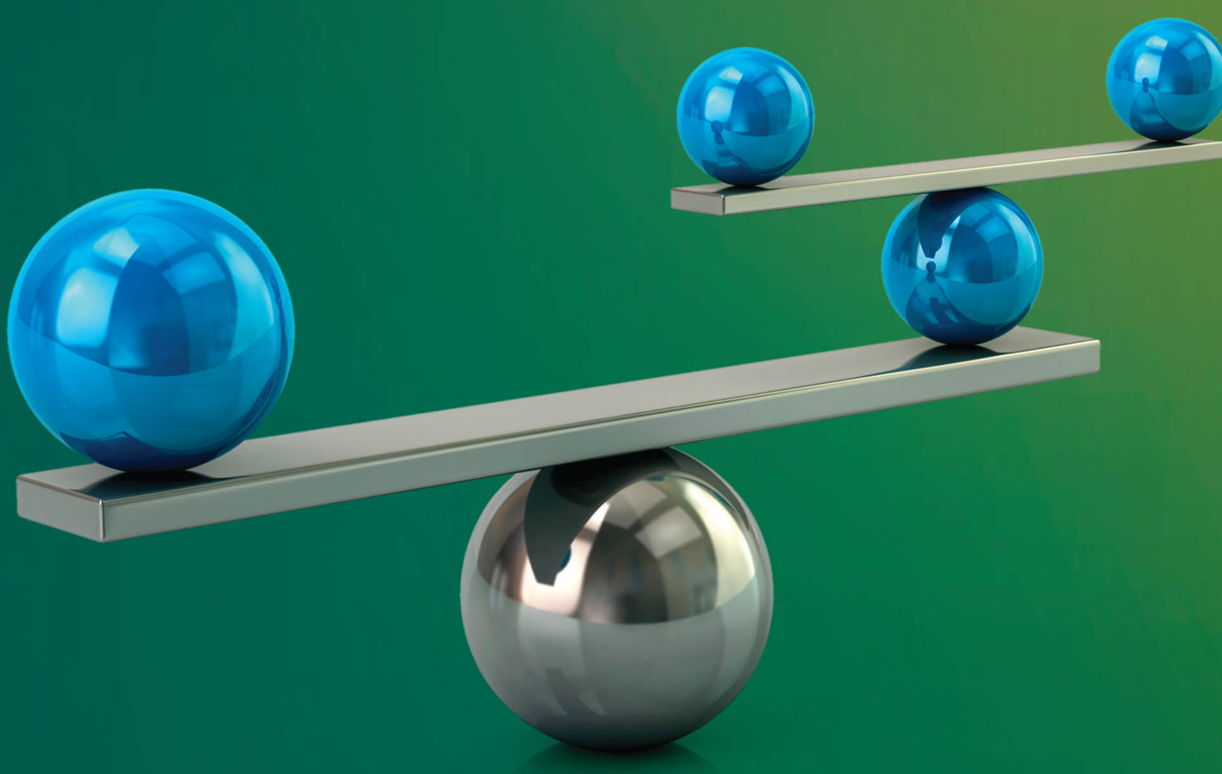
Applied Econometrics with SAS®

Modeling Demand, Supply, and Risk



Barry K. Goodwin
A. Ford Ramsey
Jan Chvosta

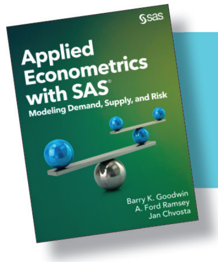
The correct bibliographic citation for this manual is as follows: Goodwin, Barry K., A. Ford Ramsey, and Jan Chvosta. 2018. *Applied Econometrics with SAS®: Modeling Demand, Supply, and Risk*. Cary, NC: SAS Institute Inc.

Applied Econometrics with SAS®: Modeling Demand, Supply, and Risk

ISBN 978-1-62960-407-7 (Hard copy)
ISBN 978-1-63526-050-2 (EPUB)
ISBN 978-1-63526-051-9 (MOBI)
ISBN 978-1-63526-052-6 (PDF)

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

March 2018


sas
Applied Econometrics with SAS
Modeling Demand, Supply, and Risk
Barry K. Goodwin
A. Ford Ramsey
Jan Chvosta

Contents

About the Authors

Barry K. Goodwin, PhD, is the William Neal Reynolds Distinguished Professor in the Department of Agricultural and Resource Economics, as well as a Graduate Alumni Distinguished Professor in the Department of Economics, at North Carolina State University, where he teaches and conducts research on policy, risk, trade, and applied econometrics. He is a fellow and past president of the Agricultural and Applied Economics Association. He has coauthored three books and more than 150 peer-reviewed journal articles, receiving numerous research awards, including best article awards from the *American Journal of Agricultural Economics*, the *Journal of Agricultural and Resource Economics*, and the *Canadian Journal of Agricultural Economics*. Professor Goodwin completed his PhD in economics at North Carolina State University in 1988.

A. Ford Ramsey, PhD, is an assistant professor in the Department of Agricultural and Applied Economics at Virginia Polytechnic Institute and State University. His research interests are risk and insurance, agricultural economics, and applied econometrics. He has been a SAS user since 2012 and was a consultant for SAS from 2014 to 2017. He was honored with an Australian Agricultural and Resource Economics Society / Agricultural and Applied Economics Association *Heading South Award* and the National Science Foundation *East Asia and Pacific Summer Institutes Fellowship*. Professor Ramsey completed his PhD in economics at North Carolina State University in 2017.

Jan Chvosta, PhD, is Senior Manager of Economics Technology Solutions, Advanced Analytics, in Research and Development at SAS. His research interests are econometric modeling, cross-sectional and panel data analysis, spatial econometrics, and high-performance computing. Dr. Chvosta was born in the Czech Republic. Shortly after receiving an MS in Informatics from the Czech University of Life Sciences in 1994, he relocated to the United States to pursue a career in economics and econometrics. He received an MS in applied economics from Montana State University in 1996. In 2004, he graduated from North Carolina State University with an MS in applied statistics and a PhD in economics.

Learn more about these authors by visiting their author pages, where you can download free book excerpts, access example code and data, read the latest reviews, get updates, and more:
http://support.sas.com/goodwin
http://support.sas.com/ramsey
http://support.sas.com/chvosta

Chapter 3
Empirical Approaches to Demand Analysis

Contents

3.1 Overview

In the previous chapter, we stated the preference axioms that are sufficient to generate the theory of demand. Some key results of the theory were presented; you will want to take these results into account when conducting empirical studies of demand. Empirical demand analysis has a history that is as long and storied as its theoretical counterpart. In 1699, well before the publication of Adam Smith's *An Inquiry into the Nature and Causes of the Wealth of Nations*, Charles Davenant published a demand schedule for corn using data from Gregory King. Both Davenant and King made statements that roughly correspond to what we now call the Law of Demand (Evans 1967). In this case, empirical observation preceded strictly theoretical developments. Since that time, economists have examined many aspects of demand, including the effects of income on consumption, the effects of prices on consumption, and a number of other important questions.

Much of the early work in demand focused on agricultural commodities. This focus was largely for practical reasons; agricultural goods are mostly homogeneous and consumer preferences for these goods are relatively stable over time. Violation of either of these conditions would complicate statistical analysis and hamper investigation of underlying economic fundamentals. The problems inherent in analyzing heterogeneous goods or goods with unstable preferences have only been partially resolved. Even today, agricultural products remain a popular topic for applied economists. Some of this popularity must also be due to the relative ease with which price and quantity data on agricultural products can be obtained. The United States Department of Agriculture and Bureau of Labor Statistics have a number of data sources that are freely available to the public.

In this chapter, we present and detail a set of commonly used functional forms for demand analysis. The theory of demand says nothing about functional form beyond properties of derivatives. In parametric empirical analyses, it is up to the analyst to choose a functional form which is used to used to estimate elasticities of demand. We provide the derivation of several of the forms and give some account of their development. Readers interested in more technical aspects of demand analysis are advised to check the references in this chapter and at the end of the book. Our main concerns are the ability of these different functional forms to

1. Satisfy restrictions from economic theory
2. Allow for theoretical restrictions to be tested
3. Generate elasticities of demand

The first two items are intimately related to the material developed in the preceding chapter. The last embeds our concerns over the practicality of the proposed methods. It is all well and good if we can test the restrictions from theory, but the estimation of these forms should also be simple. All of the approaches in this chapter can be implemented and extended using SAS. This ease of use will allow you to quickly estimate and test many different models. Different models may be preferred depending on the underlying data, so this flexibility is important in applied analysis.

Apparent in the following applications, the restrictions of economic theory are often rejected. What are the implications of rejection? While there has been some debate on this topic, economists now seem to have widely agreed that rejection of the restrictions implied by economic theory does not constitute a rejection of the law of demand. Nor does rejection in the production context constitute a violation of basic economic concepts in production. Rather, rejection likely represents some underlying aberration in the aggregate data or statistical model. As shown by Kastens and Brester (1996), models with theoretical restrictions imposed can perform better at prediction than those without. As far as out of sample prediction is a valid indicator of the goodness of a model, models with restrictions imposed performed very well. The applied researcher need not be worried upon finding the restrictions of theory rejected. While methodological debates are beyond the scope of this text, an enlightening analysis of these topics can be found in Leontief (1993).

3.2 Double Logarithmic Demand Functions

Elasticities of demand are at the core of almost all applied demand studies. Recognizing that an elasticity is the logarithmic derivative of quantities and prices (or income), it becomes clear that demand functions incorporating logarithmic terms provide an easy way of recovering elasticity estimates. The simplest demand equations of this type are often termed “logarithmic” or “double log” demands. Part of the appeal of the double log form is that the parameters of the estimated equations may immediately be understood as price and income elasticities. Since the equations are linear in parameters, standard statistical techniques can be applied for their estimation.

There are a number of theoretical problems with the double log model, but it continues to be used in applications where a single demand equation is called for. One advantage is that it is easy to incorporate other demand shifters into the double log function. You might consider using double log demand equations when you are interested in analyzing demand for a single good or your data are limited. While your results should only be considered approximate, they can provide a good first look at demand relationships.

3.2.1 The Double Log Form

The double log demand function for good i is

$$\log(x_i) = \alpha_i + \eta_i \log(m) + \sum_{k=1}^{n} \epsilon_{ik} \log(p_k) \tag{3.1}$$

where m is income or expenditure on the set of goods under consideration and p_k denotes the price of good k. The quantity x_i is given in per capita terms, as is the case in almost all demand analyses, so that changes in population do not contaminate empirical results. You can already see that the own price and cross price elasticities are ϵ_{ik} and the expenditure elasticity is given by η_i. This result can be demonstrated by taking the derivative of equation 3.1 with respect to any of the logarithmic variables. Although the single equation double log model is the most applied variant of this form, you could also consider estimating a system of double log demands.

Whether you consider a single equation or system of equations, the double log model cannot satisfy many of the restrictions from demand theory. Even the simple adding–up property is not guaranteed to be satisfied. Deaton and Muellbauer (1980b) go through the short derivations necessary to obtain this result. Adding up will only hold in the unrealistic situation where all of the income elasticities of demand are equal to one. Because of this deficiency, most analysts now choose to use other models in demand analysis.

In fact, the only condition that can be imposed on the double log demands is homogeneity. Homogeneity in a double log demand function implies that

$$\epsilon_{ik} + \cdots + \epsilon_{in} + \eta_i = 0 \tag{3.2}$$

for all $k = 1, \ldots, n$. Two approaches can be pursued to achieve this result. The first is to deflate all prices and income by a single price or income. The second is to restrict the parameters during estimation, instead of deflating the monetary variables. Both methods are easily accomplished using PROC MODEL, but deflating has been more popular historically. The deflator must be one of the monetary variables in the demand equation. As shown in Alston, Chalfant, and Piggott (2002), it is inappropriate to deflate prices and income using a general price index. Using a general price index, the parameters of the demand equation cannot be interpreted as pure elasticities.

