purch: the matrix of number of purchases for each HH
% alpha_t = alpha at time t   (General notation for all vars)
% alpha_t1 = alpha at time t-1  (General notation for all vars)
% index2A used to develop xt matrix

epsilon = 10^(-100);
JT = J*T;
M  = max(purch);
[HH a] = size(purch);

index1 = 1:HH' ;                    % used to develop xt
index2A = cumsum(purch) -purch;     % used to develop xt
index2A_1 = cumsum(purch) -purch;       % used to develop
xt

xt1 = zeros(J,T);
xt2 = zeros(J,T);

% Step 1a:
% Calculate B,Y,V's

  pit = purch;
  index1A = pit > 0;
  index1 = find(index1A == 1);
  index2A_1 = index2A_1 + index1A;
  index2 = index2A_1(find(index1A == 1));
  [q,r] = size(index2);

  Y_t = zeros(J,HH);

  for k = 1 : q
    Y_t(:,index1(k)) = yy(:,index2(k));
  end;


% Step 1:
% Calculate: loyalty and Dloy_tDgamma for the first purchase
%   at 1st purch Dloy_tDgamma = 0
%   at 1st purch loyalty = tau (market share)

loy_t=tau'; %JxHH
Dloy_tDgamma = zeros(J,HH);

% Step 3: THE BIG STEP
% Calculate probabilities then alpha and DalphaDtau,
DalphaDgam for each purch

prob = [];
for p = 1: M
tau = [];

delta = [];

new alpha and S
   % But First, some bookkeeping

   Y_t1 = Y_t;                      %previous purchase
   loy_t1 = loy_t;                  %previous loyalty
   Dloy_t1Dgamma = Dloy_tDgamma;          %previous
Dloy_tDgamma

% Step 3a: Save derivatives in an xt matrix to return to
log3tau.m
% index1 tells if HH made a purch  on this loop or not
% pit decrements the purch vector by 1 at each purchase
% index2A is the index of the purchases for each purch
% index2 is the index for xt
% and YES this is a bit confusing at first!

  pit = purch + 1 - p;
  index1A = pit > 0;
  index1 = find(index1A == 1);
  index2A = index2A + index1A;
  index2 = index2A(find(index1A == 1));
  [q,r] = size(index2);

  for k = 1 : q
    xt1(:,index2(k)) = Dloy_tDgamma(:,index1(k));
    xt2(:,index2(k)) = loy_t(:,index1(k));
  end;

  if p == M
     break
  else
   % Calculate next cycle alpha_t

   pit = purch - p;
   index1A = pit > 0;
   index1 = find(index1A == 1);
   index2A_1 = index2A_1 + index1A;
   index2 = index2A_1(find(index1A == 1));
   [q,r] = size(index2);

   Y_t = zeros(J,HH);

   for k = 1 : q
    Y_t(:,index1(k)) = yy(:,index2(k));
   end;

   loy_t = gamma .* loy_t1 + Y_t1 .* (1-gamma) ;

   Dloy_tDgamma =  gamma .* Dloy_t1Dgamma + loy_t1 -
Y_t1 ;
 end
end;


xt = [reshape(xt1,1,JT);
   reshape(xt2,1,JT)];
xt2=xt2';
```

```
% hhvector.m -- creates a vector of the number of purchases
by each hh
% 4/12/97
% requires:  vector of household indices:  hh
```

```
% Calculate the derivatives w.r.t gamma and tau and get

function purch=hhvector(hh)

idx=1;
s=size(hh);
purch(idx)=1;

for n=2:s
  if hh(n)==hh(n-1)
    purch(idx,1)=purch(idx,1)+1;
  else
    idx=idx+1;
    purch(idx,1)=1;
  end;
end;
```

```
% catatt1.m  Calculates category attractiveness for buy now
alternative
%   vpa is a #trips x 2 matrix.
% 4/11/97

function[vpa]=catatt1(tsum)

fprintf('\n2) Determining category attractiveness...\n');

vpa1=log(tsum');

vpa=[vpa1 zeros(size(vpa1,1),1)];
```

```
% catvar1.m
% Builds category choice explanatory variables:
% category loyalty, hh inventory, category price
%
% 4/12/97

%function [catloy, inventory, catprice, first_buy, ini_inv,...
%
crate,catloyff]=catvar1(yy,tyy,thh,tweek,tprice,volume,week,h
h)

fprintf('\n3) Determining category choice variables... \n');

% Setup: identify initialization purchases and trips

diff=max(tweek)-min(tweek);
init=min(tweek)+ 1*diff;     % Set initialization period

Itvector=find(tweek<=init);
tyy_ini=tyy(Itvector,:);
thh_ini=thh(Itvector,:);
tprice_ini=tprice(Itvector,:);
tweek_ini=tweek(Itvector,:);
tpurch_ini=hhvector(thh_ini);
tpurch_tot=hhvector(thh);
numhh_tini=size(tpurch_ini,1);
numhh_ttot=size(tpurch,1);

Ivector=find(week<=init);
yy_ini=yy(Ivector,:);       % Purchases corresponding to Init
Pd
hh_ini=hh(Ivector,:);
purch_ini=hhvector(hh_ini);
purch_tot=hhvector(hh);

% outputs: vector of the number of consecutive purchases by
each hh

numhh_tot=size(purch_tot,1);

volume_ini=volume(Ivector,:);

% Test sample for correctness
fprintf('\nSample correctness test:\n\n');

if (numhh_ini~=numhh_tini)
  fprintf('\tWarning: Number of households in initialization
does not match.\n');
  fprintf('\tNumber of households in purchase initialization:
%g\n',numhh_ini);
  fprintf('\tNumber of households in trip initialization:
%g\n',numhh_tini);
end;

if sum(sum(yy))~=sum(tyy(:,1))
  fprintf('\tWarning: Household purchases do not match!\n');
  fprintf('\t%g purchases in trip data, %g purchases in purchase
data.\n',...
     sum(tyy(:,1)), sum(sum(yy)));
end;

if sum(sum(yy_ini))~=sum(tyy_ini(:,1))
  fprintf('\tWarning: Household initialization purchases do not
match!\n');
  fprintf('\t%g purchases in trip data, %g purchases in purchase
data.\n',...
     sum(tyy_ini(:,1)), sum(sum(yy_ini)));
end;

fprintf('\nTesting household purchases during initialization
period...\n');

if (numhh_ini~=numhh_tot)
  fprintf('\tWarning:  %g Households are missing from the
initialization period.\n',...
   numhh_tot-numhh_ini);
end;

for i=1:numhh_ini
  if (purch_ini(i)<2)
    fprintf('\tWarning: Only %g purchases by household %g
during init.\n',purch_ini(i),i);
  end;
end;

fprintf('\nTesting household trips during initialization
period...\n');

for i=1:numhh_tini
  if (tpurch_ini(i)<2)
    fprintf('\tWarning: %g Trips by household
%g\n',numhh_tini,i);
  end;
end;


% Category loyalty

ptr=1;
catloy=zeros(size(tyy,1),2);
catloyff=zeros(numhh_tini,2);
```

```
numhh_ini=size(purch_ini,1);


temp=mean(tyy_ini(ptr:ptr+tpurch_ini(i)-1,:));
 else
   temp=tyy_ini(ptr);
 end;
 catloy(ptr:ptr+tpurch_tot(i)-1,:)=ones(tpurch_tot(i),1) * temp;
 catloyff(i,:)=temp;
 ptr=ptr+tpurch_ini(i);
end;

%Household Inventory

quant=zeros(size(tyy_ini,1),1);
def_quant=zeros(size(tyy,1),1);

Q=find(tyy_ini(:,1)); % Index of buying trips
def_Q=find(tyy(:,1));

quant(Q)=volume_ini;
def_quant(def_Q)=volume;

% A) Consumption Rate
crate=zeros(numhh_tini,1);

ptr=1; def_ptr=1;
for i=1:numhh_tini
 n=1+tweek_ini(ptr+tpurch_ini(i)-1)-tweek_ini(ptr);
   %number of weeks in initialization period for the household

 tot=sum(quant(ptr:ptr+tpurch_ini(i)-1));

 if (tot==0)
   fprintf('\tWarning: Household %g did not make a purchase
during init\n',i);
   tot=sum(def_quant(def_ptr:def_ptr+tpurch_tot(i)-1));
   n=tweek(def_ptr+tpurch_tot(i)-1)-tweek(def_ptr)+1;
 end;

 crate(i,:)=1.5*tot/n;   %150% crate to compensate for
underestimation

 ptr=ptr+tpurch_ini(i);
 def_ptr=def_ptr+tpurch_tot(i);
end;


% B) Initialize Inventory

inv=zeros(size(yy_ini,1),1);
ini_inv=zeros(numhh_tini,1);
tmp=0;

j=1;k=1;
for i=1:numhh_tini
 inv(j)=ini_inv(i);   .
 j=j+1;

 while j<k+tpurch_ini(i)
  temp=inv(j-1)-(tweek_ini(j)-tweek_ini(j-1)) + quant(j-
1)/crate(i);

  if temp>-100
    inv(j)=temp;
  else
    inv(j)=-100;

for i=1:numhh_tini
  if(tpurch_ini(i)>1)



  j=j+1;
 end;

 ini_inv(i,:)=-min(inv(k:k+tpurch_ini(i)-1));


 k=j;
end;


% C) Run complete inventory

inventory=zeros(size(tyy,1),1);
quant=zeros(size(tyy,1),1);
quant(def_Q)=volume;

j=1;k=1;
for i=1:numhh_tini
 inventory(j,:)=ini_inv(i);
 j=j+1;

 while j<k+tpurch_tot(i)
  temp=inventory(j-1)-(tweek(j)-tweek(j-1)) + quant(j-
1)/crate(i);

  if temp<0
    temp=0;
  elseif temp>100
    temp=100;
  end;

  inventory(j)=temp;
  j=j+1;
 end;

 k=j;
end;

%inventory(:,1)=inventory(:,1)/40
inventory=[inventory zeros(size(inventory,1),1)];


% Category price

catprice=zeros(size(tprice,1),1);
for i=1:size(tprice)
  f=find(tprice(i,:)~=0.0251);
  catprice(i)=mean(tprice(i,f));
end;

catprice=[catprice zeros(size(catprice,1),1)];

% First purchase opportunity dummy

first_buy=zeros(size(tyy,1),1);

flag=0;

for i=1:size(tyy,1)    % Loop through all purchase opps

 if (tyy(i,1) & ~flag)  %If there is a purchase and it is
```

```
  end;


 flag=0;
  end;

 end;

 first_buy=[first_buy zeros(size(first_buy,1),1)];


% margprol.m -- Determines the marginal probability of
making a
%  category purchase on a shopping trip.
%  Wayne Xiao, 4/13/97

%function
[delta,SEdelta,Ld,Ld0,probma]=margpro1(tyy,tcs,vpa,catloy,in
ventory,...
%   catprice,basket,first_buy,thh,tweek);

fprintf('\n4) Determining marginal probability coefficients...
\n');

% Set Up:  Define a calibration period

diff=max(tweek)-min(tweek);
init=min(tweek)+0.25*diff;
cal=init+0.5*diff;

Cvector=find(tweek<=cal);
tyy_cal=tyy(Cvector,:);
tcs_cal=tcs(Cvector,:);
vpa_cal=vpa(Cvector,:);
catloy_cal=catloy(Cvector,:);
inventory_cal=inventory(Cvector,:);
catprice_cal=catprice(Cvector,:);
basket_cal=basket(Cvector,:);
first_buy_cal=first_buy(Cvector,:);

tyy_bk=tyy; tyy=tyy_cal;
tcs_bk=tcs; tcs=tcs_cal;
vpa_bk=vpa; vpa=vpa_cal;
catloy_bk=catloy; catloy=catloy_cal;
inventory_bk=inventory; inventory=inventory_cal;
catprice_bk=catprice; catprice=catprice_cal;
basket_bk=basket; basket=basket_cal;
first_buy_bk=first_buy; first_buy=first_buy_cal;

% Run MNL regresion to find maximum likelihood
coefficients

mnl_ma1;

tyy=tyy_bk;
tcs=tcs_bk;
vpa=vpa_bk;
catloy=catloy_bk;
inventory=inventory_bk;
catprice=catprice_bk;
basket=basket_bk;
first_buy=first_buy_bk;


% Determine marginal probability matrix

TVAR=['vpa      ';
```

```
   first_buy(i)=1;    % the first in that trip
   flag=1;
  elseif (~tyy(i,1) & flag) %This is the last purch opp of trip



'inventory  ';
    'catprice  ';
    'basket    ';
    'first_buy  '];

[T J]=size(catloy);
[K a]=size(TVAR);
t=T*J;
tcs=reshape(tcs',t,1);

xtt=[];
for k=1:K
 xtt=[xtt;reshape(eval(TVAR(k,:))',t,1)'];
end;

eva=exp(delta(1:K)*xtt + reshape(([delta(K+1)
0]'*ones(1,T)),t,1)');

eva=reshape(eva .* tcs', J, T);
summ=sum(eva);

probma=(eva ./ (ones(J,1) * summ));

probma=probma';



% mnl_ma1.m -- Variant of runmnl.m that determines the
coefficients
% delta for category purchase variables.
% Category purchase variables:  Buy now dummy, category
attractiveness
%  category loyalty, household inventory, and category price.
% Wayne Xiao, 4/12/97


Starting = [0 0 0 0 0 0];  % Any better guesses?