3.2.2 Empirical Analysis

We begin with a simple analysis of the demand for dairy products. The data contain information on per capita availability of plain whole milk, butter, cheddar cheese, and processed cheese in the United States. Milk availability is measured in gallons while availabilities of the other goods are measured in pounds. The data set also contains information on the average retail prices of these products across all U.S. cities. Consumption information was obtained from the United States Department of Agriculture's ERS Food Availability (Per Capita) Data System (FADS) while prices were taken from the Bureau of Labor Statistic's Consumer Price Index - Average Price Data. Data are available from 1996 to 2012, and with only sixteen years in the sample, it seems reasonable to assume that consumer preferences are relatively constant over this period.

```
proc print data = dairy(obs = 10);
run;
```

Figure 3.1 U.S. Dairy Consumption Data

Obs	year	cheddar	processed	butter	milk	p_butter	p_cheddar	p_processed	p_milk
1	1996	9.13	5.44	4.3	8.2	2.04675	3.248	3.06550	2.62317
2	1997	9.53	4.92	4.2	8.0	2.16783	3.220	3.33533	2.61400
3	1998	9.57	4.44	4.3	7.7	2.86333	3.548	3.44925	2.70375
4	1999	9.96	4.65	4.6	7.8	2.65300	3.770	3.58992	2.84275
5	2000	9.79	4.86	4.5	7.7	2.51983	3.830	3.81092	2.78067
6	2001	9.94	4.25	4.3	7.4	3.29983	4.027	3.69825	2.88425
7	2002	9.72	4.67	4.4	7.3	3.07283	4.218	3.87142	2.75725
8	2003	9.35	4.61	4.5	7.2	2.81292	3.948	3.88058	2.76108
9	2004	10.33	4.12	4.5	7.0	3.48950	4.273	3.77142	3.15592
10	2005	10.14	4.16	4.5	6.6	3.28142	4.382	3.98433	3.18683

We can see from Figure 3.1 that there is no explicit information on expenditures in this data set. We can form a dairy expenditure variable by calculating and summing the individual expenditures on each product in the category. Because the endogenous variable in the double log demand is the logarithm of quantity, this variable must also be formed in a DATA step before being passed to the MODEL procedure.

```
data dairy;
set dairy;
   expenditures = milk * p_milk + cheddar * p_cheddar + processed * p_processed
                  + butter * p_butter;
   log_milk = log(milk);
   log_cheddar = log(cheddar);
   log_processed = log(processed);
   log_butter = log(butter);
run;
```

You now have all the variables necessary to estimate a simple system of double log demands. The following statements use PROC MODEL to estimate each of the four equations without the imposition of homogeneity. We could also estimate these demands using PROC REG, but the flexibility and syntax of PROC MODEL will be useful when we estimate more complicated demand systems later in the chapter.

```
/* Unrestricted Double Log Demands */
proc model data = dairy;
   parameters am bm gmm gmc gmp gmb
              ac bc gcm gcc gcp gcb
              ap bp gpm gpc gpp gpb
              ab bb gbm gbc gbp gbb;
   endogenous log_milk log_cheddar log_processed log_butter;
   exogenous p_milk p_cheddar p_processed p_butter expenditures;

   log_milk      = am + bm * log(expenditures) + gmm * log(p_milk)
                   + gmc * log(p_cheddar) + gmp * log(p_processed)
                   + gmb * log(p_butter);
   log_cheddar   = ac + bc * log(expenditures) + gcm * log(p_milk)
                   + gcc * log(p_cheddar) + gcp * log(p_processed)
                   + gcb * log(p_butter);
   log_processed = ap + bp * log(expenditures) + gpm * log(p_milk)
                   + gpc * log(p_cheddar) + gpp * log(p_processed)
                   + gpb * log(p_butter);
   log_butter    = ab + bb * log(expenditures) + gbm * log(p_milk)
                   + gbc * log(p_cheddar) + gbp * log(p_processed)
                   + gbb * log(p_butter);
```

```
   fit log_milk log_cheddar log_processed log_butter / ols
       outest = dairy_estimates;
run;
```

The DATA option specifies the data set that the procedure will utilize. The PARAMETERS statement is used to specify the parameters of the model, the ENDOGENOUS statement specifies the endogenous variables, and the EXOGENOUS statement specifies the exogenous variables. In most cases, SAS will be able to determine the exogenous and endogenous variables from the specified equations. We include these explicit statements for clarity and readability of code. Each equation has $n + 2$ parameters, so that the system as a whole has $n(n + 2)$ parameters.

The system of equations to be estimated is stated in the body of PROC MODEL. The FIT statement tells SAS which equations to estimate. The OLS option accompanies the FIT statement and instructs the procedure to estimate the equations by ordinary least squares. The parameter estimates from the procedure are placed in a data set called dairy_estimates and can be seen in Figure 3.2.

Figure 3.2 Parameter Estimates for Double Log Demands

The MODEL Procedure

Nonlinear OLS Parameter Estimates

Parameter	Estimate	Approx Std Err	t Value	Approx Pr > |t|
am	-4.81513	1.0396	-4.63	0.0007
bm	2.336434	0.2962	7.89	<.0001
gmm	-0.93826	0.1117	-8.40	<.0001
gmc	-1.38695	0.1172	-11.83	<.0001
gmp	-0.57894	0.1599	-3.62	0.0040
gmb	-0.04461	0.0737	-0.61	0.5571
ac	0.478044	1.3565	0.35	0.7312
bc	0.359616	0.3865	0.93	0.3721
gcm	0.248057	0.1457	1.70	0.1167
gcc	-0.45997	0.1530	-3.01	0.0119
gcp	0.411016	0.2086	1.97	0.0745
gcb	0.02284	0.0961	0.24	0.8165
ap	-4.8589	2.1881	-2.22	0.0483
bp	2.106622	0.6234	3.38	0.0061
gpm	-0.64917	0.2350	-2.76	0.0185
gpc	-0.84012	0.2468	-3.40	0.0059
gpp	-0.59288	0.3365	-1.76	0.1058
gpb	-0.46745	0.1550	-3.02	0.0118
ab	1.176331	1.0682	1.10	0.2943
bb	-0.05548	0.3043	-0.18	0.8587
gbm	0.10167	0.1147	0.89	0.3945
gbc	0.718218	0.1205	5.96	<.0001
gbp	-0.19423	0.1643	-1.18	0.2620
gbb	-0.25627	0.0757	-3.39	0.0061

It is encouraging that all of the own-price elasticities are negative with ϵ_{mm} = gmm = -0.93826, ϵ_{cc} = gcc = -0.45997, ϵ_{pp} = gpp = -0.59288, and ϵ_{bb} = gbb = -0.25627. All of the own-price elasticity estimates are also significant at the 5% level except for the own-price elasticity of processed cheese. While all of the goods are own-price inelastic, milk is the most price elastic followed by processed cheese, cheddar cheese, and butter. Milk is a complement for all of the other products, though the relationship is statistically significant only for the two types of cheese.

Many of the cross price elasticities are not statistically significant. There are also some general inconsistencies in the estimates as a result of our inability to impose symmetry. Cheddar cheese is a substitute for processed cheese, but processed cheese is a complement for cheddar cheese. Milk and processed cheese are both classified as luxury goods with respect to income. As there is probably not much income variation in this short time series data, it should not be surprising that only two of the income elasticities are significant.

We should estimate the model with homogeneity to ensure some consistency with theory. Homogeneity is imposed by deflating all prices and expenditure by a single price. The following DATA step deflates all of the prices and expenditures by the price for butter.

```
data dairy;
set dairy;
   expenditures = expenditures / p_butter;
   p_milk = p_milk / p_butter;
   p_cheddar = p_cheddar / p_butter;
   p_processed = p_processed / p_butter;
run;
```

Modifying the demand equations to account for the deflated prices, the following code estimates the same four equations with homogeneity. The price of butter is always one for every observation so the log price of butter is zero. Butter is the numeraire good which means that the relative prices of milk, cheddar, and processed products are all expressed in terms of the price of butter. Notice that the individual equations now have $n + 1$ parameters for a total of $n(n + 1) = 20$ parameters across the system.

```
/* Restricted Double Log Demands */
/* Homogeneity */
proc model data = dairy;
   parameters am bm gmm gmc gmp
              ac bc gcm gcc gcp
              ap bp gpm gpc gpp
              ab bb gbm gbc gbp;
   endogenous log_milk log_cheddar log_processed log_butter;
   exogenous p_milk p_cheddar p_processed p_butter expenditures;

   log_milk      = am + bm * log(expenditures) + gmm * log(p_milk)
                 + gmc * log(p_cheddar) + gmp * log(p_processed);
   log_cheddar   = ac + bc * log(expenditures) + gcm * log(p_milk)
                 + gcc * log(p_cheddar) + gcp * log(p_processed);
   log_processed = ap + bp * log(expenditures) + gpm * log(p_milk)
                 + gpc * log(p_cheddar) + gpp * log(p_processed);
   log_butter    = ab + bb * log(expenditures) + gbm * log(p_milk)
                 + gbc * log(p_cheddar) + gbp * log(p_processed);

   fit log_milk log_cheddar log_processed log_butter / ols
       outest = dairy_estimates;
run;
```

Elasticities for butter can be recovered after the fact using the homogeneity restrictions. The easiest way to do this is to read all of the parameter estimates into PROC IML and then construct a matrix of elasticities. The following code reads in the elasticity estimates, calculates the missing elasticities for butter, constructs a matrix of elasticities, and then prints the matrix of price elasticities.

```
proc iml;
   use dairy_estimates;
   read all var _ALL_;
   close dairy_estimates;

   gmb = 0 - bm - gmm - gmc - gmp;
   gcb = 0 - bc - gcm - gcc - gcp;
   gpb = 0 - bp - gpm - gpc - gpp;
   gbb = 0 - bb - gbm - gbc - gbp;

   price_elasticities = (gmm||gmc||gmp||gmb)//
                        (gcm||gcc||gcp||gcb)//
                        (gpm||gpc||gpp||gpb)//
                        (gbm||gbc||gbp||gbb);

   income_elasticities = (bm||bc||bc||bb);

   factors = {"Milk" "Cheddar" "Processed" "Butter"};

    print price_elasticities[label = "Price Elasticities of Demand"
                             rowname = factors colname = factors format = d7.3],
          income_elasticities[label = "Income Elasticities of Demand"
                              colname = factors format = d7.3];
quit;
```

Figure 3.3 Elasticity Matrix

Price Elasticities of Demand

	Milk	Cheddar	Processed	Butter
Milk	-0.839	-1.835	-0.153	-0.192
Cheddar	0.154	-0.0346	0.00672	0.163
Processed	-0.578	-1.164	-0.285	-0.574
Butter	0.0509	0.948	-0.412	-0.181

Income Elasticities of Demand

Milk	Cheddar	Processed	Butter
3.019	-0.289	-0.289	-0.406

The estimates with homogeneity imposed are also unsatisfactory. It doesn't seem reasonable that all of the dairy products except milk are inferior, or that plain whole milk is a luxury good. Similar inconsistencies arise in substitution relationships as we observed in the unrestricted model. Possibly weak data, coupled with a model that cannot satisfy the requirements of demand theory in general, lead to questionable estimates of elasticities. These weaknesses lead us to consider models that can provide a more satisfactory analysis.

3.3 Rotterdam Model

A major advance in demand system modeling was the development of the Rotterdam model by Theil (1965) and Barten (1964). The name for the model derives from the city of Rotterdam, where both Theil and Barten were stationed for a time. Unlike the double log demand functions, the Rotterdam model is a system-wide approach to demand. Its derivation is firmly rooted in the consumer's maximization problem. For these reasons the Rotterdam model continues to be popular for purposes of demand analysis and testing of economic theory.

In contrast to the linear expenditure system of Stone (1954), the Rotterdam model starts with very general consumer demand functions, and then generalizes up to the consumer's utility function. As Clements and Gao (2015) note, this gives the Rotterdam model a certain type of directness, in the sense that the researcher starts with a basic demand function and at the end of the exercise obtains estimates of demand functions. The form of the utility function is never explicitly given, but work by McFadden (1964) has shown that specific forms for the demand functions may imply restrictions on the consumer's preferences. In spite of the rather minor restrictions on preferences that the Rotterdam model can imply, it is one of the most easily applied and theoretically consistent approaches to the study of demand.

Unlike the double log model, you will see with the Rotterdam model that it is possible to empirically test many implications of economic theory. We saw in the preceding chapter that demand systems must satisfy a host of conditions. The Rotterdam model allows for these conditions to be imposed, or for the system to be estimated without restrictions to test the consistency of economic theory with the data. The Rotterdam model is also specified in terms of first differences of the variables. This makes it a particularly attractive model when time series of prices and income are nonstationary. Early critics of the model noted that its properties hold only in very restrictive cases and this motivated later authors to obtain strong theoretical properties under weaker assumptions. Readers interested in these developments can find additional information in Barnett (1979) and Mountain (1988).

3.3.1 Absolute Price and Relative Price Rotterdam Formulations

There are two distinct versions of the Rotterdam model in use: the Absolute Price model and the Relative Price model. The derivation of either version of the model starts from the consideration of the total differential of a demand function. For this reason, the Rotterdam model and several other approaches to demand analysis have been classified as differential approaches by Barnett and Serletis (2009). Starting from a standard demand function for good i,

$$x_i = x_i(\mathbf{p}, m) \tag{3.3}$$

total differentiation yields

$$dx_i = \frac{\partial x_i}{\partial m} dm + \sum_{k=1}^{n} \frac{\partial x_i}{\partial p_k} dp_k \tag{3.4}$$

The rest of the derivation relies on some of the theoretical developments of the previous chapter. From basic calculus, we know that $\frac{d\log(x)}{dx} = \frac{1}{x}$ which is equivalent to $dx = xd\log(x)$. Substituting for the differentials in the equation above yields

$$x_i d\log(x_i) = \frac{\partial x_i}{\partial m} md\log(m) + \sum_{k=1}^{n} \frac{\partial x_i}{\partial p_k} p_k d\log(p_k) \tag{3.5}$$

Then dividing the whole equation by x_i gives

$$d\log(x_i) = \frac{\partial x_i}{x_i \partial m} md\log(m) + \sum_{k=1}^{n} \frac{\partial x_i}{x_i \partial p_k} p_k d\log(p_k) \tag{3.6}$$

You might be concerned that we are moving further and further away from a usable result, but this is not the case. There are several elasticities hiding in equation 3.6. Note that $\frac{\partial x_i}{\partial m}\frac{m}{x_i}$ is the income elasticity of demand and $\frac{\partial x_i}{\partial p_k}\frac{p_k}{x_i}$ is the price elasticity of demand. Simplifying the elasticity terms, the equation can then be written as

$$d\log(x_i) = \eta_i d\log(m) + \sum_{k=1}^{n} \epsilon_{ik} d\log(p_k) \tag{3.7}$$

Substituting for the Slutsky equation,

$$d\log(x_i) = \eta_i \left(d\log(m) - \sum_{k=1}^{n} s_k d\log(p_k) \right) + \sum_{k=1}^{n} \epsilon^*_{ik} d\log(p_k) \tag{3.8}$$

where s_k is the budget share of good k, and ϵ^*_{ik} are the compensated own and cross price elasticities of demand. The model is formulated in continuous time, but we never have economic data in continuous time. In many aggregate demand studies, we will only have data every year or month. Certainly at the time the Rotterdam model was formulated, this was the case. The empirical version of the Rotterdam model approximates the theoretical version in discrete time with $c_{ik} = s_i\epsilon^*_{ik}$, $b_i = p_i \partial f/\partial m$, $s_t = 0.5(s_t + s_{t-1})$, and $\Delta\log\bar{x}_t = \Delta\log x_t - \sum_{k=1}^{n} \bar{s}_t \Delta\log p_{kt}$. The absolute price version of the Rotterdam model can then be stated as

$$\bar{s_{it}}\Delta\log x_{it} = b_i \Delta\log \bar{m_{it}} + \sum_{k=1}^{n} c_{ik}\Delta\log p_{kt} + \epsilon_{it} \tag{3.9}$$

where Δ is a difference operator over time and an error term has been appended for estimation.

The curvature restrictions can then be tested. Homogeneity holds if $\sum_k c_{ik} = 0$ for all i. Negativity if $c_{ii} < 0$ for all i. And symmetry holds if $c_{ik} = c_{ki}$ for all i, k with $i \neq k$. The Rotterdam model is far more useful, at least compared to the double log demands, because it presents an avenue for such tests.

The elasticities of the absolute price version of the Rotterdam model can be calculated directly from the parameters. One assumption inherent in the model is that the parameters are constant. The elasticities must then be evaluated at various points in the sample, most often at the sample average. Budget shares are required for the elasticities that follow, and typically the elasticities are evaluated at the mean budget shares. The income and compensated price elasticities are fairly direct, while the uncompensated price elasticities are calculated using the Slutsky equation.

$$\epsilon^*_{ik} = c_{ik}/s_i \tag{3.10}$$

$$\eta_i = \beta_i/s_i \tag{3.11}$$

$$\epsilon_{ik} = \epsilon_{ij} - s_j\eta_i = c_{ij}/s_i - s_j\beta_i/s_i \tag{3.12}$$

One limitation of the demand system defined by equation 3.9 is that the number of parameters grows rapidly as more goods are added to the model. Partly for this reason, a relative price version of the Rotterdam model was developed. While the absolute price version is linear in parameters, the relative price version is nonlinear. More details on the differences between the two models are given in Barnett and Serletis (2009). Because the estimation of systems of equations with many parameters has been greatly simplified, we do not estimate the relative version of the Rotterdam model in the empirical example that follows. However, it is instructive to show how the model can be derived. From the absolute version of the Rotterdam, and equation 3.10, we know

$$c_{ik} = s_i \epsilon^*_{ik} \tag{3.13}$$

Barten (1964) showed that the substitution effect of a change in the price of good k and demand for good i can be expressed as

$$S_{ik} = \lambda U^{-1}_{ik} - \frac{\lambda}{\partial\lambda/\partial m}\frac{\partial x_i}{\partial m}\frac{\partial x_k}{\partial m} \tag{3.14}$$

where λ is the marginal utility of income. In this case, U_{ij} is the Hessian matrix of the utility function which gives the change in marginal utility as consumption of a good varies. Putting these two elements together

$$c_{ik} = \frac{p_i p_k}{m} S_{ik} = \frac{p_i p_k}{m}[\lambda U^{-1}_{ik} - \frac{\lambda}{\partial\lambda/\partial m}\frac{\partial x_i}{\partial m}\frac{\partial x_k}{\partial m}] \tag{3.15}$$

This can be rewritten as

$$c_{ik} = v_{ik} - \theta b_i b_k \tag{3.16}$$

where:

$$v_{ik} = \frac{p_i p_k}{m}\lambda U^{-1}_{ik}, \quad \theta = \frac{\lambda}{\partial\lambda/\partial m}, \quad b_i = p_i \frac{\partial x_i}{\partial m} \tag{3.17}$$

Now, consider the absolute price version of the Rotterdam:

$$s_i d \ln x_i = b_i d \ln \bar{m} + \sum_k (v_{ik} - \theta b_i b_k) d \ln p_k \tag{3.18}$$

We know that $\sum_k c_{ik} = 0$, and $\sum_k b_k = 1$, which implies that

$$\sum_k v_{ik} = \theta b_i \tag{3.19}$$

Considering again the expression for the Rotterdam

$$s_i d \ln x_i = b_i d \ln \bar{m} + \sum_k (v_{ik} - \theta b_i b_k) d \ln p_k \tag{3.20}$$

Multiply this out and we obtain

$$s_i d \ln x_i = b_i d \ln \bar{m} + \sum_k v_{ik} d \ln p_k - \theta b_i \sum_k b_k d \ln p_k \tag{3.21}$$

Now, the trick—you know that $\sum_k b_k d \ln p_k$ is the same as $\sum_i b_i d \ln p_i$ for any i or k. To prevent confusion, change the subscript in this last summation from k to i and use the fact that $\sum_k v_{ik} = \theta b_i$

$$s_i d \ln x_i = b_i d \ln \bar{m} + \sum_k v_{ik} d \ln p_k - \sum_k v_{ik} \sum_i b_i d \ln p_i, \tag{3.22}$$

which is equivalent to

$$s_i d \ln x_i = b_i d \ln \bar{m} + \sum_k v_{ik} (d \ln p_k - \sum_i b_i d \ln p_i), \tag{3.23}$$

This gives us the relative price version of the Rotterdam model. Equation 3.23 implies that with want independence, demand for a commodity can be expressed as a function of real income and the relative price of good. As opposed to an average price index, the denominator in the relative price is a marginal price index. Likewise, restrictions on preferences can imply restrictions on the coefficients of the model. While it is easier to estimate the relative model, improved computing power has obviated the need for this simplification. With adequate data and currently available statistical software, the absolute version of the model is preferred.

3.3.2 Empirical Analysis

In this application, the Rotterdam model is applied to quarterly meat consumption data from 1975 to 1999. Figure 3.4 shows the first few observations from the data set. The data include variables on U.S. population (in millions of persons), per capita consumption of meats in pounds, average nominal retail prices of meats in cents per pound, the value of the consumer price index, and per capita nominal consumption expenditures on all goods. This is enough data to estimate the Rotterdam model, but not without some minor adjustments. Meat prices in the data set are nominal and must be normalized by the consumer price index in order to obtain an accurate depiction of prices over time. We also need to construct expenditures and expenditure shares for each type of meat. As this is the first true demand system that we have estimated, note that the number of observations in the systems context is equal to the number of observations in the sample multiplied by the number of equations in the system.

```
proc print data = meat(obs = 10);
run;
```

Figure 3.4 Quarterly Meat Consumption Data

Obs	year	qtr	pop	q_beef	q_pork	q_chick	q_turk	p_beef	p_pork	p_chick	p_turk	cpi	pc_exp
1	1975	1	215.132	22.0991	11.8074	9.2631	1.0724	134.833	120.8	58.9000	72.1333	52.43	1143.95
2	1975	2	215.646	21.0304	11.0085	10.1645	1.3916	152.633	129.8	58.9667	70.8333	53.23	1175.19
3	1975	3	216.294	22.2717	9.6583	10.2047	1.8937	163.200	157.4	68.9333	73.8667	54.37	1210.39
4	1975	4	216.851	22.7502	10.4302	9.7381	3.9068	158.167	161.8	66.3000	78.2667	55.23	1240.48
5	1976	1	217.315	23.8753	10.9390	10.3294	1.1450	148.700	149.4	61.9333	78.1000	55.77	1278.21
6	1976	2	217.773	22.8964	10.3597	10.9598	1.5151	148.200	146.3	60.7000	77.0000	56.47	1298.49
7	1976	3	218.337	24.4652	11.0404	11.0310	2.0091	142.800	145.1	60.9000	76.8667	57.37	1329.14
8	1976	4	218.920	23.1085	13.1403	10.1654	4.2247	142.933	126.6	55.1333	78.2333	58.03	1365.91
9	1977	1	219.424	22.9976	11.9593	10.3395	1.2198	142.167	127.2	58.2667	76.0667	59.03	1403.22
10	1977	2	219.953	22.6131	11.4488	11.2841	1.3872	143.933	128.8	60.8000	74.1000	60.33	1432.47

Using PROC MEANS to summarize the meat data, Figure 3.5 shows that over the entirety of the sample, the price for beef has been higher than the price of other meats. Chicken is the cheapest meat. There are 99 observations in the sample corresponding to roughly 25 years of quarterly data.

```
proc means data = meat
   n mean max min range std;
run;
```

Figure 3.5 Summary Statistics for Meat Data

The MEANS Procedure

Variable	N	Mean	Maximum	Minimum	Range	Std Dev
year	99	1986.88	1999.00	1975.00	24.0000000	7.1817125
qtr	99	2.4848485	4.0000000	1.0000000	3.0000000	1.1190706
pop	99	243.4708131	273.5195000	215.1315000	58.3880000	17.0189387
q_beef	99	18.7595734	24.4652000	15.8915000	8.5737000	2.1352497
q_pork	99	12.6880525	14.8637000	9.6583000	5.2054000	0.9764712
q_chick	99	14.4615214	19.9552523	9.2631000	10.6921523	2.8201397
q_turk	99	3.4497231	6.4864000	1.0724000	5.4140000	1.4057030
p_beef	99	242.2377088	300.4000000	134.8333000	165.5667000	45.1326845
p_pork	99	189.8373737	248.1000000	120.8000000	127.3000000	34.5168595
p_chick	99	81.5728949	107.3300000	55.1333000	52.1967000	13.4142885
p_turk	99	95.5736020	109.0000000	70.8333000	38.1667000	9.3858706
cpi	99	113.6405051	167.2300000	52.4300000	114.8000000	34.7966874
pc_exp	99	3232.37	5761.66	1143.95	4617.70	1334.28

The following code uses arrays to simplify the process of constructing expenditures and real prices. The nominal expenditures are quantities of beef multiplied by their prices. Real prices are obtained by dividing nominal prices by the consumer price index (CPI); real expenditures are simply the product of the quantities of meat consumed and their real prices. Total expenditure on meat is obtained by summing expenditures on the individual categories. Because the demand system is expressed in log differences, you also need to create the differenced variables in the DATA step.

```
data meat;
set meat;
   array prices {4} p_beef p_pork p_chick p_turk;
   array quantities {4} q_beef q_pork q_chick q_turk;
   array expenditures {4} exp_beef exp_pork exp_chick exp_turk;
   array shares {4} s_beef s_pork s_chick s_turk;
   array real_prices {4} rp_beef rp_pork rp_chick rp_turk;
   array dlog_quantities {4} dlq_beef dlq_pork dlq_chick dlq_turk;
   array dlog_prices {4} dlp_beef dlp_pork dlp_chick dlp_turk;
   array d_shares {4} ds_beef ds_pork ds_chick ds_turk;
   do i = 1 to 4;
      real_prices{i} = prices{i} / cpi;
      dlog_quantities{i} = dif(log(quantities{i}));
      dlog_prices{i} = dif(log(prices{i}));
   end;
   do i = 1 to 4;
      expenditures{i} = quantities{i} * real_prices{i};
   end;
   exp_meat = sum(exp_beef, exp_pork, exp_chick, exp_turk);
   do i = 1 to 4;
      shares{i} = expenditures{i} / exp_meat;
      d_shares{i} = .5 * (shares{i} + lag(shares{i})) * dlog_quantities{i};
   end;
   dexp_meat = sum(ds_beef, ds_pork, ds_chick, ds_turk);
   date = intnx( 'qtr', '1jan1975'd, _n_-1 );
run;
```

You can use the SGPLOT procedure to examine the behavior of per capita consumption and real price changes over time. When creating the graphs, the VALUESFORMAT option specified that the x axis dates are displayed by year and not another date format.

```
proc sgplot data = meat;
   series x = date y = q_beef / markers markerattrs = (symbol = circle);
   series x = date y = q_pork / markers markerattrs = (symbol = square);
   series x = date y = q_chick / markers markerattrs = (symbol = star);
   series x = date y = q_turk / markers markerattrs = (symbol = diamond);
   title 'Per Capita Consumption of Meats';
   xaxis label = 'Year' valuesformat = year4.;
   yaxis label = 'Pounds';
run;

proc sgplot data = meat;
   series x = date y = rp_beef / markers markerattrs = (symbol = circle);
   series x = date y = rp_pork / markers markerattrs = (symbol = square);
   series x = date y = rp_chick / markers markerattrs = (symbol = star);
   series x = date y = rp_turk / markers markerattrs = (symbol = diamond);
   title 'Average Real Retail Price of Meats';
   xaxis label = 'Year' valuesformat = year4.;
   yaxis label = 'Cents/Pound';
run;
```

One interesting feature of these graphs is the seasonal nature of consumption. Consumption of turkey tends to spike in winter, likely from consumption of whole turkeys at Thanksgiving and Christmas. The average real retail price of meats has gone down over time. The price of turkey and chicken per pound is now equal. Note the dramatic increase in the consumption of chicken over the 25 year period. Consumption has increased from around 10 pounds per quarter to nearly 20 pounds. This increase in the consumption of chicken has been accompanied by a decrease in beef consumption.