VAR = ['tyy              ';
      'tcs              ';
      'vpa              ';
      'catloy           ';
      'inventory        ';
      'catprice         ';
      'basket           ';
      'first_buy        ';
      'buy_now          ';];



[K S] = size(VAR);   % K = No. of variables
  [T J] = size(eval(VAR(1,:)));
t = T * J;
xt = [];
for k = 3 : K-(J-1)
 xt = [ xt; reshape(eval(VAR(k,:))', t, 1)' ];
end;   % for k


[delta,SEdelta,Ld,Ld0]=logitma1(tyy, tcs, xt, VAR, Starting);
```

```
'catloy     ';


% logitmal.m -- Variation of logit3.m to optimize coefficents
% delta of category purchase variables.
% 4/12/97


function [delta,SEdelta,Ld,Ld0] = logitmal(tyy, tcs, xt, VAR, starting)

[T J] = size(tyy);
t = T * J;

tyy = tyy';    % JxT
tcs = reshape(tcs',t,1);

[K a] = size(xt);    % K = No. of attributes


delta = [starting 0];
deltalist = delta;
testdeltalist = [];
tolerance = 10^(-8)*ones(1,K+J-1);
maxit = 20;

for it = 1 : maxit

% Newton-Raphson

  num = exp( delta(1:K) * xt ...
    + reshape([delta(K+1) 0]'* ones(1,T),t,1)'); % num=1xJT

  num = reshape(num .* tcs',J,T);   % num=0 (JxT) if not in choice set

  prob = num ./ (ones(J,1) * sum(num));    % JxT

% Gradient
  num = tyy(:) - prob(:);   % JTx1 residual vector

  delL = [ xt * num ; sum(reshape(num,J,T)')'];   % (K+J)x1

    % Hessian
  A = [];
  num = zeros(K,T);
  for j = 1 : J
   tmp = xt(:, j:J:J*T) .* ( ones(K,1) * prob(j,:) );  % KxT
   A = [ A; sum(tmp') ];   % append 1xK
   num = num + tmp;
  end     % num = KxT
  H11 = - xt * ((prob(:) * ones(1,K)) .* xt' ) + num * num';  % kxk
  H12 = - A' + num * prob';   % KxJ and KxJ
  H22 = - diag(sum(prob')) + prob * prob';   % JxJ
  H = [H11 H12 ; H12' H22];    % merge block matrices

  idx = [1:K+1];   % index to eliminate ASC1 to avoid singularity

   ddelta = -delL(idx)'/ H(idx,idx);
   delta = delta + ddelta;
   testdelta = abs(ddelta)./abs(delta);
   deltalist = [deltalist;delta];
   testdeltalist = [testdeltalist;testdelta];

   if (abs(ddelta)./abs(delta) < tolerance)

% clear VAR K S T J k t xt;


end

end

format short
Ld=log(prob(:)'+(1-tcs'))*tyy(:);
Ld0=T*log(1/J);

covdelta=-inv(H(idx,idx));
SEdelta=sqrt(diag(covdelta))';



%repnes1.m -- Outputs the results of the nested run to a file
% 4/14/97
repfile='c:\wayne\nest\marion2\report.txt';

if exist(repfile)
 eval(['delete ' repfile]);
end;

fprintf('\n5) Reporting results... \n');
[s1 ss1]=size(namvar);
[s2 ss2]=size(nampvar);
f= [ ' loyalty                        '] ;
VAR=[nampvar(3:s2,:);f(1,:);namvar(2:s1,:)]; %create the VAR vector from namvar

TVAR=['category attractiveness';
    'category loyalty       ';
    'household inventory    ';
    'category price         ';
    'basket                 ';
    'first_buy              ';
    'buy-now dummy          '];

fprintf(repfile,'\n\tResults from Nested Run on ');
tit=['Data Set ' num2str(size(yy,1))];
fprintf(repfile,tit);
fprintf(repfile,'\n\n\t\tConditional probability...\n');
fprintf(repfile,'VARIABLE    COEFF    STD.ERROR  T-VALUE\n');

for k=1:size(VAR,1)
  fprintf(repfile,VAR(k,:));
  fprintf(repfile,'    %#8.4f   %#8.4f    %#8.4f\n',...
   beta(k), SEbeta(k), beta(k)./SEbeta(k));
end;

fprintf(repfile,'\n Loglikelihood value = %7.2f\n',L);
fprintf(repfile,' Null Loglikelihood  = %7.2f\n',L0);
fprintf(repfile,' U-square = %7.5f\n',1-(L/L0));
fprintf(repfile,' N = %g purchasing events\n',size(yy,1));

fprintf(repfile,'\n\n\t\tMarginal probability...\n');

fprintf(repfile,'VARIABLE                COEFF        STD.ERROR     T-VALUE\n');

for k=1:size(TVAR,1)
  fprintf(repfile,TVAR(k,:));
  fprintf(repfile,' %#8.4f      %#8.4f       %#8.4f\n',...
   delta(k), SEdelta(k), delta(k)./SEdelta(k));
```

```
    break

fprintf(repfile,' Null Loglikelihood  = %7.2f\n',Ld0);
fprintf(repfile,' U-square = %7.5f\n',1-(Ld/Ld0));
fprintf(repfile,' N = %g purchasing events\n',size(yy,1));
fprintf(repfile,' N = %g purchasing opportunities\n',size(tyy,1));
```

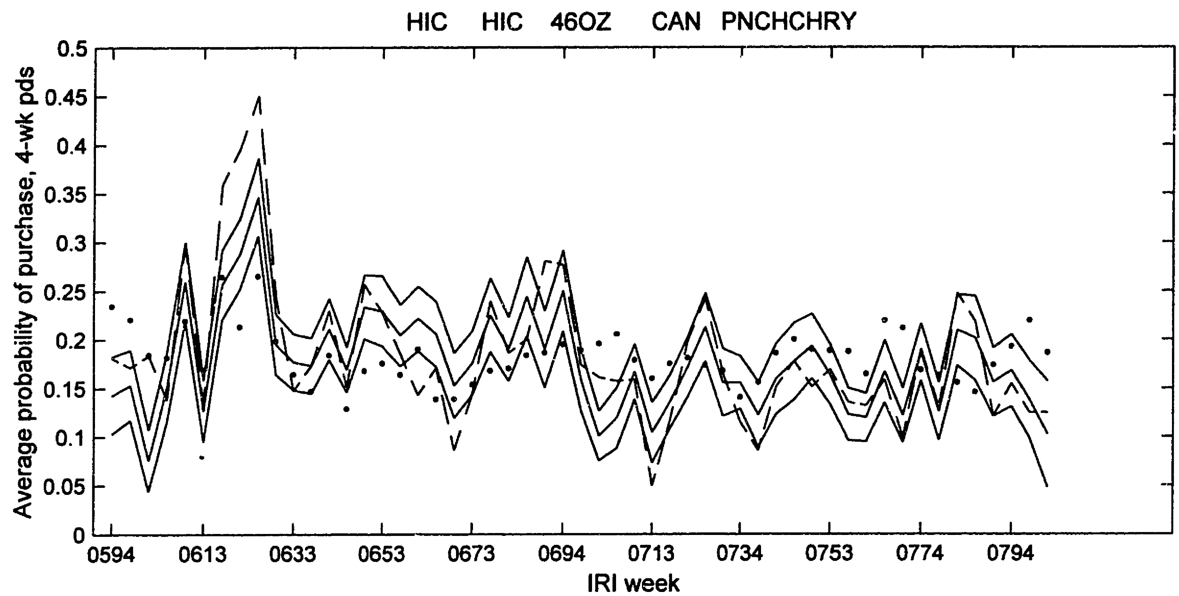
```
% plotnes2.m -- Plots the overall probability of buying a
particular UPC
% last updated 5/3/97

fprintf('\n6) Plotting the graphs...\n');

bsm=namvar;

% Build a full trip driven yy matrix...
tyyf=zeros(size(tyy,1),size(yy,2));

q=find(tyy(:,1));

probtt=probba(q,:);

[b I]=sort(week);
% Sort trips based on week &day

yytt=yy(I,:)';
probtt=probtt(I,:)';
weektt=week(I,:);


% Print the graphs...

wkcountt=hhvector(weektt);
numofwkt=size(wkcountt,1);
int=4;

fig=3;
for ct=1:size(probtt,1)
 t=[];
 xx=[];
 p=1;j=1;

 for i=1:int:numofwkt

  if (numofwkt-i)<int

wkprobtt(j)=mean(probtt(ct,p:p+sum(wkcountt(i:numofwkt))-
1));
        a=(probtt(ct,p:p+sum(wkcountt(i:numofwkt))-1))*...
          (1-(probtt(ct,p:p+sum(wkcountt(i:numofwkt))-
1)))';
        standerr(j)=sqrt(a)/sum(wkcountt(i:numofwkt));
        standhi(j)=wkprobtt(j)+standerr(j)*1.67;
        standlo(j)=wkprobtt(j)-standerr(j)*1.67;
   %if standlo(j)<=0
   %    standlo(j)=0;
   %end;
   wkyytt(j)=mean(yytt(ct,p:p+sum(wkcountt(i:numofwkt))-
1));
   kd=weektt(floor(p+sum(wkcountt(i:numofwkt))/2),1);
   p=p+sum(wkcountt(i:numofwkt));
  else
   wkprobtt(j)=mean(probtt(ct,p:p+sum(wkcountt(i:i+int-1))-
end;

fprintf(repfile,'\n Loglikelihood value = %7.2f\n',Ld);


b=probtt(ct,p:p+sum(wkcountt(i:i+int-1))-1)*...
        (1-(probtt(ct,p:p+sum(wkcountt(i:i+int-1))-1)))';
        standerr(j)=sqrt(b)/sum(wkcountt(i:i+int-1));
   standhi(j)=wkprobtt(j)+standerr(j)*1.67;
   standlo(j)=wkprobtt(j)-standerr(j)*1.67;
   %if standlo(j)<=0
   %    standlo(j)=0;
   %end;
   wkyytt(j)=mean(yytt(ct,p:p+sum(wkcountt(i:i+int-1))-1));
   kd=weektt(floor(p+sum(wkcountt(i:i+int-1))/2),1);
   p=p+sum(wkcountt(i:i+int-1));
  end;

   kd=int2str(kd);
   if size(kd,2)<4
     kd=['0' kd];
   end;

  xx=[xx;kd];
  t=[t j]; j=j+1;
 end; %end i loop

 tx=floor(j/10);

 xx=xx(1:tx:size(xx,1),:);
 xxt=t(1,1:tx:size(t,2));

% Load totstors.dat to compare store sales data to panel data
infile='c:\wayne\nest\marion2\totstors.dat';
eval(['!load ' infile]);
[b,bb]=size(totstors);