Figure 3.6 Changes in Consumption and Prices Over Time

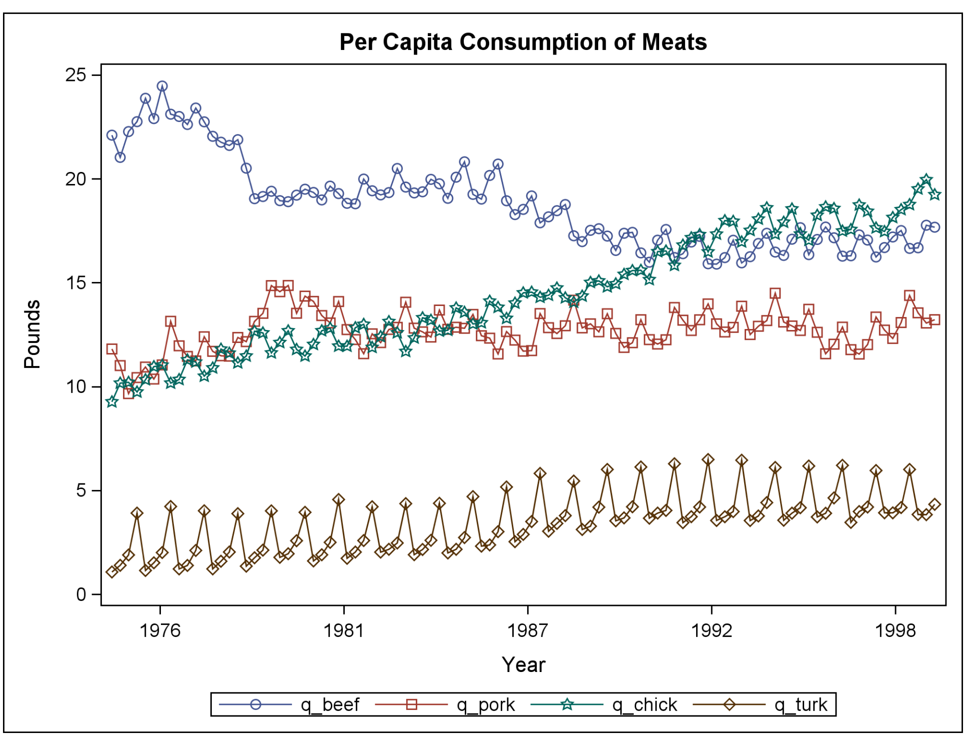
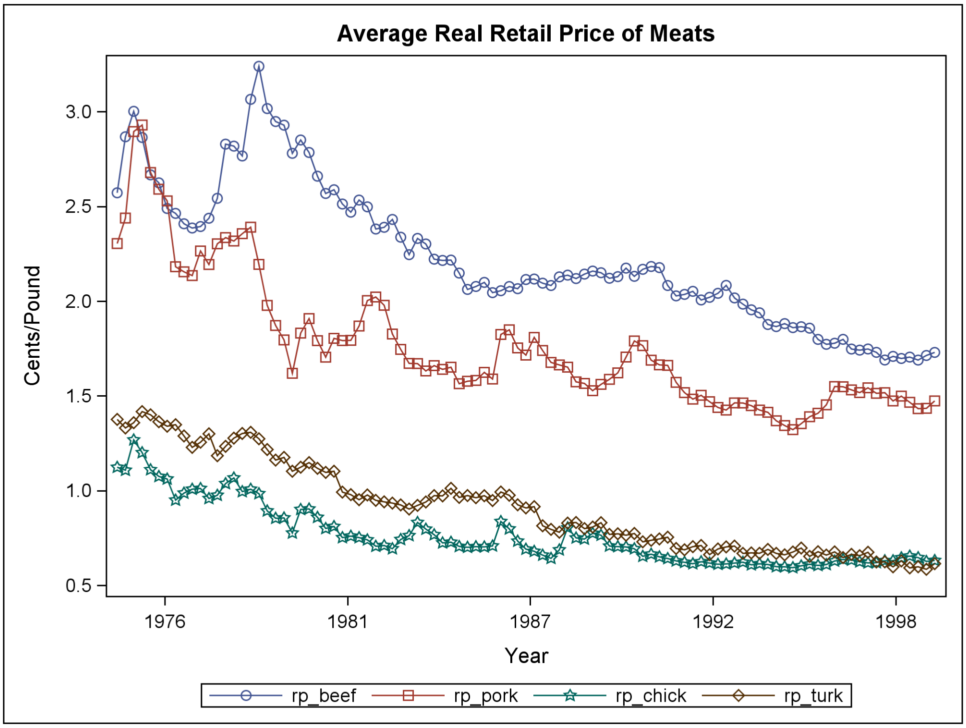


Figure 3.6 *continued*



The following code estimates the unrestricted Rotterdam model. Because the error terms are not linearly independent the equation for turkey is dropped. This singularity problem is equivalent to invariance when using maximum likelihood estimation. Barten (1964) shows that the choice of the deleted equation does not affect the parameter values. There are then eighteen parameters to be estimated using the MODEL procedure. The model is estimated using full information maximum likelihood, although it could also be estimated through seemingly unrelated regression techniques. The two approaches are equivalent and, as indicated by Barnett and Seck (2008), it does not matter which method is used. TEST statements specify likelihood ratio tests of the parameter restrictions for homogeneity, symmetry, and joint homogeneity and symmetry. The system consists of three equations with $n + 2$ parameters in each equation, so a total of $(n - 1)(n + 2) = 18$ parameters must be estimated.

```
/* Unrestricted Rotterdam Model */
/* Test for Homogeneity and Symmetry */
proc model data = meat;
   parameters ab bb gbb gbp gbc gbt
              ap bp gpb gpp gpc gpt
              ac bc gcb gcp gcc gct;
   endogenous ds_beef ds_pork ds_chick;
   exogenous dexp_meat dlp_beef dlp_pork dlp_chick dlp_turk;

   ds_beef  = ab + bb * dexp_meat + gbb * dlp_beef + gbp * dlp_pork
              + gbc * dlp_chick + gbt * dlp_turk;
   ds_pork  = ap + bp * dexp_meat + gpb * dlp_beef + gpp * dlp_pork
              + gpc * dlp_chick + gpt * dlp_turk;
   ds_chick = ac + bc * dexp_meat + gcb * dlp_beef + gcp * dlp_pork
              + gcc * dlp_chick + gct * dlp_turk;

   fit ds_beef ds_pork ds_chick / fiml outest = rott_unrest;

   test "Homogeneity"
```

```
         gbb + gbp + gbc + gbt = 0,
         gpb + gpp + gpc + gpt = 0,
         gcb + gcp + gcc + gct = 0, / lr;

      test "Symmetry"
         gbp = gpb,
         gbc = gcb,
         gpc = gcp, / lr;

      test "Joint Homogeneity and Symmetry"
         gbb + gbp + gbc + gbt = 0,
         gpb + gpp + gpc + gpt = 0,
         gcb + gcp + gcc + gct = 0,
         gbp = gpb, gbc = gcb, gpc = gcp, / lr;
   run;
```

Results from the estimation of the unrestricted model, and the likelihood ratio tests are shown in Figure 3.7. All three of the tests for homogeneity, symmetry, and joint homogeneity and symmetry are rejected. As noted in the introduction of the chapter, these rejections should not be viewed as completely invalidating the empirical model.

Figure 3.7 Estimates from Unrestricted Rotterdam Model

The MODEL Procedure

Nonlinear FIML Parameter Estimates

Parameter	Estimate	Approx Std Err	t Value	Approx Pr > |t|
ab	0.361906	0.0488	7.42	<.0001
bb	1.007012	0.0123	81.55	<.0001
gbb	0.07328	0.0877	0.84	0.4053
gbp	0.411893	0.1006	4.09	<.0001
gbc	0.321596	0.0980	3.28	0.0015
gbt	0.469816	0.1007	4.66	<.0001
ap	-0.18463	0.0282	-6.55	<.0001
bp	-0.03926	0.00973	-4.03	0.0001
gpb	-0.17859	0.0487	-3.67	0.0004
gpp	-0.46625	0.0561	-8.31	<.0001
gpc	-0.25703	0.0535	-4.81	<.0001
gpt	-0.33636	0.0537	-6.26	<.0001
ac	-0.05074	0.0427	-1.19	0.2376
bc	-0.0155	0.0121	-1.28	0.2048
gcb	0.147735	0.0640	2.31	0.0233
gcp	0.120819	0.0694	1.74	0.0849
gcc	0.017055	0.0655	0.26	0.7952
gct	-0.0459	0.0726	-0.63	0.5287

Test Results

Test	Type	Statistic	Pr > ChiSq	Label
Homogeneity	L.R.	43.89	<.0001	gbb + gbp + gbc + gbt = 0, gpb + gpp + gpc + gpt = 0, gcb + gcp + gcc + gct = 0
Symmetry	L.R.	41.40	<.0001	gbp = gpb, gbc = gcb, gpc = gcp
Joint Homogeneity and Symmetry	L.R.	81.68	<.0001	gbb + gbp + gbc + gbt = 0, gpb + gpp + gpc + gpt = 0, gcb + gcp + gcc + gct = 0, gbp = gpb, gbc = gcb, gpc = gcp

The homogeneity and symmetry restrictions can be imposed by using RESTRICT statements in the MODEL procedure. The following code is similar to the unrestricted Rotterdam model except for the addition of these statements. Once the restrictions are put in place, there are only 12 free parameters to estimate.

```
/* Restricted Rotterdam Model */
proc model data = meat;
   parameters ab bb gbb gbp gbc gbt
              ap bp gpb gpp gpc gpt
              ac bc gcb gcp gcc gct;
   endogenous ds_beef ds_pork ds_chick;
   exogenous dexp_meat dlp_beef dlp_pork dlp_chick dlp_turk;
   restrict gbb + gbp + gbc + gbt = 0,
            gpb + gpp + gpc + gpt = 0,
            gcb + gcp + gcc + gct = 0,
            gbp = gpb, gbc = gcb, gpc = gcp;

   ds_beef  = ab + bb * dexp_meat + gbb * dlp_beef + gbp * dlp_pork
              + gbc * dlp_chick + gbt * dlp_turk;
   ds_pork  = ap + bp * dexp_meat + gpb * dlp_beef + gpp * dlp_pork
              + gpc * dlp_chick + gpt * dlp_turk;
   ds_chick = ac + bc * dexp_meat + gcb * dlp_beef + gcp * dlp_pork
              + gcc * dlp_chick + gct * dlp_turk;

   fit ds_beef ds_pork ds_chick / fiml outest = rott_rest;

 run;
```

The parameter estimates from the restricted Rotterdam model are shown in Figure 3.8 and were stored in the data set rott_rest. As with most demand systems, the elasticities are functions of the parameters. The parameters themselves are usually not of primary interest given their difficult interpretation. Nonetheless, nearly all of the parameters in the restricted model are significant compared to the unrestricted model. This change is particularly evident in the parameters of the chicken demand equation.

Figure 3.8 The Restricted Rotterdam Model

The MODEL Procedure

Model Summary	
Model Variables	8
Endogenous	3
Exogenous	5
Parameters	18
Equations	3
Number of Statements	10

Figure 3.8 *continued*

Nonlinear FIML Parameter Estimates

Parameter	Estimate	Approx Std Err	t Value	Approx Pr > |t|	Label
ab	0.406016	0.0433	9.37	<.0001	
bb	1.035698	0.0146	71.17	<.0001	
gbb	-0.22452	0.0233	-9.64	<.0001	
gbp	0.121823	0.0140	8.71	<.0001	
gbc	0.072819	0.0202	3.61	0.0005	
gbt	0.029879	0.00514	5.81	<.0001	
ap	-0.19699	0.0231	-8.53	<.0001	
bp	-0.05206	0.00912	-5.71	<.0001	
gpb	0.121823	0.0140	8.71	<.0001	
gpp	-0.15467	0.0114	-13.52	<.0001	
gpc	0.023187	0.0114	2.04	0.0440	
gpt	0.009663	0.00455	2.12	0.0364	
ac	-0.07623	0.0424	-1.80	0.0755	
bc	-0.03161	0.0112	-2.82	0.0059	
gcb	0.072819	0.0202	3.61	0.0005	
gcp	0.023187	0.0114	2.04	0.0440	
gcc	-0.07755	0.0229	-3.38	0.0010	
gct	-0.01845	0.00703	-2.62	0.0101	
Restrict0	-59.8607	27.9149	-2.14	0.0312	gbb + gbp + gbc + gbt = 0
Restrict1	-5.11839	33.5960	-0.15	0.8799	gpb + gpp + gpc + gpt = 0
Restrict2	179.6788	35.2299	5.10	<.0001	gcb + gcp + gcc + gct = 0
Restrict3	44.25436	44.8651	0.99	0.3266	gbp = gpb
Restrict4	322.6924	51.3303	6.29	<.0001	gbc = gcb
Restrict5	324.3695	71.3950	4.54	<.0001	gpc = gcp

You can evaluate the elasticities of the Rotterdam model at different points in the data. The elasticity formulas require parameter estimates and a budget share; we evaluate the elasticities at the mean budget share over the 25 years of data. The following code calculates elasticities based on the restricted Rotterdam model. First use PROC MEANS to obtain the mean budget shares and then output this data into the data set mean shares.

```
proc means data = meat noprint mean;
   var s_beef s_pork s_chick s_turk;
   output out = meanshares mean = sb sp sc st;
run;
```

The following code uses PROC IML to form the elasticities. Read in the estimates of the parameters, calculate the parameters for the missing turkey equation, and then read in the mean budget shares. The rest of the code uses matrix operations to calculate the elasticities according to equation 3.10 and the following formulas. The elasticities are printed in matrix form.

```
proc iml;
   use rott_rest;
   read all var {gbb gbp gbc gbt gpp gpc gpt gcc gct bb bp bc};
   close rott_rest;

   gtt = 0 - gbt - gpt - gct;
   bt = 1 - bb - bp - bc;

   use meanshares;
```

```
         read all var {sb sp sc st};
   close meanshares;

   s = sb//sp//sc//st;
   b = bb//bp//bc//bt;

   print s;

   gij = (gbb||gbp||gbc||gbt)//
         (gbp||gpp||gpc||gpt)//
         (gbc||gpc||gcc||gct)//
         (gbt||gpt||gct||gtt);

   print gij;

   nk = ncol(gij);
   mi = -1#I(nk);
   uep = j(nk, nk, 0);
   cep = j(nk, nk, 0);
   exe = j(nk, 1, 0);

   do i=1 to nk;
   do j=1 to nk;
      cep[i,j] = (gij[i,j]/s[i]);
      uep[i,j] = (gij[i,j]/s[i]-s[j]#b[i]/s[i]);
      exe[i] = b[i]/s[i];
   end;
   end;

   meats = {"Beef" "Pork" "Chicken" "Turkey"};

   print
      uep[label="Uncompensated Price Elasticities of Demand"
          rowname = meats  colname = meats format = d7.3],
      cep[label="Compensated Price Elasticities of Demand"
          rowname = meats colname = meats format = d7.3],
      exe[label="Expenditure Elasticities" rowname = meats
          format = d7.3];
quit;
```

Figure 3.9 Elasticity Matrices

Uncompensated Price Elasticities of Demand

	Beef	Pork	Chicken	Turkey
Beef	-1.455	-0.324	-0.135	-0.0188
Pork	0.525	-0.490	0.107	0.0409
Chicken	0.639	0.229	-0.521	-0.123
Turkey	0.108	-0.104	-0.653	-0.594

Figure 3.9 *continued*

Compensated Price Elasticities of Demand				
	Beef	**Pork**	**Chicken**	**Turkey**
Beef	-0.419	0.227	0.136	0.0558
Pork	0.427	-0.542	0.0813	0.0339
Chicken	0.519	0.165	-0.552	-0.131
Turkey	0.774	0.250	-0.478	-0.546

Expenditure Elasticities	
Beef	1.933
Pork	-0.183
Chicken	-0.225
Turkey	1.243

According to the IML output in Figure 3.9, all of the own-price uncompensated and compensated elasticities are negative. According to the uncompensated price elasticities, the only type of meat that is own-price elastic is beef. Beef is also found, somewhat surprisingly, to be a complement to all other forms of meat. Beef and turkey are both classified as luxury goods. You may not be inclined to believe these elasticity estimates because of seasonality in the consumption of the meats, particularly turkey. Turkey consumption increases in winter around the holidays. We have also aggregated all cuts of beef into one category so lower quality cuts are included with high quality cuts. To remove the effects of seasonal behavior, we can append seasonal dummy variables to the model. In SAS, the DATA step is used to create the dummies. A boolean expression constructs the dummies for each quarter. The command creates a variable equal to one when the statement is true, and equal to zero otherwise. The parameter estimate on the first quarter dummy in Figure 3.10 is significant and validates our extension of the model to seasonal consumption behavior.

```
data meat;
set meat;
   qtr1 = (qtr = 1);
   qtr2 = (qtr = 2);
   qtr3 = (qtr = 3);
   qtr4 = (qtr = 4);
run;

/* Restricted Rotterdam Model */
/* Seasonal Dummies           */
proc model data = meat;
   parameters ab ab1 ab2 ab3 bb gbb gbp gbc gbt
              ap ap1 ap2 ap3 bp gpb gpp gpc gpt
              ac ac1 ac2 ac3 bc gcb gcp gcc gct;
   endogenous ds_beef ds_pork ds_chick;
   exogenous dexp_meat dlp_beef dlp_pork dlp_chick dlp_turk
             qtr1 qtr2 qtr3;
   restrict gbb + gbp + gbc + gbt = 0,
            gpb + gpp + gpc + gpt = 0,
            gcb + gcp + gcc + gct = 0,
            gbp = gpb, gbc = gcb, gpc = gcp;

   ds_beef  = ab + ab1 * qtr1 + ab2 * qtr2 + ab3 * qtr3
              + bb * dexp_meat + gbb * dlp_beef + gbp * dlp_pork
              + gbc * dlp_chick + gbt * dlp_turk;
```

```
  ds_pork  = ap + ap1 * qtr1 + ap2 * qtr2 + ap3 * qtr3
             + bp * dexp_meat + gpb * dlp_beef + gpp * dlp_pork
             + gpc * dlp_chick + gpt * dlp_turk;
  ds_chick = ac + ac1 * qtr1 + ac2 * qtr2 + ac3 * qtr3
             + bc * dexp_meat + gcb * dlp_beef + gcp * dlp_pork
             + gcc * dlp_chick + gct * dlp_turk;

  fit ds_beef ds_pork ds_chick / fiml outest = rott_seas;

run;
```

Figure 3.10 The Seasonal Rotterdam Model

The MODEL Procedure

Model Summary	
Model Variables	11
Endogenous	3
Exogenous	8
Parameters	27
Equations	3
Number of Statements	10

Figure 3.10 *continued*

Nonlinear FIML Parameter Estimates					
Parameter	Estimate	Approx Std Err	t Value	Approx Pr > |t|	Label
ab	0.169286	0.0448	3.77	0.0003	
ab1	0.075477	0.0102	7.42	<.0001	
ab2	-0.00239	0.00888	-0.27	0.7883	
ab3	0.006438	0.00864	0.74	0.4583	
bb	1.117039	0.0134	83.19	<.0001	
gbb	-0.13134	0.0213	-6.17	<.0001	
gbp	0.10478	0.0117	8.92	<.0001	
gbc	0.001124	0.0164	0.07	0.9454	
gbt	0.025434	0.00356	7.15	<.0001	
ap	-0.12003	0.0241	-4.99	<.0001	
ap1	-0.04588	0.00562	-8.16	<.0001	
ap2	-0.02565	0.00486	-5.28	<.0001	
ap3	-0.02701	0.00453	-5.96	<.0001	
bp	-0.07036	0.00707	-9.95	<.0001	
gpb	0.10478	0.0117	8.92	<.0001	
gpp	-0.15553	0.00743	-20.94	<.0001	
gpc	0.047534	0.00975	4.87	<.0001	
gpt	0.003216	0.00190	1.70	0.0931	
ac	0.059274	0.0349	1.70	0.0926	
ac1	-0.01395	0.00654	-2.13	0.0355	
ac2	0.040722	0.00598	6.81	<.0001	
ac3	0.031891	0.00594	5.37	<.0001	
bc	-0.09055	0.00916	-9.88	<.0001	
gcb	0.001124	0.0164	0.07	0.9454	
gcp	0.047534	0.00975	4.87	<.0001	
gcc	-0.0403	0.0166	-2.43	0.0171	
gct	-0.00836	0.00331	-2.52	0.0134	
Restrict0	5.76821	44.3806	0.13	0.8974	gbb + gbp + gbc + gbt = 0
Restrict1	-55.5392	61.4689	-0.90	0.3691	gpb + gpp + gpc + gpt = 0
Restrict2	298.7035	58.1941	5.13	<.0001	gcb + gcp + gcc + gct = 0
Restrict3	-155.121	78.1106	-1.99	0.0464	gbp = gpb
Restrict4	458.7251	83.4773	5.50	<.0001	gbc = gcb
Restrict5	541.0947	117.1	4.62	<.0001	gpc = gcp

Following its inception, the Rotterdam model has since been applied in a number of different situations and contexts. For instance, Duffy (1987) used a Rotterdam model with advertising to test whether advertising affected demand within industries. Because the incorporation of advertising into the model is fairly simple, it was used by Kinnucan et al. (1997) to investigate the effects of health information on U.S. meat demand. In a similar vein, Marsh, Schroeder, and Mintert (2004) examined the impact of meat recalls on consumer demand and found that elasticity estimates from an absolute price Rotterdam model indicated a shift to non-meat consumption after recalls. Selvanathan (1991) used the model to test whether consumption patterns for alcoholic beverages differed across countries. These studies are only a small subset of the many empirical situations to which the Rotterdam model has been applied. Given the flexibility of the model and its ease of use, it is sure to see application in the future.

3.4 Almost Ideal Demand System

The almost ideal demand system was developed and introduced in a seminal paper by Deaton and Muellbauer (1980a). Since that time, it has become one of the most widely used approaches to demand. The system has several favorable properties that make it "almost ideal". It gives an arbitrary first-order approximation to any demand system, aggregates perfectly over consumers, and satisfies the axioms of choice. As with the Rotterdam model, the AIDS is a system approach to demand where the implications of consumer theory can be tested. We will see that the derivation of the AIDS model begins from the expenditure (cost) function of the consumer and associated assumptions about the form of this function.

Indeed there are several similarities between the AIDS and Rotterdam models. Both models are locally flexible functional forms, possessing sufficient flexibility to approximate elasticities at any point. They are also both linear in parameters (in fact they have the same number of parameters) and thus similar in terms of difficulty of estimation. Because of this similarity a number of studies have tested the suitability of these models, with the aim of identifying a best approach for a given data set. Alson and Chalfant (1993) found that the Rotterdam model was not rejected in an application to meat demand, while the AIDS model was rejected. Later work by Barnett and Seck (2008) concluded that a best approach could not be specified. Like most empirical work, the suitability of any model depends on the application.

3.4.1 Full and Linear Approximate AIDS Models

Our derivation of the almost ideal demand system follows Deaton and Muellbauer (1980a). First consider the following expenditure function

$$\log\left(e(u, \mathbf{p})\right) = \ln a(\mathbf{p}) + u \ln b(\mathbf{p}) \tag{3.24}$$

This particular form is an expenditure function that adheres to what are commonly known as PIGLOG preferences or "price-independent, generalized logarithmic" preferences. These preferences have the useful property that they allow exact aggregation over consumers. Several other locally flexible functional forms have PIGLOG preferences, most notably the exactly aggregable translog model found in Christensen, Jorgenson, and Lau (1975). The two price functions are assumed to have the following forms

$$\ln a(\mathbf{p}) = \alpha_0 + \sum_k \alpha_k + \frac{1}{2} \sum_k \sum_j \gamma^*_{kj} \ln p_k \ln p_j \tag{3.25}$$

and

$$\ln b(\mathbf{p}) = \beta_0 \prod_k p_k^{\beta_k} \tag{3.26}$$

Now allow $\gamma_{ij} = \frac{1}{2}(\gamma^*_{ij} + \gamma^*_{ji})$. Applying Shepherd's lemma results in share equations of the form

$$s_i = \alpha_i + \sum_{j=1}^{n} \gamma_{ij} \log\left(p_j\right) + \beta_i \log\left(m/P\right) \tag{3.27}$$

where P is a price index. The price index takes the form

$$\log P = \alpha_0 + \sum_k \alpha_k \ln p_k + \frac{1}{2}\sum_k \sum_j \gamma_{kj} \ln p_k \ln p_j \tag{3.28}$$

The parameters of equation 3.27 have a simple interpretation. The various β_i determine whether the good is a luxury or necessity as they are measures of the response of the expenditure shares to changes in total expenditure. The parameters γ_{ij} measure the change in the budget share to proportional change in the price of good j. Because we started from an expenditure function, the theoretical properties of the expenditure function imply restrictions on the demand functions in equation 3.27. Linear homogeneity holds when $\sum_{k=1} \gamma_{ik} = 0$ for all i and k. Symmetry implies that $\gamma_{ik} = \gamma_{ki}$ for all i and k. Adding up requires that $\sum_{i=1} \alpha_i = 1$ and $\sum_{i=1} \beta_i = 0$.

Let δ_{ik} be the Kronecker delta, which equals 1 when $i = j$ and zero otherwise. The elasticities for the AIDS model are given by

$$\epsilon_{ik} = \frac{\gamma_{ik} - \beta_i [s_k - \beta_k \log(m/P)]}{s_i} - \delta_{ik} \tag{3.29}$$

$$\epsilon^h_{ik} = \frac{\gamma_{ik} - \beta_i [s_k - \beta_k \log(m/P)]}{s_i} - \delta_{ik} + s_j \left(\frac{\beta_i}{s_i} + 1\right) = \epsilon_{ij} + s_j \eta_i \tag{3.30}$$

$$\eta_i = \frac{\beta_i}{s_i} + 1 \tag{3.31}$$

The elasticity formulas make it more clear that the income elasticities are governed by β_i and the price elasticities are governed by γ_{ij}.

The AIDS model is clearly nonlinear, and while such nonlinearities do not present a problem for today's computing power, it was common to estimate a linear version of the AIDS model. This was achieved by modifying the price index, or approximating the index with a simpler form. One common approach is to use Stone's index where $\log(P) = \sum_i s_i \log p_i$. Then we have

$$s_i = \alpha_i^* + \sum_k \gamma_{ki} \log p_k + \beta_i \log(m/P) \tag{3.32}$$

In fact, if prices are exactly collinear and move proportionally to Stone's index, then the linear approximate model can be used to precisely estimate the parameters of the full AIDS model. In other cases, the relationship between the parameters of the linear approximate model and the AIDS model are not known as the two models are nonnested systems. It is not entirely clear whether the elasticities should be based on the linear approximate model or the full AIDS model. Green and Alston (1990) show that when using the LA/AIDS model, the same formulas for computing elasticities cannot be used. Using theoretically correct formulas for elasticities, they found estimates from AIDS and LA/AIDS models to be similar. Asche and Wessels (1997) later reconciled the AIDS and LA/AIDS models by demonstrating that, where all prices and income are normalized at unity, the two models have the same elasticity formulas.

There has also been debate as to the suitability of the properties of the LA/AIDS model with respect to Stone's price index. Moschini (1995) demonstrated that Stone's index is not invariant to units of measurement in prices, while Eales and Unnevehr (1988) noted that budget shares appear on both sides of the estimated equations when Stone's index is used. Other authors have examined the flexibility of the linear approximate model when constraints from theory are imposed. LaFrance (2004) found the form of the expenditure function in equation 3.24 to be greatly restricted when integrability conditions were imposed. Given the ease of estimating the full AIDS model, concerns with the LA/AIDS model now attract less attention. Nonetheless, the LA/AIDS can be useful as a means of obtaining starting values for the estimation of a full AIDS specification.

As concern over linear approximations to the AIDS decreased, attention has turned to the increasingly popular quadratic almost ideal demand system (QUAIDS). The QUAIDS was derived from the utility function by Banks, Blundell, and Lewbel (1997) and maintains the desirable properties of the AIDS model with increased flexiblity. As its name indicates, quadratic terms are added to the AIDS model to provide this increased flexibility. At the time the QUAIDS was introduced, computing power was already sufficient to allow for nonlinear estimation QUAIDS models of reasonable size. Nonetheless, Matsuda (2006) provides a linear approximation to the QUAIDS and argues that the approximation is particularly useful when working with nonstationary time series data. Applications of the QUAIDS include Moro and Sckokai (2000) who investigated household food consumption and Fisher, Fleissig, and Serletis (2001) who included the QUAIDS in a larger empirical comparison of demand systems.