% clean store data so that they are consistent with panel data
rem=[]
for i=1:b
 if ~any(find(weektt==totstors(i,1)))
   rem=[rem;i];
 end;
end;
totstors(rem,:)=[]; % clean data
[b,bb]=size(totstors);
store=totstors';
weeksum=(sum(store(2:bb,:)))';

m=1;
pi=b/int;
flr=floor(b/int);
if pi==flr
  for j=1:b/int
        storeshr(j)=sum(totstors(m:m+int-
1,ct+1))/sum(weeksum(m:m+int-1,1));
    m=m+int;
  end;
else
  for j=1:floor(b/int)
        storeshr(j)=sum(totstors(m:m+int-
1,ct+1))/sum(weeksum(m:m+int-1,1));
    m=m+int;
  end;

storeshr(j+1)=sum(totstors(flr*int+1:b,ct+1))/sum(weeksum(flr
*int+1:b,1));
```

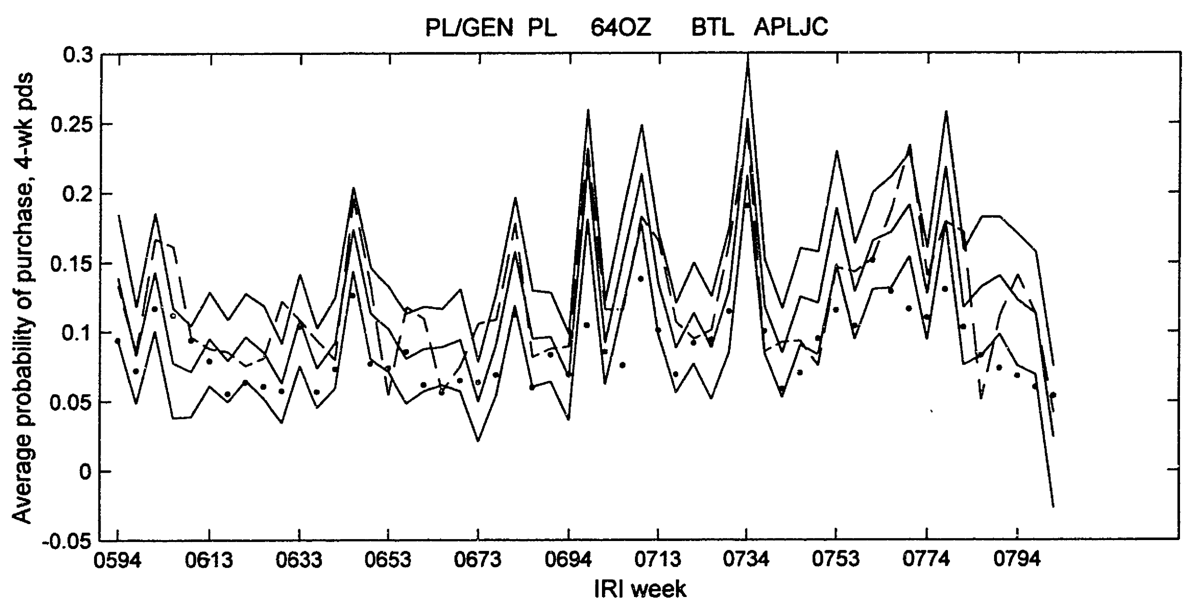
```
1));


if fig<3
   subplot('position',[0.15 0.08 0.70 0.42]);
   fig=fig+1;
 else
   figure;
   fig=1;
   s=[num2str(size(yy,1)) ' purchases'];
   text('String', '"-" is predicted share and 90%
CI','fontsize',[12],'position',...
        [-0.05 -0.15 0],'units','Normalized');
   text('String', '"--" is actual panel
share','fontsize',[12],'position',...
        [0.45 -0.15 0],'units','Normalized');
   text('String', '".." is actual store
share','fontsize',[12],'position',...
        [-.05 -0.20 0],'units','Normalized');
   text('String','Results of the Nested Model at the Conditional
Level','fontsize',[12],...
        'position',[.12 1.40
0],'Units','Normalized','fontunderline','on');
   text('String',s,'fontsize',[12],...
        'position',[.35 1.31
0],'Units','Normalized','fontunderline','on');

   set(gca,'Visible','off');
   subplot('position',[0.15 0.62 0.70 0.42]);
   fig=fig+1;
 end;

 plot(t,wkprobtt,'-',t,wkyytt,'--',t,storeshr,'.',t,standhi,'-
',t,standlo,'-');
 tmax=size(t,2);
 %axis([0 800 ])

 s1=['IRI week'];
 s2=['Average probability of purchase, ' ...
   num2str(int) '-wk pds'];
 tt=[bsm(ct,:)];

 set(gca,'XTickLabels',xx,'fontsize',[8],'XTick',xxt);
 xlabel(s1,'fontsize',[9])
 ylabel(s2,'fontsize',[9])
 title(tt,'fontsize',[9])

 dim=axis;

 if fig==3
 % print;
 end;
end; %end ct loop


for i=1:size(q,1)
 tyyf(q(i),:)=yy(i,:);
end;


% Build a full trip driven prob matrix...
prob=probba .* (probma(:,1) * ones(1,size(probba,2)));


% Now, sort the data.  The trip matrix includes a row for each
trip,
% identifying [HH DAY WEEK STORE].
end;



[b I]=sort((trip(:,3)*10)+trip(:,2));
% Sort trips based on week &day

tyyf=tyyf(I,:)';
prob=prob(I,:)';
tweek=tweek(I,:);
sumact=sum(tyyf);
sumpredi=sum(prob');


% Print the graphs...

wkcount=hhvector(tweek);
numofwk=size(wkcount,1);
int=4;

figure;
fig=3;
for ct=1:size(prob,1)
 t=[];
 xx=[];

 p=1;j=1;

 for i=1:int:numofwk

  if (numofwk-i)<int
   wkprob(j)=sum(prob(ct,p:p+sum(wkcount(i:numofwk))-
1));
   wkyy(j)=sum(tyyf(ct,p:p+sum(wkcount(i:numofwk))-1));
   kd=tweek(floor(p+sum(wkcount(i:numofwk))/2),1);
   p=p+sum(wkcount(i:numofwk));
  else
   wkprob(j)=sum(prob(ct,p:p+sum(wkcount(i:i+int-1))-1));
   wkyy(j)=sum(tyyf(ct,p:p+sum(wkcount(i:i+int-1))-1));
   kd=tweek(floor(p+sum(wkcount(i:i+int-1))/2),1);
   p=p+sum(wkcount(i:i+int-1));
  end;

  kd=int2str(kd);
   if size(kd,2)<4
    kd=['0' kd];
   end;

  xx=[xx;kd];
  t=[t j]; j=j+1;
 end;

 tx=floor(j/10);
 xx=xx(1:tx:size(xx,1),:);
 xxt=t(1,1:tx:size(t,2));

if fig<3
  subplot('position',[0.15 0.08 0.70 0.42]);
  fig=fig+1;
 else
  figure;
  s=[num2str(size(tyy,1)) ' purchases opportunities'];
  fig=1;
  text('String', '"-" is predicted sales
```

```
level','fontsize',[12],'position',...
        [-.05 -0.15 ]);
   text('String', '"--" is actual sales
level','fontsize',[12],'position',...
        [.45 -0.15 ]);
text('String','Results of the Overall Nested
Model','fontsize',[12],...
        'position',[.12 1.40 ],'fontunderline','on');
   text('String',s,'fontsize',[12],...
        'position',[.12 1.31
0],'Units','Normalized','fontunderline','on');

   set(gca,'Visible','off');
   subplot('position',[0.15 0.62 0.70 0.42]);
   fig=fig+1;
 end;


 plot(t,wkprob,'-',t,wkyy,'--');

 s1=['IRI week'];
 s2=['Total units purchased, ' ...
   num2str(int) '-wk pds'];
 tt=[bsm(ct,:)];

 set(gca,'XTickLabels',xx,'fontsize',[8],'XTick',xxt);
 xlabel(s1,'fontsize',[9])
 ylabel(s2,'fontsize',[9])
 title(tt,'fontsize',[9])

 dim=axis;