3.4.2 Empirical Analysis

Our analysis of the almost ideal demand system is based on an annual meat consumption data set. Data is provided on per capita retail quantities of beef, veal, pork, poultry, and fish and seafood in pounds. Consumer price index measurements are available for beef and veal as a composite commodity, pork, poultry, and fish and seafood. Constant dollar per capita retail quantities of the four goods can be computed. Information on the overall consumer price index, U.S. population, and total consumption expenditures are provided. These can be used to derive per capita consumption expenditures on each of the items as well as total per capita consumption expenditure. The data are based on Christensen and Manser (1977).

Because we have time series data, the model that we estimate is a first difference AIDS model. Time series variables are unlikely to be stationary and must first be differenced to be rendered stationary. The model can then be estimated using the differenced variables. After reading in the data, we normalize prices for the meats and then construct differenced quantities and prices. The total expenditure on meat is simply the sum of expenditures on each of the different categories. Mentioned by Barnett and Seck (2008), the LA-AIDS model in first differences has the same dependent variables as an absolute price Rotterdam model.

```
data meat_annual;
set meat_annual;

   bfvlp = (bfvlp / 119.5287457) / 1.0415710;
   porkp = (porkp / 118.1589017) / 1.0278743;
   poultp = (poultp / 124.0272949) / 1.0314877;
   fishp = (fishp / 131.2487286) / 1.0411223;

   lbfvlq = log(bfvlq);
   lporkq = log(porkq);
   lpoultq = log(poultq);
   lfishq = log(fishq);

   lbfvlp = log(bfvlp);
   lporkp = log(porkp);
   lpoultp = log(poultp);
   lfishp = log(fishp);

   dlbfvlq = dif(lbfvlq);
   dlporkq = dif(lporkq);
```

```
      dlpoultq = dif(lpoultq);
      dlfishq = dif(lfishq);

      dlbfvlp = dif(lbfvlp);
      dlporkp = dif(lporkp);
      dlpoultp = dif(lpoultp);
      dlfishp = dif(lfishp);

      xmeat = xbfvl + xpork + xpoult + xfish;

      wbfvl = xbfvl / xmeat;
      wpork = xpork / xmeat;
      wpoult = xpoult / xmeat;
      wfish = xfish / xmeat;

     dwbfvl = dif(wbfvl);
     dwpork = dif(wpork);
     dwpoult = dif(wpoult);
     dwfish = dif(wfish);

     xmeat = (xmeat / 464.7737898)/ 1.0327489;
     dlxmeat = dif(log(xmeat));

   run;
```

PROC MEANS can be used to construct the mean shares of the four types of meat. The mean shares are first output to the data set MeanShares. The Stone's price index is then constructed by multiplying the mean shares by their differenced log prices and summing.

```
   proc means data = meat_annual noprint;
      variables wbfvl wpork wpoult wfish;
      output out = meanshares mean = wbfvl0 wpork0 wpoult0 wfish0;
   run;

   data meat_annual;
      if _N_ = 1 then set meanshares(drop = _TYPE_  _FREQ_) ;
   set meat_annual;
      dlp = wbfvl0 * dlbfvlp + wpork0 * dlporkp + wpoult0 * dlpoultp
            + wfish0 * dlfishp;
   run;
```

The unrestricted linear approximate AIDS model is estimated first. Like the Rotterdam model, one equation is dropped in estimation of the system. In this case, we have chosen to omit the equation for fish. The model thus has 18 free parameters. In the FIT statement, we have instructed SAS to estimate the system using full information maximum likelihood. We can also test the homogeneity, symmetry, and joint restrictions using likelihood ratio tests. The syntax for PROC MODEL remains the same whether estimating an AIDS or Rotterdam model.

```
   /* LA/AIDS Model Unrestricted */
   proc model data = meat_annual;
      parameters ab bb gbb gbp gbo gbf
                 ap bp gpb gpp gpo gpf
                 ao bo gob gop goo gof;
      endogenous dwbfvl dwpork dwpoult;
      exogenous dlxmeat dlbfvlp dlporkp dlpoultp dlfishp;

      dwbfvl  = ab + bb * (dlxmeat - dlp) + gbb * dlbfvlp + gbp * dlporkp
```

```
             + gbo * dlpoultp + gbf * dlfishp;
   dwpork  = ap + bp * (dlxmeat - dlp) + gpb * dlbfvlp + gpp * dlporkp
             + gpo * dlpoultp + gpf * dlfishp;
   dwpoult = ao + bo * (dlxmeat - dlp) + gob * dlbfvlp + gop * dlporkp
             + goo * dlpoultp + gof * dlfishp;

   fit dwbfvl dwpork dwpoult / fiml outest = la_aids_unrest;

   test "Homogeneity"
      gbb + gbp + gbo + gbf = 0,
      gpb + gpp + gpo + gpf = 0,
      gob + gop + goo + gof = 0, / lr;

   test "Symmetry"
      gbp = gpb,
      gbo = gob,
      gpo = gop, / lr;

   test "Joint Homogeneity and Symmetry"
      gbb + gbp + gbo + gbf = 0,
      gpb + gpp + gpo + gpf = 0,
      gob + gop + goo + gof = 0,
      gbp = gpb, gbo = gob, gpo = gop, / lr;
run;
```

Figure 3.11 The Unrestricted Linear Approximate AIDS Model

The MODEL Procedure

Model Summary	
Model Variables	8
Endogenous	3
Exogenous	5
Parameters	18
Equations	3
Number of Statements	18

Figure 3.11 *continued*

Nonlinear FIML Parameter Estimates				
Parameter	**Estimate**	**Approx Std Err**	**t Value**	**Approx Pr > |t|**
ab	-0.00229	0.00361	-0.63	0.5300
bb	0.052324	0.0985	0.53	0.5986
gbb	0.077731	0.0265	2.93	0.0060
gbp	0.010424	0.0271	0.38	0.7028
gbo	-0.06242	0.0369	-1.69	0.0998
gbf	-0.03951	0.0379	-1.04	0.3040
ap	-0.00167	0.00227	-0.73	0.4683
bp	-0.00905	0.0917	-0.10	0.9220
gpb	0.008317	0.0225	0.37	0.7145
gpp	0.013281	0.0199	0.67	0.5094
gpo	-0.03133	0.0218	-1.44	0.1591
gpf	0.008097	0.0380	0.21	0.8325
ao	0.003418	0.00140	2.44	0.0200
bo	-0.02573	0.0371	-0.69	0.4927
gob	-0.03466	0.0234	-1.48	0.1483
gop	-0.01774	0.0207	-0.86	0.3972
goo	0.095642	0.0237	4.04	0.0003
gof	-0.04342	0.0259	-1.67	0.1031

Test Results				
Test	**Type**	**Statistic**	**Pr > ChiSq**	**Label**
Homogeneity	L.R.	0.66	0.8821	gbb + gbp + gbo + gbf = 0, gpb + gpp + gpo + gpf = 0, gob + gop + goo + gof = 0
Symmetry	L.R.	3.44	0.3293	gbp = gpb, gbo = gob, gpo = gop
Joint Homogeneity and Symmetry	L.R.	4.60	0.5964	gbb + gbp + gbo + gbf = 0, gpb + gpp + gpo + gpf = 0, gob + gop + goo + gof = 0, gbp = gpb, gbo = gob, gpo = gop

In this case, none of the restrictions implied by theory are rejected but few of the model parameters are significant. As with the Rotterdam model, it's easy to impose the linear restrictions in PROC MODEL. Calculation of the elasticities is omitted for the unrestricted LA/AIDS so we move ahead to the restricted version. The number of free parameters falls from 18 to 12 with six restrictions.

```
/* LA/AIDS Model Restricted*/
proc model data = meat_annual;
   parameters ab bb gbb gbp gbo gbf
              ap bp gpb gpp gpo gpf
              ao bo gob gop goo gof;
   endogenous dwbfvl dwpork dwpoult;
   exogenous dlxmeat dlbfvlp dlporkp dlpoultp dlfishp;
   restrict gbb + gbp + gbo + gbf = 0,
            gpb + gpp + gpo + gpf = 0,
            gob + gop + goo + gof = 0,
            gbp = gpb, gbo = gob, gpo = gop;

   dwbfvl  = ab + gbb * dlbfvlp + gbp * dlporkp + gbo * dlpoultp
             + gbf * dlfishp + bb * (dlxmeat - dlp);
   dwpork  = ap + gpb * dlbfvlp + gpp * dlporkp + gpo * dlpoultp
             + gpf * dlfishp + bp * (dlxmeat - dlp);
   dwpoult = ao + gob * dlbfvlp + gop * dlporkp + goo * dlpoultp
```

```
            + gof * dlfishp + bo * (dlxmeat - dlp);

fit dwbfvl dwpork dwpoult/fiml outest = la_aids_rest;
run;
```

Figure 3.12 The Restricted Linear Approximate AIDS Model

The MODEL Procedure

Model Summary	
Model Variables	8
Endogenous	3
Exogenous	5
Parameters	18
Equations	3
Number of Statements	10

Nonlinear FIML Parameter Estimates

Parameter	Estimate	Approx Std Err	t Value	Approx Pr > |t|	Label
ab	-0.0025	0.00138	-1.81	0.0782	
bb	0.091543	0.0500	1.83	0.0753	
gbb	0.0809	0.0251	3.22	0.0027	
gbp	0.010363	0.0129	0.80	0.4282	
gbo	-0.04018	0.0183	-2.19	0.0350	
gbf	-0.05109	0.0167	-3.06	0.0042	
ap	-0.00152	0.000986	-1.54	0.1328	
bp	0.009095	0.0560	0.16	0.8719	
gpb	0.010363	0.0129	0.80	0.4282	
gpp	0.013694	0.0174	0.78	0.4377	
gpo	-0.0215	0.0121	-1.78	0.0840	
gpf	-0.00256	0.0108	-0.24	0.8132	
ao	0.003083	0.000704	4.38	<.0001	
bo	-0.05254	0.0303	-1.73	0.0919	
gob	-0.04018	0.0183	-2.19	0.0350	
gop	-0.0215	0.0121	-1.78	0.0840	
goo	0.086335	0.0188	4.59	<.0001	
gof	-0.02466	0.0108	-2.29	0.0281	
Restrict0	24.75419	42.5354	0.58	0.5679	gbb + gbp + gbo + gbf = 0
Restrict1	8.492706	42.4846	0.20	0.8448	gpb + gpp + gpo + gpf = 0
Restrict2	81.89187	48.9266	1.67	0.0945	gob + gop + goo + gof = 0
Restrict3	-21.4413	38.1585	-0.56	0.5814	gbp = gpb
Restrict4	89.59009	45.4094	1.97	0.0469	gbo = gob
Restrict5	115.4098	53.8786	2.14	0.0301	gpo = gop

Under the restricted LA/AIDS the majority of the coefficient estimates are now significant at the 10% level. The procedure for calculating the elasticities is similar to the Rotterdam model examples and involves the use of PROC IML. The missing parameters of the demand equation for fish are calculated from the estimated parameters. The mean shares for the meats were used in constructing Stone's price index and are used for evaluation of the elasticities. As before, a final print statement gives the elasticities in matrix form.

```
proc iml;
   use la_aids_rest;
   read all var {gbb gbp gbo gbf gpp gpo gpf goo gof bb bp bo};
   close la_aids_rest;

   gff = 0 - gbf - gpf - gof;
   bf = 0 - bb - bp - bo;

   use meanshares;
         read all var {wbfvl0} into wb;
         read all var {wpork0} into wp;
         read all var {wpoult0} into wo;
         read all var {wfish0} into wf;
   close meanshares;

   w = wb//wp//wo//wf;
   b = bb//bp//bo//bf;

   print w;

   gij = (gbb||gbp||gbo||gbf)//
         (gbp||gpp||gpo||gpf)//
         (gbo||gpo||goo||gof)//
         (gbf||gpf||gof||gff);

   print gij;

   nk = ncol(gij);
   mi = -1#I(nk);
   uep = j(nk, nk, 0);
   cep = j(nk, nk, 0);
   exe = j(nk, 1, 0);

   do i=1 to nk;
   do j=1 to nk;
      uep[i,j] = ((gij[i,j]-b[i]#w[j])/w[i]) + mi[i,j];
      cep[i,j] = ((gij[i,j] - b[i] # w[j])/w[i]) + mi[i,j]
                 + w[j] # (1 + (b[i] / w[i]));
      exe[i]   = 1 + b[i] / w[i];
   end;
   end;

   factors = {"Beef/Veal" "Pork" "Poultry" "Fish"};

   print
      uep[label="Uncompensated Price Elasticities of Demand"
          rowname=factors colname=factors format=d7.3],
      cep[label="Compensated Price Elasticities of Demand"
          rowname=factors colname=factors format=d7.3],
      exe[label="Expenditure Elasticities" rowname=factors
          format=d7.3];
quit;
```

Figure 3.13 Elasticity Matrices

Uncompensated Price Elasticities of Demand				
	Beef/Veal	Pork	Poultry	Fish
Beef/Veal	-0.918	-0.0253	-0.120	-0.132
Pork	0.0252	-0.953	-0.0952	-0.0149
Poultry	-0.0898	-0.0503	-0.452	-0.106
Fish	-0.246	0.0782	-0.140	-0.279

Compensated Price Elasticities of Demand				
	Beef/Veal	Pork	Poultry	Fish
Beef/Veal	-0.360	0.264	0.0882	0.00700
Pork	0.510	-0.701	0.0855	0.106
Poultry	0.236	0.119	-0.330	-0.0252
Fish	0.0281	0.220	-0.0378	-0.211

Expenditure Elasticities	
Beef/Veal	1.196
Pork	1.038
Poultry	0.698
Fish	0.587

Interpretation of the elasticities is left to the reader. Our next task is to estimate the full AIDS model and, in any event, we will compare the full model estimates with those of the linear approximate. The AIDS model includes the price index of equation 3.28 and additional restrictions. These statements are added in PROC MODEL and the equation for fish is again omitted. The model has 22 total parameters, but only 15 free parameters after accounting for the seven restrictions. The parameter estimates are suppressed and we move immediately to the elasticity estimates of Figure 3.14.

```
/*AIDS Model Restricted*/
proc model data = meat_annual;
   parameters cb ab bb gbb gbp gbo gbf
              cp ap bp gpb gpp gpo gpf
              co ao bo gob gop goo gof
              cf;
   endogenous dwbfvl dwpork dwpoult;
   exogenous dlxmeat dlbfvlp dlporkp dlpoultp dlfishp;
   restrict gbb + gbp + gbo + gbf = 0,
            gpb + gpp + gpo + gpf = 0,
            gob + gop + goo + gof = 0,
            gbp = gpb, gbo = gob, gpo = gop,
            cb + cp + co + cf = 1;

   dlpindex = cb * dlbfvlp + cp * dlporkp + co * dlpoultp + cf * dlfishp
              + 0.5 * (gbb * dif(lbfvlp * lbfvlp)
              + 2* gbp * dif(lbfvlp * lporkp)
              + 2 * gbo * dif(lbfvlp * lpoultp)
              + 2 * gbf * dif(lbfvlp * lfishp)
              + gpp * dif(lporkp * lporkp)
              + 2 * gpo * dif(lporkp * lpoultp)
              + 2 * gpf * dif(lporkp * lfishp)
              + goo * dif(lpoultp * lpoultp)
              + 2 * gof * dif(lpoultp * lfishp)
```

```
              - (gbf + gpf + gof) * dif(lfishp * lfishp));

   dwbfvl  = ab + bb * (dlxmeat - dlpindex) + gbb * dlbfvlp
             + gbp * dlporkp + gbo * dlpoultp + gbf * dlfishp;
   dwpork  = ap + bp * (dlxmeat - dlpindex) + gpb * dlbfvlp
             + gpp * dlporkp + gpo * dlpoultp + gpf * dlfishp;
   dwpoult = ao + bo * (dlxmeat - dlpindex) + gob * dlbfvlp
             + gop * dlporkp + goo * dlpoultp + gof * dlfishp;

   fit dwbfvl dwpork dwpoult / fiml outest = aids_rest;
run;

proc iml;
   use aids_rest;
   read all var {gbb gbp gbo gbf gpp gpo gpf goo gof bb bp bo cb cp co cf};
   close aids_rest;

   gff = 0 - gbf - gpf - gof;
   bf = 0 - bb - bp - bo;

   use meanshares;
         read all var {wbfvl0} into wb;
         read all var {wpork0} into wp;
         read all var {wpoult0} into wo;
         read all var {wfish0} into wf;
   close meanshares;

   w = wb//wp//wo//wf;
   b = bb//bp//bo//bf;
   c = cb//cp//co//cf;

   print w;

   gij = (gbb||gbp||gbo||gbf)//
         (gbp||gpp||gpo||gpf)//
         (gbo||gpo||goo||gof)//
         (gbf||gpf||gof||gff);

   print gij;

   nk = ncol(gij);
   mi = -1#I(nk);
   uep = j(nk, nk, 0);
   cep = j(nk, nk, 0);
   exe = j(nk, 1, 0);

   do i=1 to nk;
   do j=1 to nk;
      uep[i,j] = ((gij[i,j]-b[i]#c[j])/w[i])+ mi[i,j];
      cep[i,j] = ((gij[i,j]-b[i]#c[j])/w[i])+ mi[i,j] +w[i]#(1+(b[i]/w[i]));
      exe[i] = 1 + b[i] / w[i];
   end;
   end;

   factors = {"Beef/Veal" "Pork" "Poultry" "Fish"};
```

```
      print
         uep[label = "Uncompensated Price Elasticities of Demand"
             rowname = factors colname = factors format = d7.3],
         cep[label = "Compensated Price Elasticities of Demand"
             rowname = factors colname = factors format = d7.3],
         exe[label = "Expenditure Elasticities" rowname = factors
             format = d7.3];
   quit;
```

Figure 3.14 Elasticity Matrices

Uncompensated Price Elasticities of Demand

	Beef/Veal	Pork	Poultry	Fish
Beef/Veal	-0.822	0.0228	-0.117	-0.110
Pork	0.0273	-0.951	-0.0878	0.00389
Poultry	-0.184	-0.0975	-0.400	-0.335
Fish	-0.495	-0.0482	-0.245	-0.0652

Compensated Price Elasticities of Demand

	Beef/Veal	Pork	Poultry	Fish
Beef/Veal	-0.343	0.502	0.362	0.370
Pork	0.271	-0.707	0.156	0.248
Poultry	-0.0070	0.0797	-0.223	-0.158
Fish	-0.395	0.0510	-0.146	0.0340

Expenditure Elasticities	
Beef/Veal	1.026
Pork	1.007
Poultry	1.018
Fish	0.853

The income elasticities for beef, pork, and poultry are all very close to one so a one percent increase in the consumer's income leads to a one percent increase in the quantity demanded of the respective meat. Given that the data used in the analysis are aggregate time series data, these income elasticities seem reasonable. Beef is a substitute for pork, but a complement for poultry and fish. One concern is that the compensated own-price elasticity for fish is positive, though close to zero. It is important to remember that the elasticities are evaluated at a point in the sample and would vary if we were to evaluate them at different shares.

3.5 Demand for Differentiated Products

In the previous sections of this chapter, we have seen examples of demand estimation for homogeneous products or aggregates. When we estimate demand functions for homogeneous goods, a product is usually treated as a single fully integrated entity. Our attention now turns to the estimation of demand for differentiated products. Within the differentiated products framework, the most commonly used approach is often termed the "characteristics space" approach, where a product can be decomposed into several characteristics. In the following sections, we will introduce various estimation methods within the "characteristics space" framework.

Most studies in the characteristics space framework assume that each consumer purchases only one unit of one product on each shopping occasion. For example, a consumer purchases one cell phone every two years. A household with several family members buys one box of cereal per week. A businessman purchases a new suit every six months.

Discrete choice models are appropriate to model these shopping events and types of purchasing behavior. As with our descriptions of demand for homogenous goods, we start with important properties of the theoretical model. Then we define choice probabilities and derive the demand model that is applied in practice.

3.5.1 Discrete Choice

Discrete choice models describe decision makers' choices among alternatives in shopping events. The decision makers could be individual consumers, households, firms, or any other decision making units. The alternatives could be competing products, courses of action, or any other options. The choice set, which is the set of alternatives from which the consumer can select, must satisfy the following three properties.

1. **Mutually exclusive**: the consumer only purchases one unit of one product from the choice set
2. **Exhaustive**: all alternatives are included in the choice set
3. **Finite**: The number of alternatives must be finite

In some cases a consumer may purchase multiple products. Suppose there are three types of mobile phones in the choice set: Android, Apple, and BlackBerry. Some consumers may purchase more than one type of phone in a shopping trip. Instead of defining the choice set to have three products, we can define the alternatives to be Android, Apple, BlackBerry, Android and Apple, Android and BlackBerry, Apple and BlackBerry, and all three products. By re-defining the choice set, the choice set satisfies all three criteria. Theoretically speaking, this method can then be applied to any number of products. However, as the number of products increases, the number of alternatives increases, and so do the practical difficulties of estimating such a model. In the case where a consumer purchases no products, we can define no purchase as an alternative in the choice set. This allows us to answer the following question, what factors determine a consumers' decision between making no purchases and making a purchase?

Discrete choice models are derived under the assumption of utility maximization. We start with the indirect utility function of consumer n, which gives the utility the consumer obtains from product j:

$$U_{nj} = V_{nj} + \epsilon_{nj} \quad j = 1, \ldots, J, n = 1, \ldots, N \tag{3.33}$$

where the utility, U_{nj}, is observed by the decision maker. U_{nj} is decomposed into two parts: V_{nj} and ϵ_{nj}. V_{nj} is revealed to the researcher. ϵ_{nj} is observed to the decision maker, but not the researcher. We assume ϵ_{nj} is a random variable. The probability that consumer n chooses alternative i is then given by

$$\begin{aligned} P_{ni} &= \text{Prob}\left(U_{ni} > U_{nj}, \forall j \neq i\right) \\ &= \text{Prob}\left(\epsilon_{nj} - \epsilon_{ni} > V_{ni} - V_{nj}, \forall j \neq i\right) \\ &= \int I\left(\epsilon_{nj} - \epsilon_{ni} > V_{ni} - V_{nj}, \forall j \neq i\right) f(\epsilon_n) d\epsilon_n, \end{aligned} \tag{3.34}$$

where $f(\cdot)$ is the joint density of the unobserved portion of utility, $\epsilon_{n.}$. Different discrete choice models are obtained by varying the specification for $f(\cdot)$. The total market demand for alternative i can be written as

$$Q_i = \sum_{n=1}^{N} P_{ni}$$

A key concept in econometric analysis is identification: whether we have enough information in the data to estimate the parameters of interest. When we specify a discrete choice model, we need to keep in mind that only differences in utility matter. That is to say, the absolute value of utility is meaningless. The ranking of utilities obtained from different options will not be changed if a constant is added to utilities from these alternatives. Below are three different types of variables that may be present in the data. For each case, problems of identification are considered.

3.5.1.1 Case I: Product Specific Constant Term

In Case I, we assume a very simple model where there are only two products in the choice set. The representative utility has a simple form

$$U_{u1} = C_1 + \epsilon_{n1}$$
$$U_{u2} = C_2 + \epsilon_{n2}$$

where C_1 and C_2 are the product specific constants for the two options. The constant terms do not vary across consumers. We can write the difference between U_{n1} and U_{n2} in the following form:

$$U_{u1} - U_{u2} = C_1 - C_2 + \epsilon_{n1} - \epsilon_{n2}$$

The average of $\epsilon_{n1} - \epsilon_{n2}$ is zero. The average difference in two utility levels identifies the difference between two constant terms, C_1 and C_2. Since the absolute value of utility does not matter, we cannot separately identify these two constant terms. If we normalize $C_1 = 0$, the demand model then becomes

$$U_{u1} = \epsilon_{n1}$$
$$U_{u2} = C_2 + \epsilon_{n2}$$

C_2 should be interpreted as the impact of all factors of alternative 2 that do not vary across decision makers on utility relative to that of all factors of alternative 1 that do not vary across decision makers. In general, if there are J alternatives in the choice set, we can only identify $J - 1$ product specific constants.

3.5.1.2 Case II: Demographic Variables

The demographic variables of decision makers can have an important impact on choices. For example, consumers with high income are more likely to purchase organic food than those with low income. Similar to Case I, we assume there are only two alternatives in the choice set. The utilities obtained from the two options for decision maker n are

$$U_{u1} = x_n\beta_1 + \epsilon_{n1}$$
$$U_{u2} = x_n\beta_2 + \epsilon_{n2}$$

where x_n captures all the demographic variables, such as income, education level, gender, and so on. These demographic variables do not vary across alternatives. However, their impacts can differ across alternatives. The difference in utility is

$$U_{u1} - U_{u2} = x_n(\beta_1 - \beta_2) + \epsilon_{n1} - \epsilon_{n2}$$

Since only the differences in utility matter, only the difference between β_1 and β_2 can be identified. However, β_1 and β_2 cannot be separately identified. Usually we normalize $\beta_1 = 0$. The interpretation of β_2 is the impact of x_n on utility of alternative 2 relative to the impact of x_n on utility of alternative 1.

If the demographic variables are believed to have the same impacts on utilities of the two alternatives, the model is written as

$$U_{u1} = x_n\beta + \epsilon_{n1}$$
$$U_{u2} = x_n\beta + \epsilon_{n2}$$

and the difference in utility is

$$U_{u1} - U_{u2} = \epsilon_{n1} - \epsilon_{n2}$$

In this case, no parameter can be identified. Therefore, only the demographic variables that are assumed to have different impacts on utilities of alternatives can enter the discrete choice model.

3.5.1.3 Case III: Decision Maker and Product Specific Variables

Some variables vary across both alternatives and decision makers. For example, an individual may choose to go to a park near his home. The distance variable then varies across both individuals and alternatives.

Assuming there are two alternatives in the choice set, utility is written as

$$U_{u1} = x_{n1}\beta_1 + \epsilon_{n1}$$
$$U_{u2} = x_{n2}\beta_2 + \epsilon_{n2}$$

where $x_{n\cdot}$ varies across choices. The difference in utility is

$$U_{u1} - U_{u2} = x_{n1}\beta_1 - x_{n2}\beta_2 + \epsilon_{n1} - \epsilon_{n2}$$

In this case, both β_1 and β_2 can be separately identified. β_1 can be interpreted as the impact of x_n on utility of alternative 1 and β_2 is interpreted as the impact of x_n on utility of alternative 2.