 if fig==3
  %print;
 end;
end;
```

APPENDIX 2. Matlab plots of predicted vs. actual brandsize shares for dataset 5974

Results of the Nested Model at the Conditional Level
5974 purchases

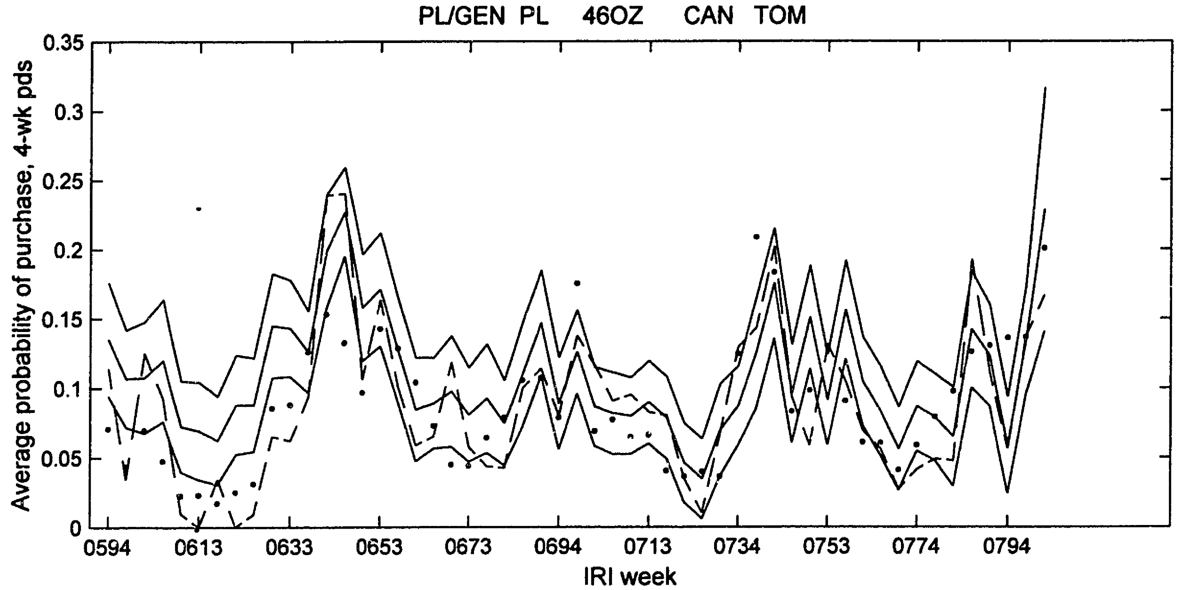


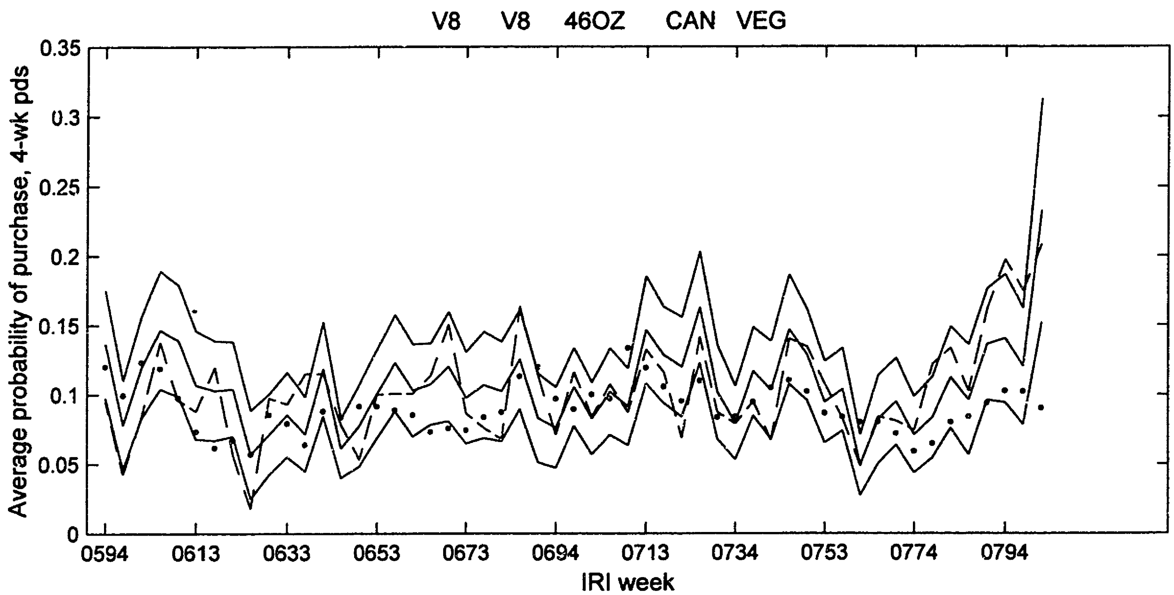


"-" is predicted share and 90% CI "--" is actual panel share
".." is actual store share

APPENDIX 2 (continued).

Results of the Nested Model at the Conditional Level
5974 purchases



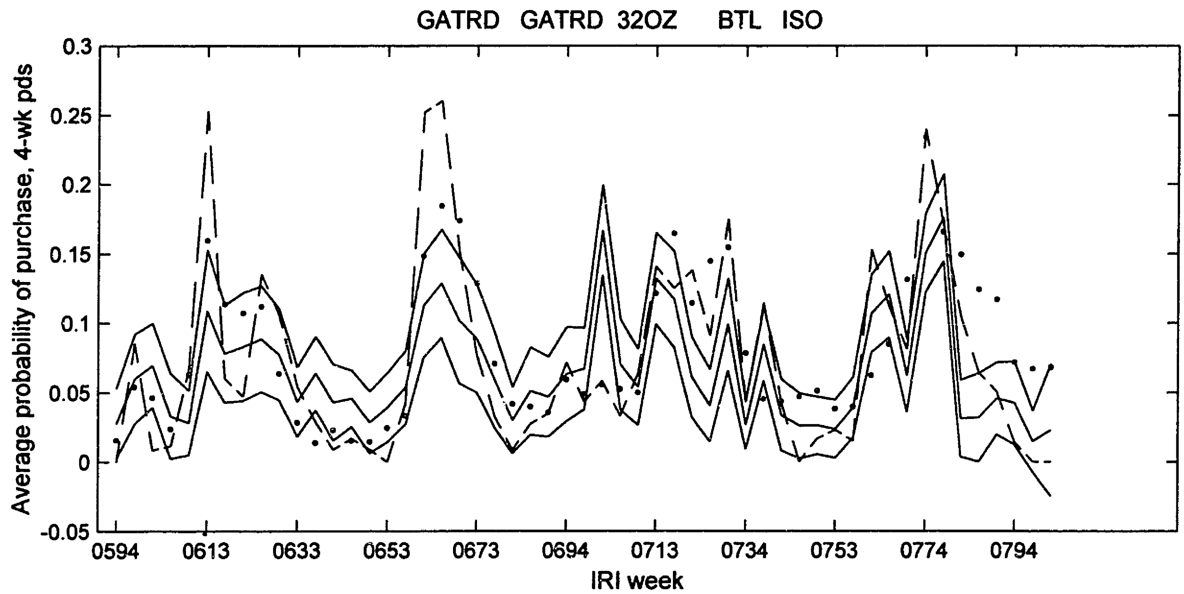


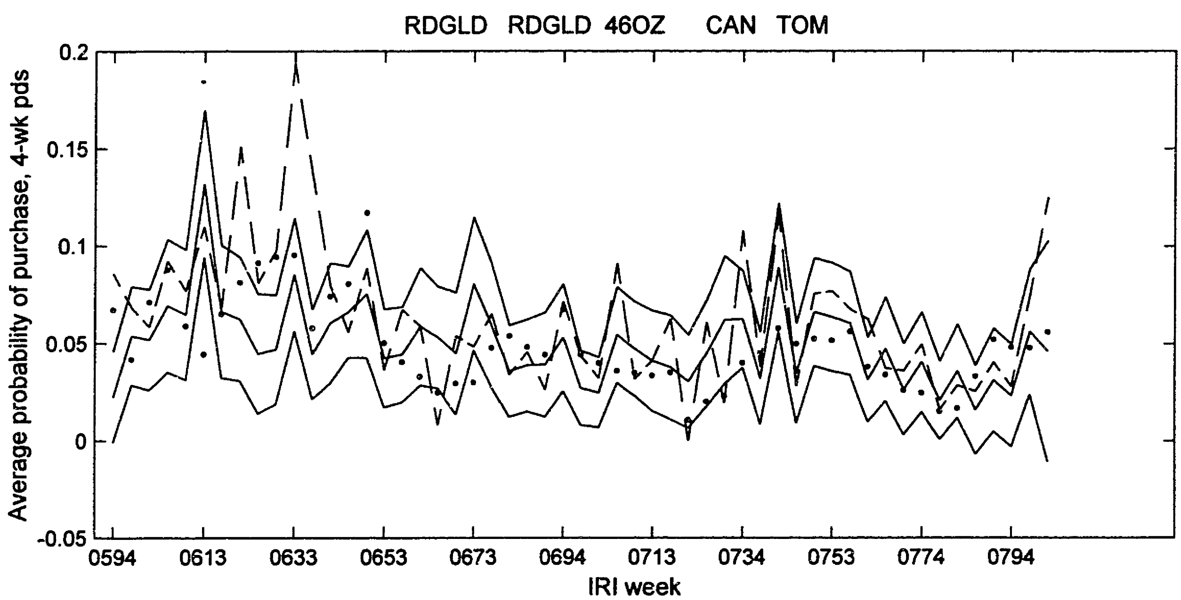
"-" is predicted share and 90% CI "--" is actual panel share

".." is actual store share

APPENDIX 2 (continued).

Results of the Nested Model at the Conditional Level
5974 purchases



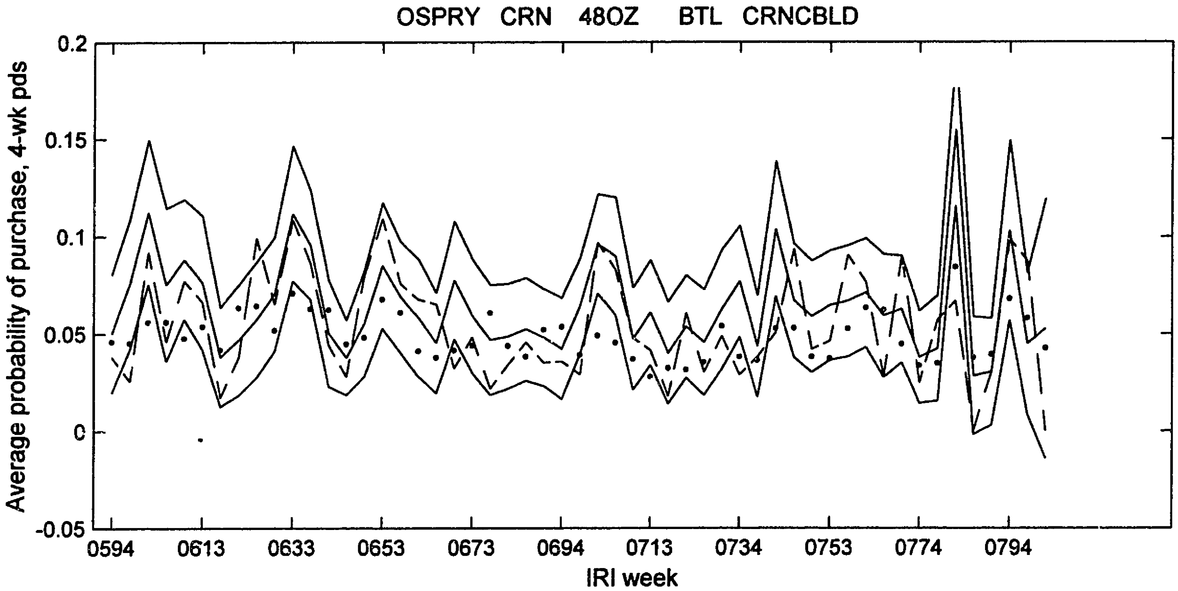


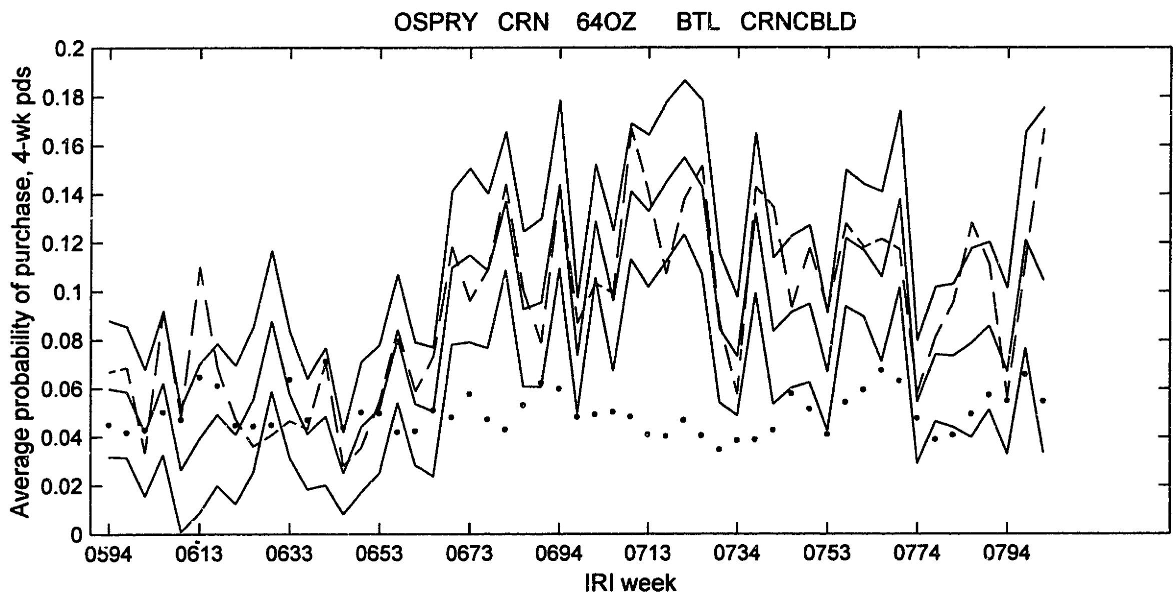
"-" is predicted share and 90% CI "--" is actual panel share

".." is actual store share

APPENDIX 2 (continued).

Results of the Nested Model at the Conditional Level
5974 purchases

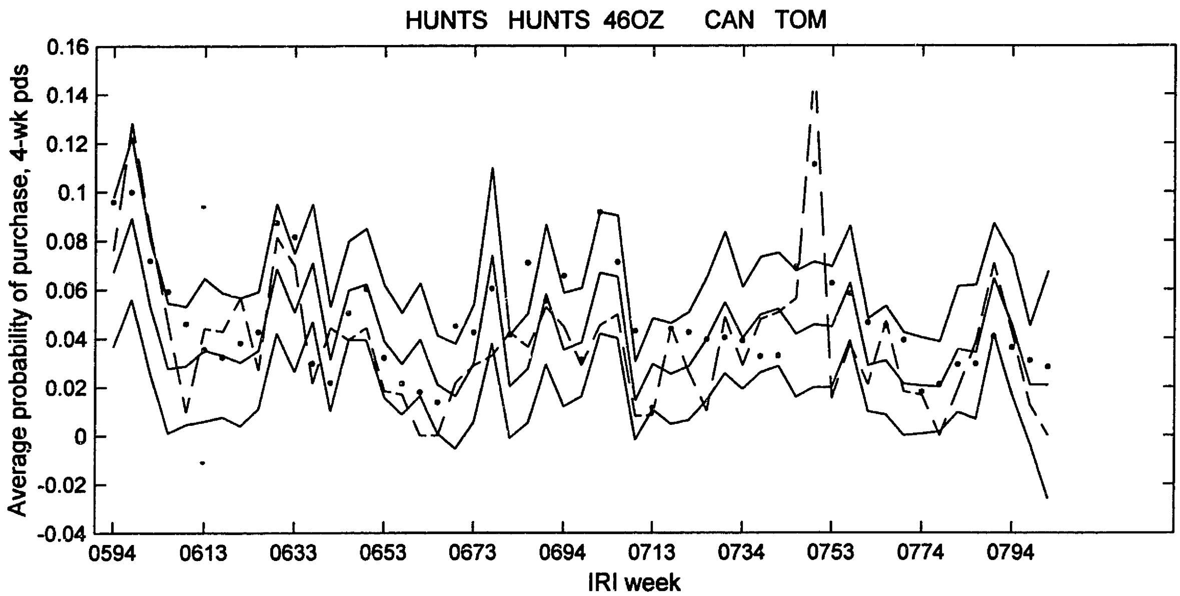


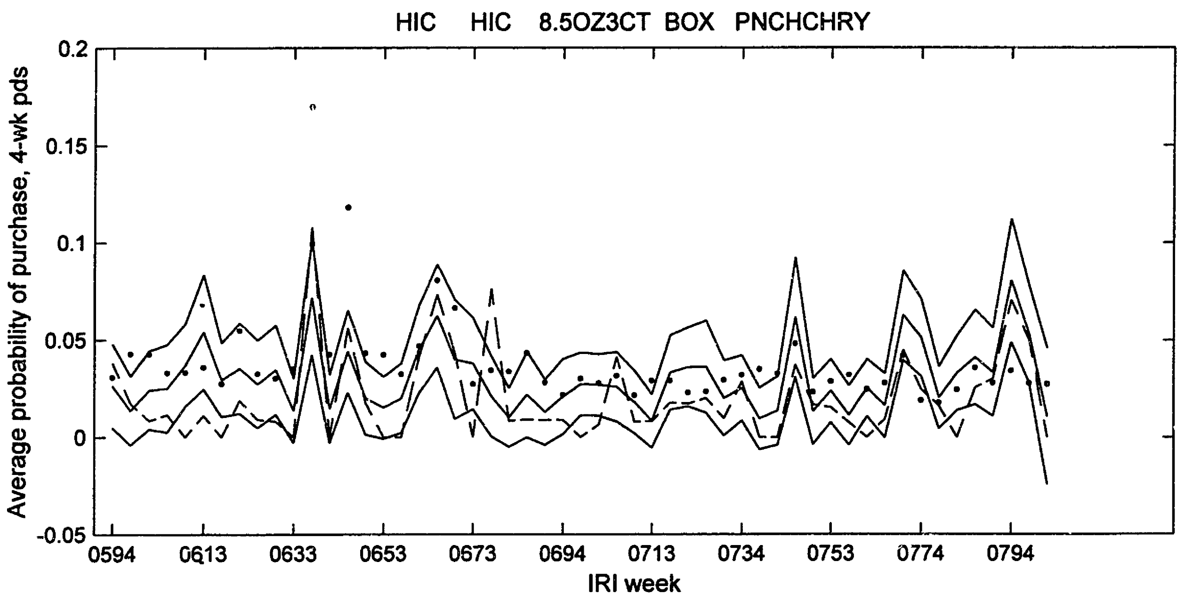


"-" is predicted share and 90% CI "--" is actual panel share
".." is actual store share

APPENDIX 2 (continued).

Results of the Nested Model at the Conditional Level
5974 purchases



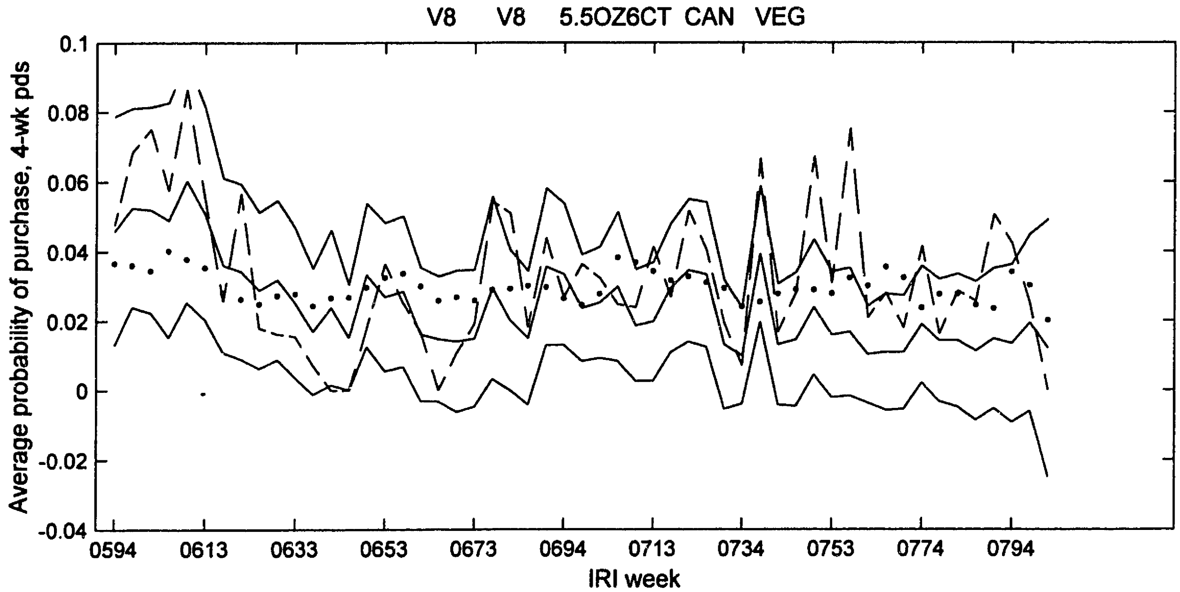


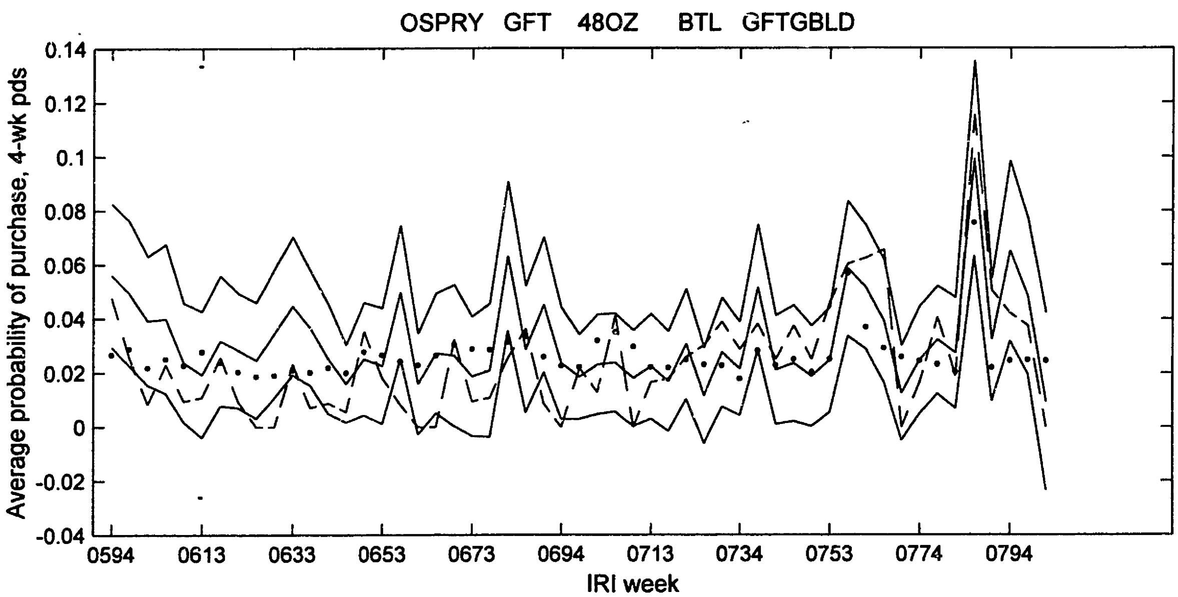
"-" is predicted share and 90% CI "--" is actual panel share

".." is actual store share

APPENDIX 2 (continued).

Results of the Nested Model at the Conditional Level
5974 purchases



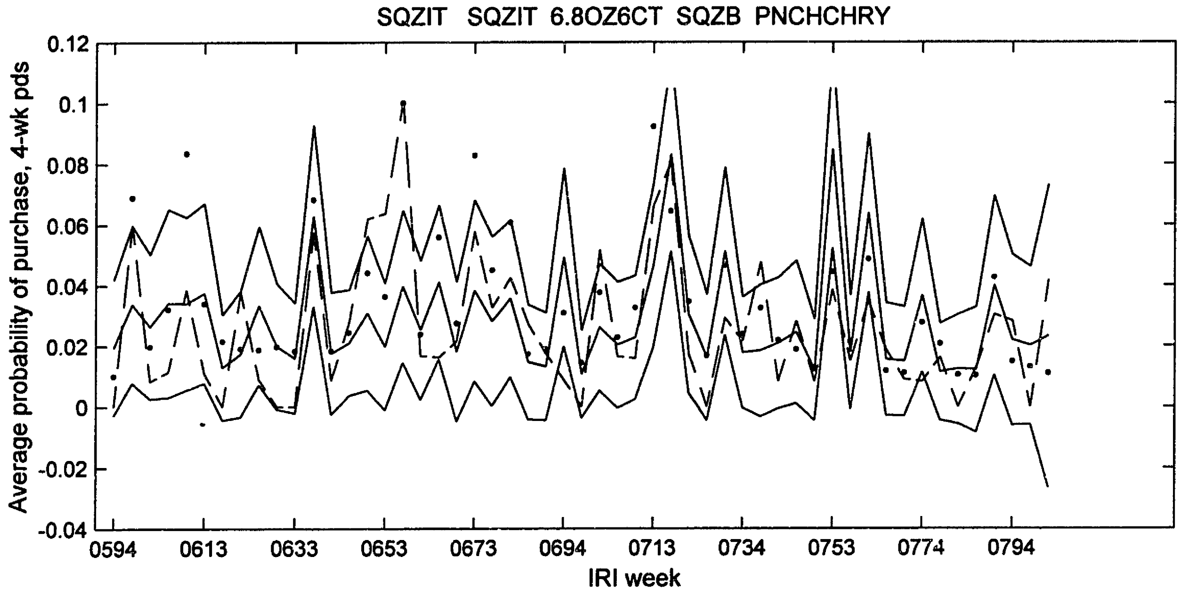


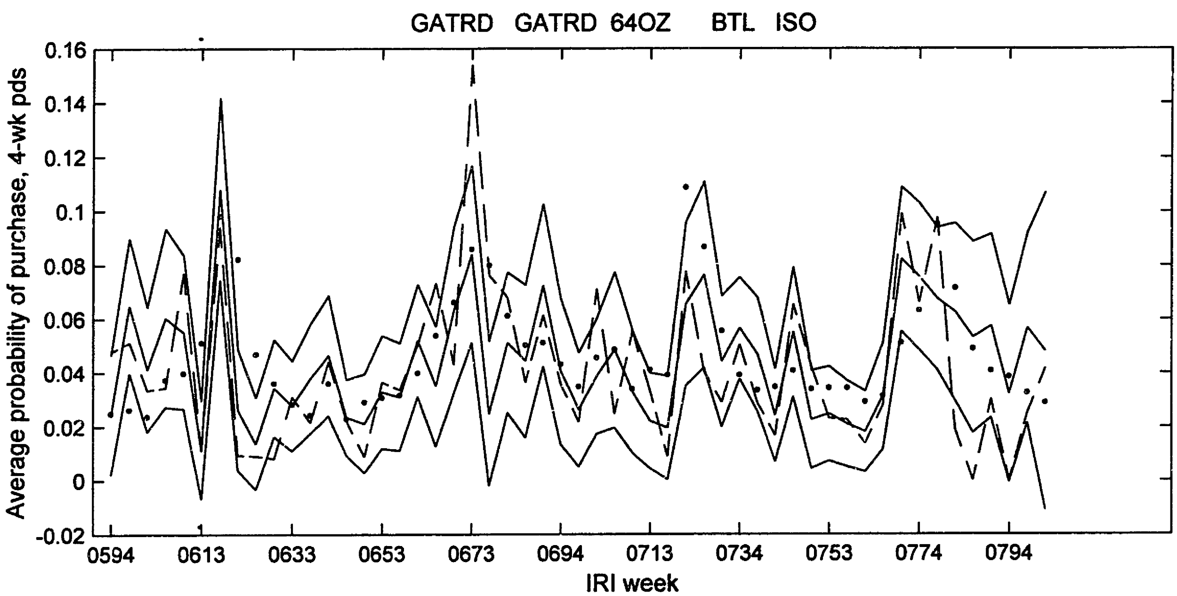
"-" is predicted share and 90% CI "--" is actual panel share

".." is actual store share

APPENDIX 2 (continued).

Results of the Nested Model at the Conditional Level
5974 purchases



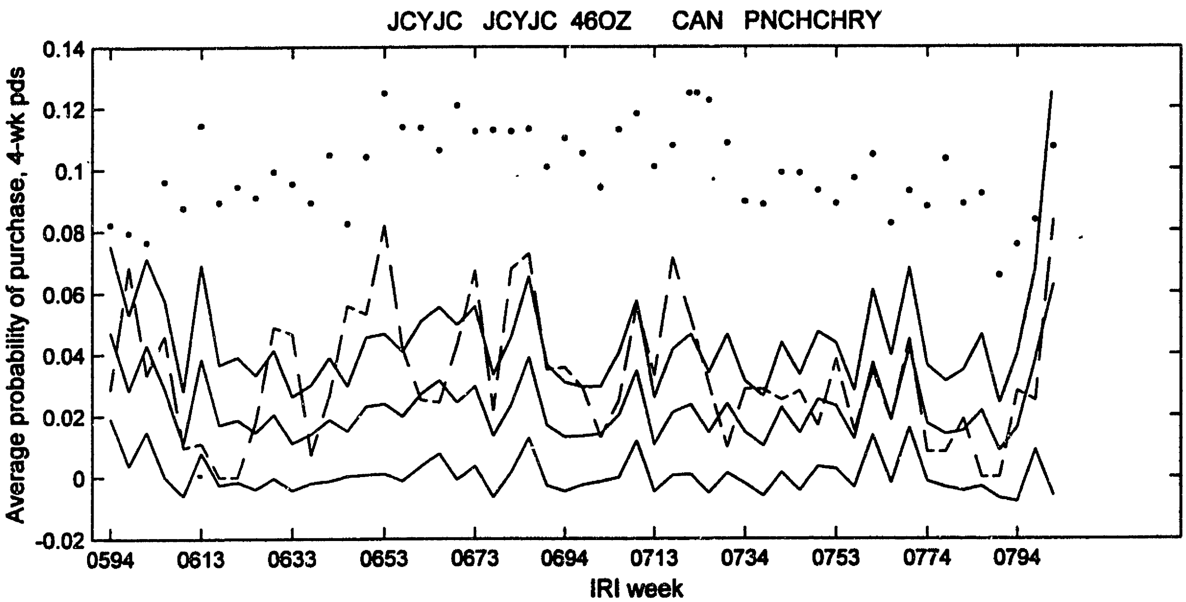


"-" is predicted share and 90% CI "--" is actual panel share

".." is actual store share

APPENDIX 2 (continued).

Results of the Nested Model at the Conditional Level
5974 purchases



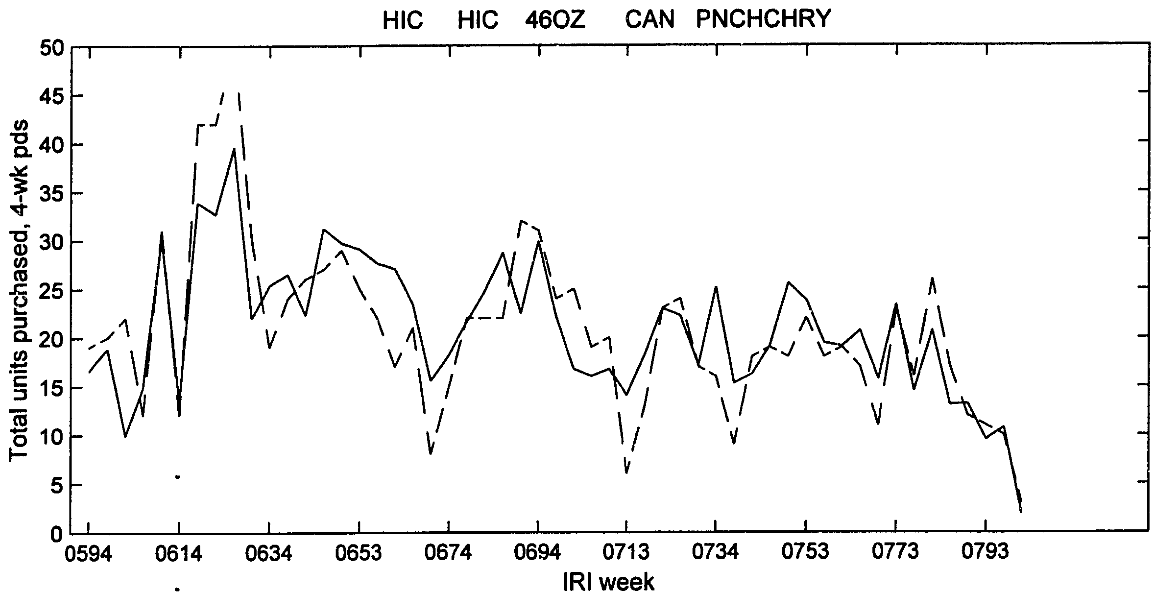
"-" is predicted share and 90% CI

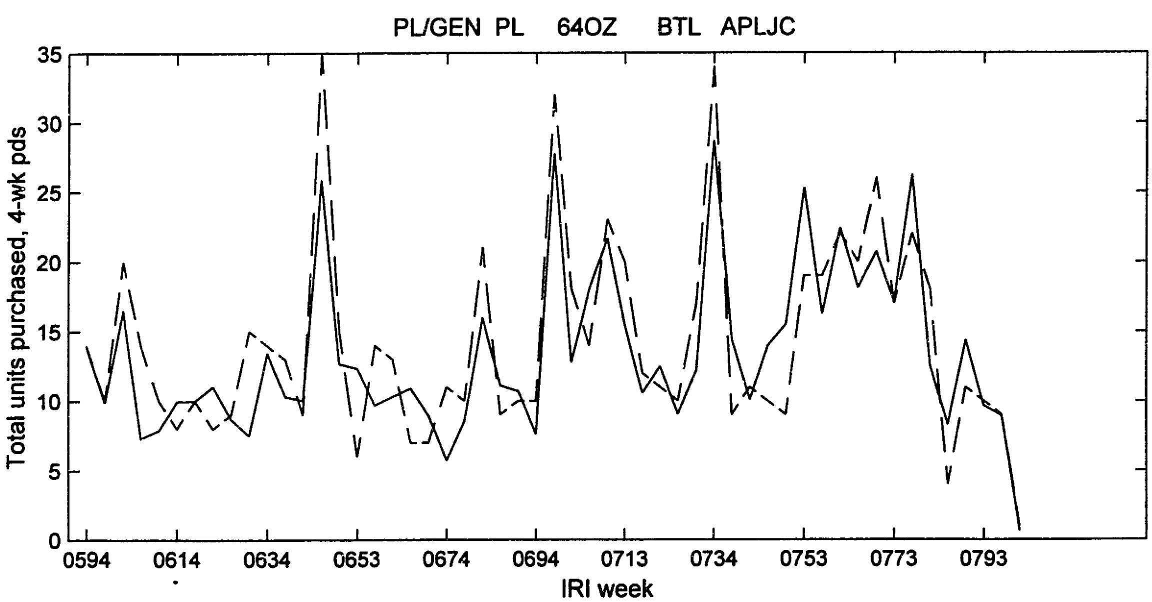
"--" is actual panel share

".." is actual store share

APPENDIX 3. Matlab plots of predicted vs. actual brandsize sales for dataset 5974

Results of the Overall Nested Model
2.83e+004 purchases opportunities

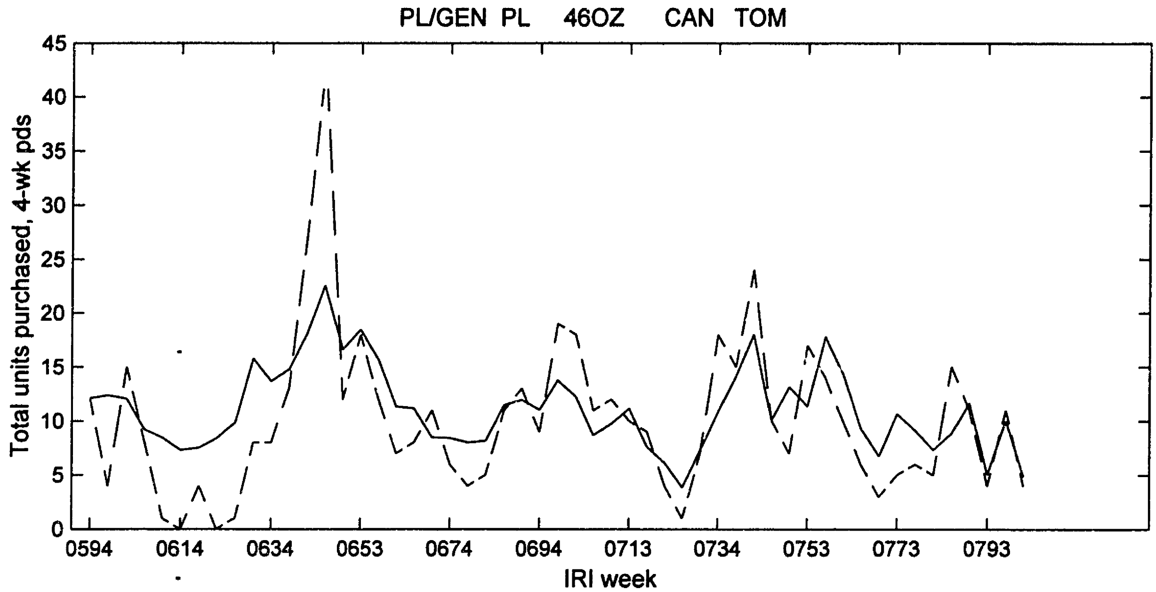


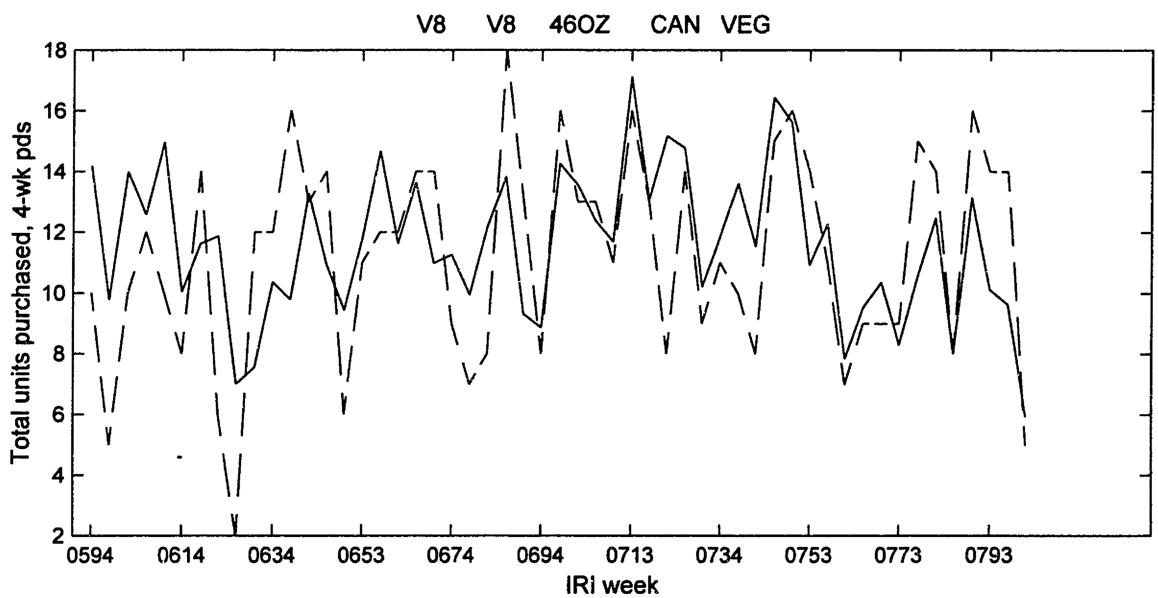


"-" is predicted sales level "--" is actual sales level

APPENDIX 3 (continued).

.Results of the Overall Nested Model
2.83e+004 purchases opportunities

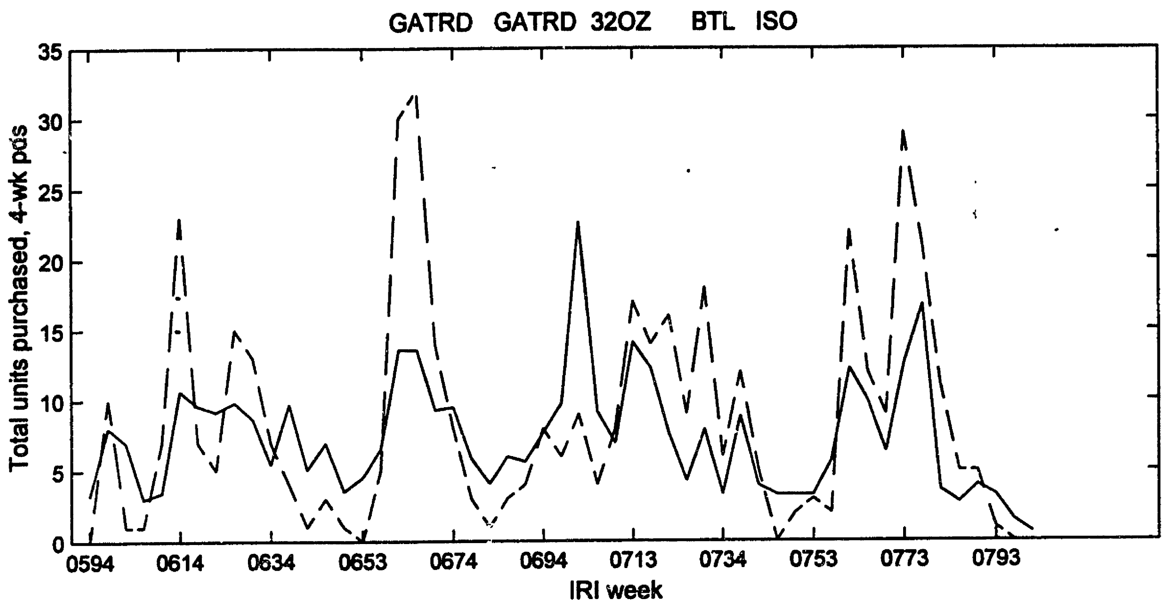


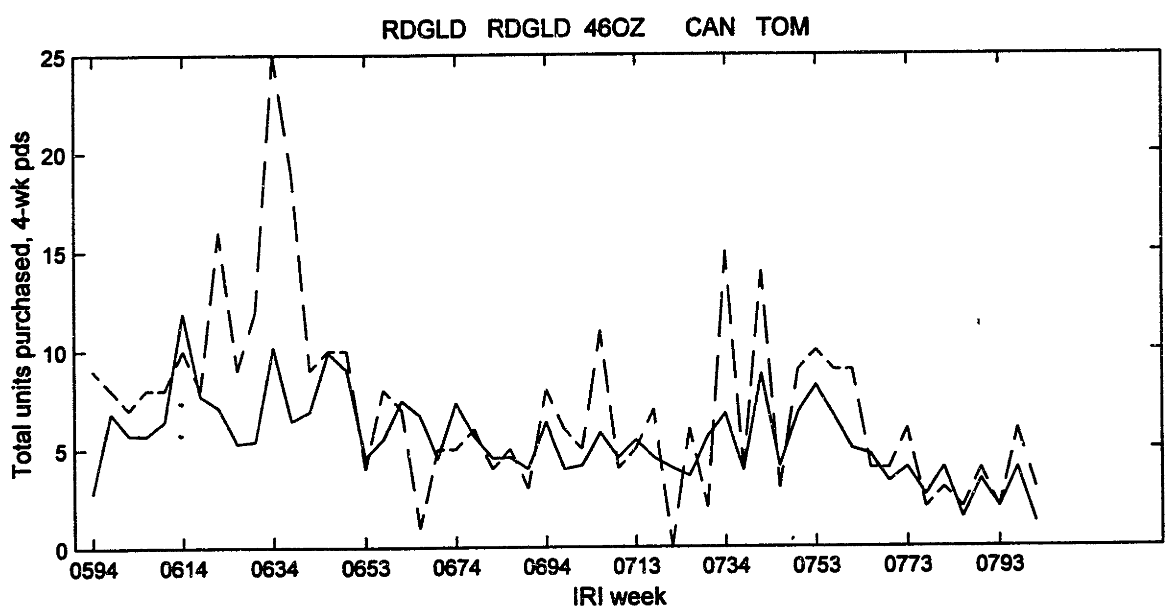


"-" is predicted sales level "--" is actual sales level

APPENDIX 3 (continued).

Results of the Overall Nested Model
2.83e+004 purchases opportunities

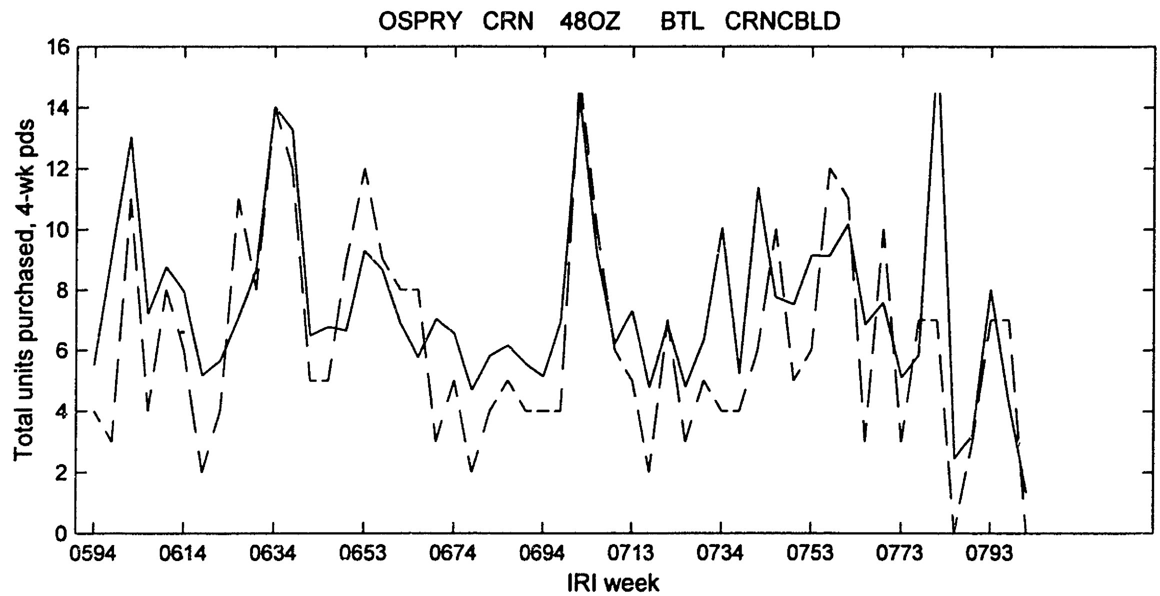


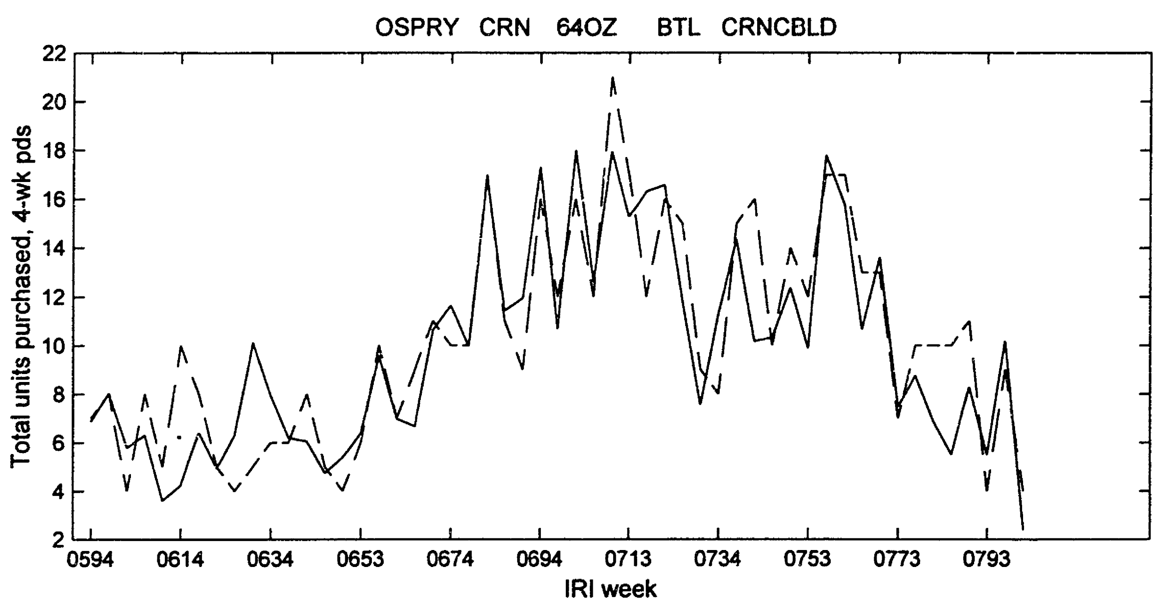


"-" is predicted sales level "--" is actual sales level

APPENDIX 3 (continued).

Results of the Overall Nested Model
2.83e+004 purchases opportunities

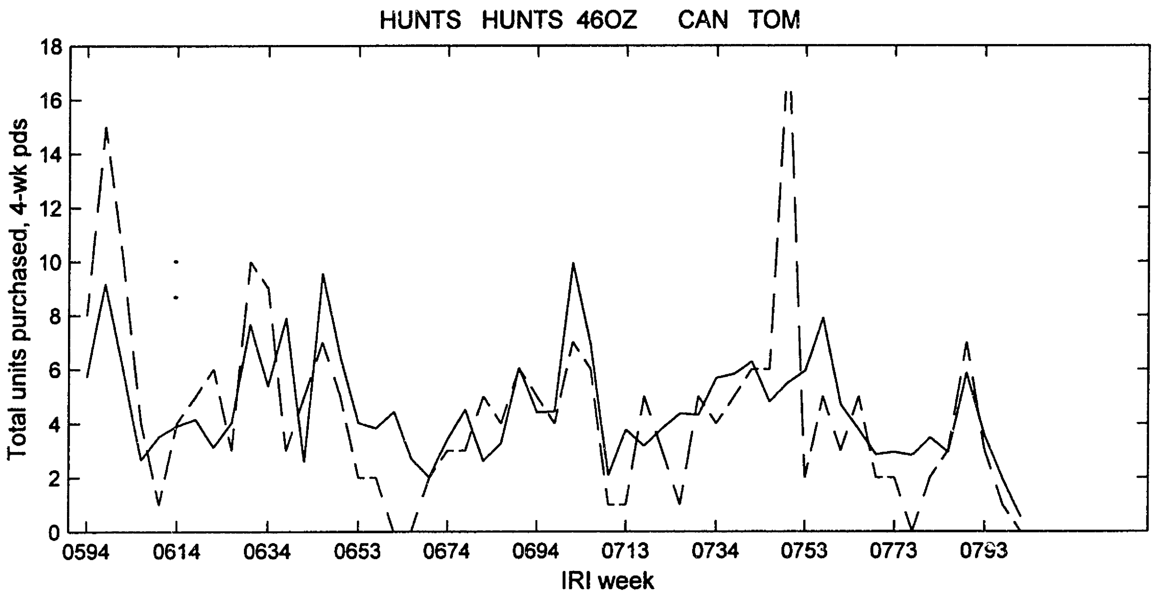


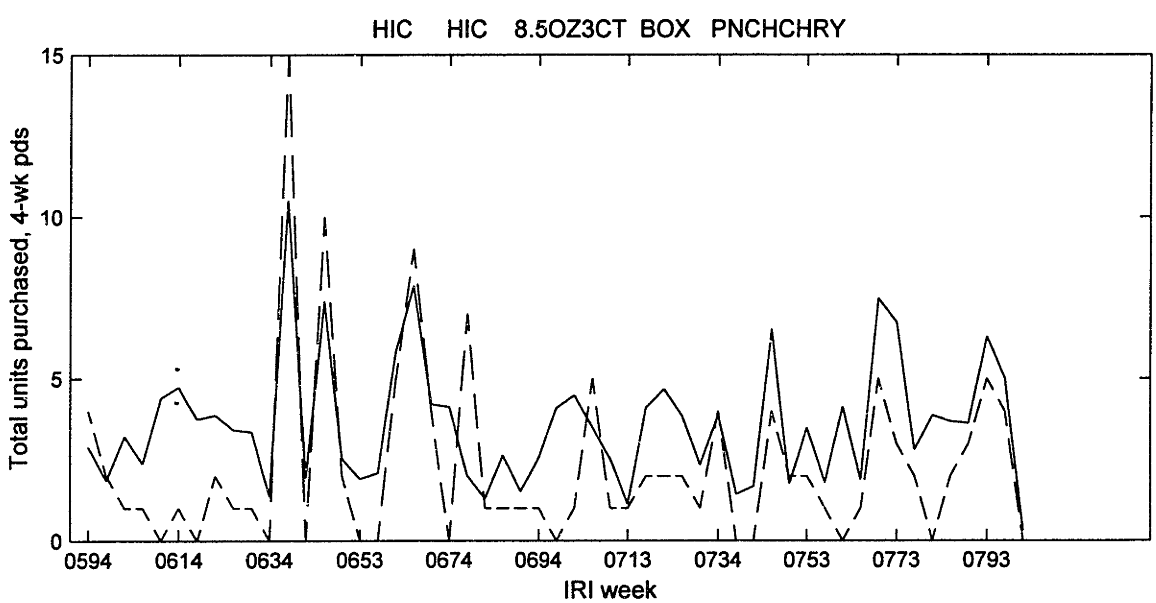


"-" is predicted sales level "--" is actual sales level

APPENDIX 3 (continued).

Results of the Overall Nested Model
2.83e+004 purchases opportunities

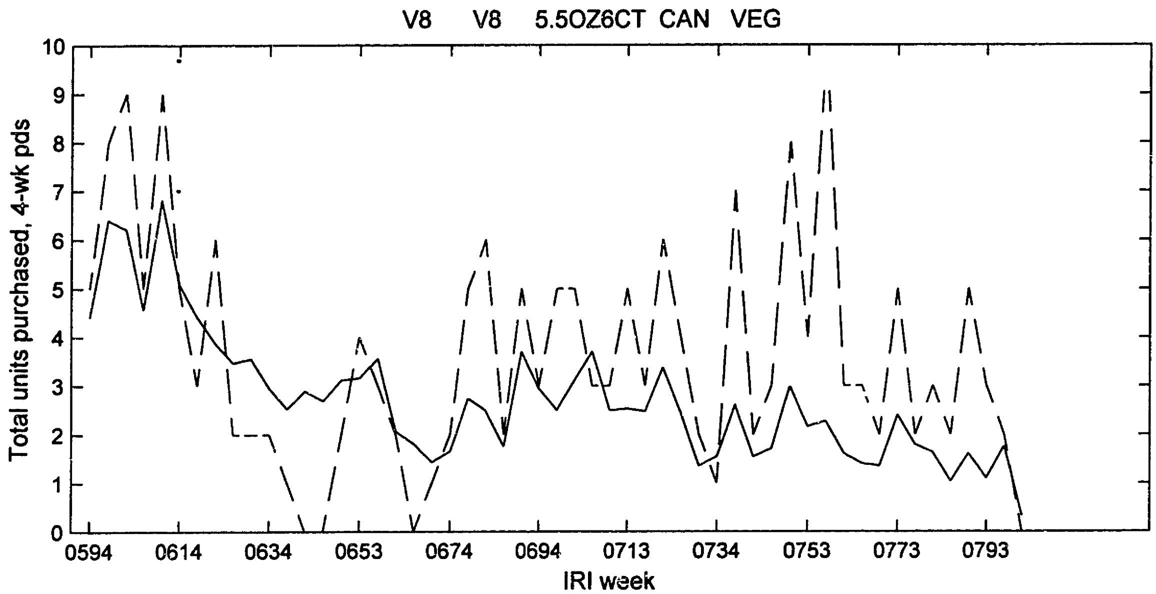


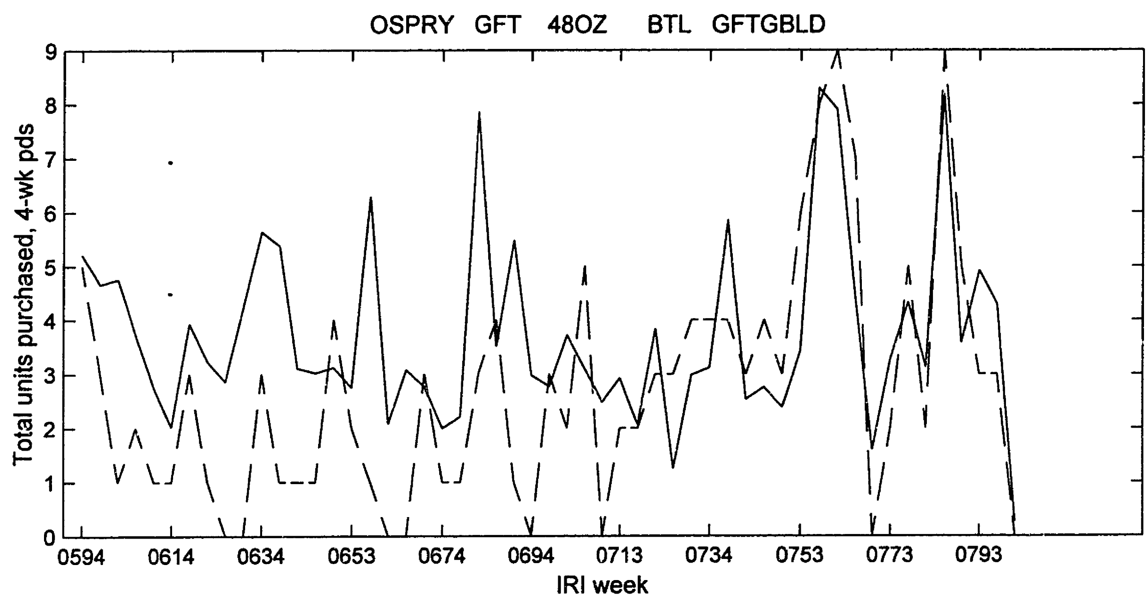


"-" is predicted sales level "--" is actual sales level

APPENDIX 3 (continued).

Results of the Overall Nested Model
2.83e+004 purchases opportunities

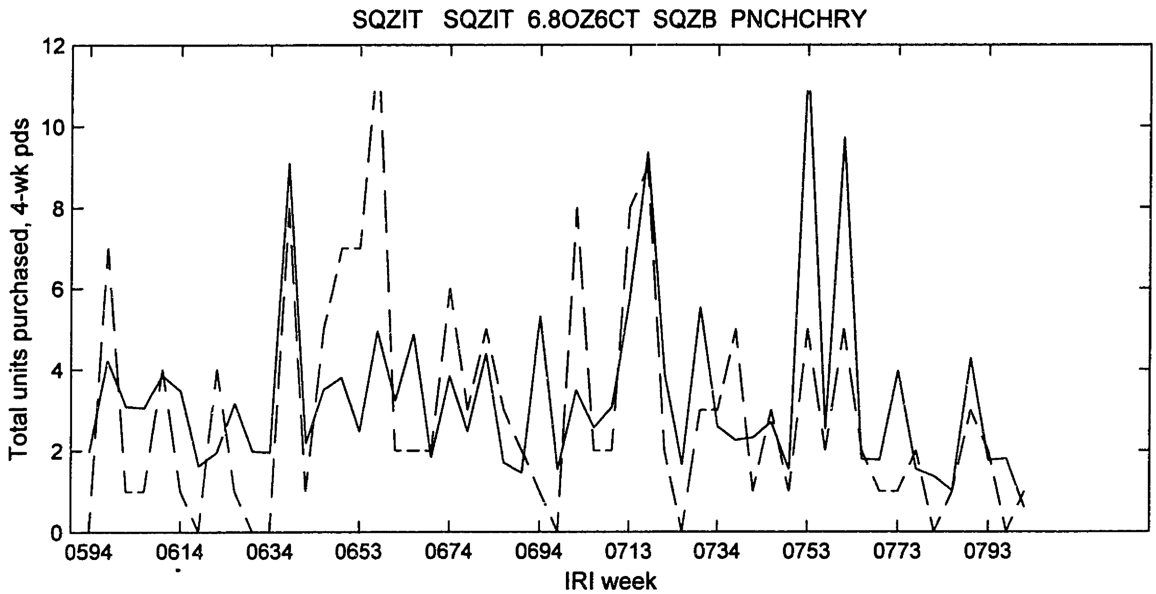


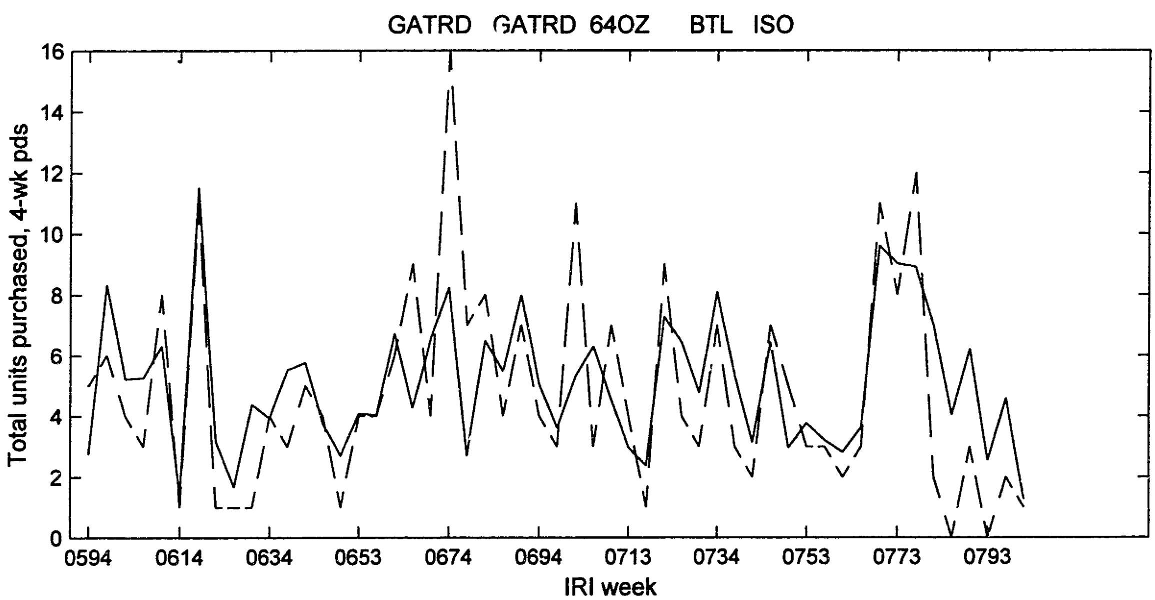


"-" is predicted sales level "--" is actual sales level

APPENDIX 3 (continued).

Results of the Overall Nested Model
2.83e+004 purchases opportunities

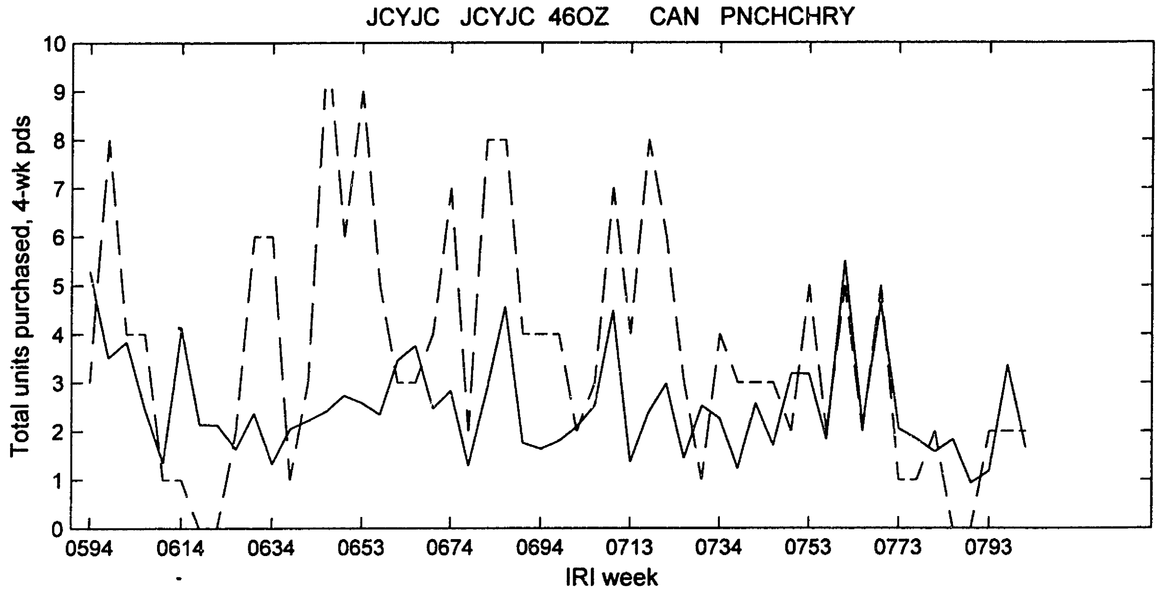




"-" is predicted sales level "--" is actual sales level

APPENDIX 3 (continued).

Results of the Overall Nested Model
2.83e+004 purchases opportunities



"-" is predicted sales level "--" is actual sales level

THESIS PROCESSING SLIP

FIXED FIELD ill ______ name ______

index ______ biblio ______

► COPIES (Archives) Aero (Dewey) Eng Hum

Lindgren Music Rotch Science

TITLE VARIES ►☐ ______

NAME VARIES: ►☐ ______

IMPRINT (COPYRIGHT) ______

►COLLATION: 87ℓ

►ADD. DEGREE: ______ ► DEPT.: ______

SUPERVISORS: ______

NOTES:

cat'r: date:

page: J207

►DEPT: Mgt

►YEAR: 1997 ►DEGREE: M.S.

►NAME: XIAO, Weizhong