3.5.2 Logit Models

Different specifications of discrete choice models are generated from different distributions of the error term. These differences can be important and are discussed in more detail in Greene (2018). If we assume the error term, ϵ_{nj} is independently and identically distributed and follows a type-I extreme value distribution (also called the Gumbel distribution), the discrete choice model is a multinomial logit model. The density of the distribution is

$$f(\epsilon_{nj}) = e^{-\epsilon_{nj}} e^{-e^{-\epsilon_{nj}}}, \tag{3.35}$$

and its cumulative distribution function is

$$F(\epsilon_{nj}) = e^{-e^{-\epsilon_{nj}}} \tag{3.36}$$

Using the Gumbel form for the distribution of the error terms, the probability in equation 3.34 can then be calculated. The probability for consumer n to choose alternative i is

$$\begin{aligned}
P_{ni} &= \int_{-\infty}^{+\infty} I\left(\epsilon_{nj} - \epsilon_{ni} > V_{ni} - V_{nj}, \forall j \neq i\right) f(\epsilon_n) d\epsilon_n \\
&= \int_{-\infty}^{+\infty} I\left(\epsilon_{nj} > V_{ni} - V_{nj} + \epsilon_{ni}, \forall j \neq i\right) f(\epsilon_n) d\epsilon_n \\
&= \int_{-\infty}^{+\infty} \prod_{j \neq i} \int_{-\infty}^{+\infty} I\left(\epsilon_{nj} > V_{ni} - V_{nj} + \epsilon_{ni}\right) f(\epsilon_{nj}) d\epsilon_{nj} f(\epsilon_{ni}) d\epsilon_{ni} \\
&= \int_{-\infty}^{+\infty} \prod_{j \neq i} F\left(V_{ni} - V_{nj} + \epsilon_{ni}\right) f(\epsilon_{ni}) d\epsilon_{ni} \\
&= \int_{-\infty}^{+\infty} \prod_{j \neq i} e^{-e^{-\left(V_{ni} - V_{nj} + \epsilon_{ni}\right)}} e^{-\epsilon_{ni}} d\epsilon_{ni}
\end{aligned} \tag{3.37}$$

With some algebraic manipulation, the logit probability becomes

$$P_{ni} = \frac{e^{V^{ni}}}{\sum_j e^{V_{nj}}} \tag{3.38}$$

You may notice that we have not specified a functional form for representative utility, V_{nj}. Representative utility is typically given a linear specification, that is, $V_{nj} = x'_{nj}\beta$, where x_{nj} is a vector of characteristics variables of alternative j. The multinomial logit formula has desirable properties. First, it guarantees the value of choice probability, P_{ni} is between 0 and 1. Second, the sum of choice probabilities for all alternatives, $\sum^J P_{j=1} P_{nj}$, is equal to 1. This is consistent with the exhaustiveness property of discrete choice models.

3.5.2.1 Elasticities

One important purpose of demand estimation is to calculate price elasticities. As explained in chapter 2, the own price elasticity measures the percentage change in quantity demanded for a product when its own price changes by one percent. The cross price elasticity measures the responsiveness of the quantity demanded for a good to a change in the price of another good, all else being equal. Assume that the representative utility for individual n obtaining from product i is $V_{ni} = \alpha p_{ni} + x'_{ni}\beta$, where α is the coefficient for the price variable, and β is a vector of coefficients for all the product characteristics other than price, x_{ni}. From the previous section, we know the quantity demanded of product i for consumer n can be written as

$$q_{ni} = 1 * P_{ni} = \frac{e^{V_{ni}}}{\sum_j e^{V_{nj}}} = \frac{e^{\alpha p_{ni} + x'_{ni}\beta}}{\sum_j e^{\alpha p_{nj} + x'_{nj}\beta}}$$

where P_{ni} is the probability for consumer n to choose product i. The own price elasticity of product i for consumer n can then be calculated as

$$e_{n,ii} = \frac{p_i \partial qni}{q_{ni} \partial p_i} = \alpha p_i (1 - P_{ni}) \tag{3.39}$$

Since the price coefficient α is negative, the own price elasticity is negative. This is consistent with the law of demand.

The cross price elasticity of product i can be calculated as

$$e_{n,ij} = \frac{p_j \partial qni}{q_{ni} \partial p_j} = -\alpha p_j P_{nj} \tag{3.40}$$

Since α is negative, cross price elasticity is positive. It is interesting to point out that i does not enter this formula; for a one percent decrease in the probability of product j, the probabilities of all other alternatives increase by one percent. This property is by construction of the logit model.

3.5.2.2 Consumer Welfare

Researchers are often interested in measuring how consumer welfare responds to a change in policies. For example, we expect consumer welfare to increase when a more energy efficient vehicle enters the car market. When the market becomes more competitive, car producers may decrease prices to attract more customers. Or car producers may invest more resources into research and development (R&D) and try to adapt to new technology and provide better quality. Another example is mergers and acquisitions. The merger of two cable companies may bring synergy and cut costs. However, the merger could significantly reduce market competition if both firms are big players in the market and have high market shares. Economists and policy makers are often interested in predicting how a merger will affect consumer welfare.

Consumer welfare is the area below the demand curve and above the market price. After we estimate expected demand using the logit model, we can calculate the consumer welfare. The consumer chooses the product that yields the highest utility. Consumer welfare is $CS_n = (1/\alpha_n)\max_i(U_{ni})$, where α_n is the marginal utility of income of consumer n. The division of α allows us to convert welfare into dollars. The monetary term that shows up in the utility is $\alpha_n(I_n - p_i)$, where I_n is the income of consumer n and drops out of the specification because it is a demographic variable that does not vary across products.

Since we observe the representative utility, V_{ni} but not U_{ni}, we can calculate the expected consumer welfare in the following way:

$$E(CS_n) = \frac{1}{\alpha_n} E\left[\max_i(V_{ni} + \epsilon_{ni})\right] \tag{3.41}$$

where the expectation is with respect to the error term ϵ_{ni}. When ϵ_{ni} is i.i.d. type-I extreme value distributed and α_n does not vary with respect to income, Small and Rosen (1981) show that the expected consumer welfare can be written as

$$E(CS_n) = \frac{1}{\alpha_n} ln\left(\sum_i e^{V_{ni}}\right) + C \tag{3.42}$$

where C is an unknown constant. Therefore, we don't know the exact value of $E(CS_n)$. Because utility and welfare are ordinal measures, we are not interested in knowing their absolute values. We are normally interested in the change of welfare due to a change in policy or the market structure. The change in consumer welfare can be calculated as

$$E(CS_n^1) - E(CS_n^0) = \frac{1}{\alpha_n}\left[ln\left(\sum_j^{J^1} e^{V_{nj}^1}\right) - ln\left(\sum_j^{J^0} e^{V_{nj}^0}\right)\right] \tag{3.43}$$

where 0 stands for the old policy environment and 1 stands for the new policy environment. It is then possible to conduct counterfactual analysis.

3.5.3 Empirical Analysis

Using the discrete choice models developed in the previous section, we can model individual shopping choices. The purpose of the following empirical applications is to use simple model and data analysis commands in SAS to estimate discrete choice models. We do not cover data cleaning, assumption verification, and other research processes relevant to the research enterprise. The applied researcher would, however, be required to complete these steps on their own.

3.5.3.1 An Example with Individual Data

In this example, an individual has three choices of where to shop: a department store, a luxury department store, and an outlet mall. Individuals' choices may be influenced by their income status and age. In reality, there are other factors that affect individuals' decisions, but for simplicity, we only consider these two predictors: age and income.

We start with model specification and identification. The utility of individual i choosing each option can be written as

$$\begin{aligned}
U_{i,\text{dept}} &= C_{\text{dept}} + \alpha_{\text{dept}} age_i + \beta_{dept}^2(inc_i = 2) + \beta_{\text{dept}}^3(inc_i = 3) + \epsilon_{i,\text{dept}} \\
U_{i,\text{lux}} &= C_{\text{lux}} + \alpha_{\text{lux}} age_i + \beta_{\text{lux}}^2(inc_i = 2) + \beta_{\text{lux}}^3(inc_i = 3) + \epsilon_{i,\text{lux}} \\
U_{i,\text{out}} &= C_{\text{out}} + \alpha_{\text{out}} age_i + \beta_{\text{out}}^2(inc_i = 2) + \beta_{\text{out}}^3(inc_i = 3) + \epsilon_{i,\text{out}}
\end{aligned} \tag{3.44}$$

where $U_{i,\text{dept}}$ is the utility of individual i choosing a department store as a shopping location. Note that the demographic variables, age and income status, do not vary across different choices. age_i is the age of individual i. $\epsilon_{i,\text{dept}}$ is the error term of individual i choosing a department store and is type I extreme value distributed. inc_i is a dummy variable with three different values, 1, 2, and 3, with 1 being the lowest and 3 being the highest income status. Because this dummy variable has three values, it can be defined with two variables. Here we choose $inc_i = 1$ as our reference, and write it as a row vector [0 0]. $inc_i = 2$ and $inc_i = 3$ can be written as [1 0] and [0 1], respectively.

Before we run the regression, it is important to understand which parameters can actually be identified in our model. Since the scale of utility does not matter, we have to choose one option to be our base when we use the logistic regression. In this example the luxury department store is chosen as the reference. The system of equations can be written as

$$\begin{aligned}
U_{i,\text{dept}}^* &= C_{\text{dept}}^* + \alpha_{\text{dept}}^* age_i + \beta_{\text{dept}}^{*2}(inc_i = 2) + \beta_{\text{dept}}^{*3}(inc_i = 3) + \epsilon_{i,\text{dept}}^* \\
U_{i,\text{out}}^* &= C_{\text{out}}^* + \alpha_{\text{out}}^* age_i + \beta_{\text{out}}^{*2}(inc_i = 2) + \beta_{\text{out}}^{*3}(inc_i = 3) + \epsilon_{i,\text{out}}^*
\end{aligned} \tag{3.45}$$

where

$$
\begin{aligned}
U^*_{i,\text{dept}} &= U_{i,\text{dept}} - U_{i,\text{lux}} \\
U^*_{i,\text{out}} &= U_{i,\text{out}} - U_{i,\text{lux}} \\
C^*_{\text{dept}} &= C_{\text{dept}} - C_{\text{lux}} \\
C^*_{\text{out}} &= C_{\text{out}} - C_{\text{lux}} \\
\alpha^*_{\text{dept}} &= \alpha_{\text{dept}} - \alpha_{\text{lux}} \\
\alpha^*_{\text{out}} &= \alpha_{\text{out}} - \alpha_{\text{lux}} \\
\beta^{*2}_{\text{dept}} &= \beta^2_{\text{dept}} - \beta^2_{\text{lux}} \\
\beta^{*2}_{\text{out}} &= \beta^2_{\text{out}} - \beta^2_{\text{lux}} \\
\beta^{*3}_{\text{dept}} &= \beta^3_{\text{dept}} - \beta^3_{\text{lux}} \\
\beta^{*3}_{\text{out}} &= \beta^3_{\text{out}} - \beta^3_{\text{lux}} \\
\epsilon^*_{i,\text{dept}} &= \epsilon_{i,\text{dept}} - \epsilon_{i,\text{lux}} \\
\epsilon^*_{i,\text{out}} &= \epsilon_{i,\text{out}} - \epsilon_{i,\text{lux}}
\end{aligned}
$$

Since the demographic variables do not vary across choices, we can only identify the relative impact of the change in a predictor on the choice of department store relative to the luxury department store (or outlet store to luxury department store). Therefore, the coefficients that can be estimated are C^*_{dept}, C^*_{out}, α^*_{dept}, α^*_{out}, β^{*2}_{dept}, β^{*2}_{out}, β^{*3}_{dept}, and β^{*3}_{out}. By applying the logit fomula, the final estimated equations are

$$
\begin{aligned}
\ln\left(\frac{P(\text{shop = department store})}{P(\text{shop = luxury store})}\right) &= C^*_{\text{dept}} + \alpha^*_{\text{dept}} age_i + \beta^{*2}_{\text{dept}}(inc_i = 2) + \beta^{*3}_{\text{dept}}(inc_i = 3) + \epsilon^*_{i,\text{dept}} \\
\ln\left(\frac{P(\text{shop = outlet store})}{P(\text{shop = luxury store})}\right) &= C^*_{\text{out}} + \alpha^*_{\text{out}} age_i + \beta^{*2}_{\text{out}}(inc_i = 2) + \beta^{*3}_{\text{out}}(inc_i = 3) + \epsilon^*_{i,\text{out}}
\end{aligned}
\tag{3.46}
$$

The data are read into SAS from a comma-separated text file using the IMPORT procedure. The output data set is named shopping and the CONTENTS procedure is used to view information on the variables.

```
proc import datafile = "shopping.csv" out = shopping
dbms = csv replace;
getnames = yes;
run;

proc contents data = shopping;
run;
```

Figure 3.15 Contents

The CONTENTS Procedure

Data Set Name	WORK.SHOPPING	**Observations**	200
Member Type	DATA	**Variables**	4
Engine	V9	**Indexes**	0
Created	03/01/2018 12:59:22	**Observation Length**	32
Last Modified	03/01/2018 12:59:22	**Deleted Observations**	0
Protection		**Compressed**	NO
Data Set Type		**Sorted**	NO
Label			
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat
4	age	Num	8	BEST12.	BEST32.
1	id	Num	8	BEST12.	BEST32.
2	income	Num	8	BEST12.	BEST32.
3	store	Num	8	BEST12.	BEST32.

The data set contains 4 variables on 200 individuals. The choice variable is $store$, store type. This is a categorical variable. If an individual chooses a department store to shop, the value of $store = 1$. Choice of a luxury store is denoted by $store = 2$ and choice of an outlet is denoted by $store = 3$. $income$ is also a categorical variable, the value of which can be 1, 2, or 3, with 1 being the lowest and 3 the highest. age is the age of each individual and is the single continuous variable. We start with PROC FREQ to obtain summary statistics on the variables of interest. The FREQ procedure is useful for obtaining crosstabulation tables that can be used to summarize association between variables. The TABLES statement specifies a crosstabulation of store with income.

```
proc freq data = shopping;
   tables store * income / chisq norow nocol nofreq;
run;
```

Figure 3.16 Frequency of Shopping Choices

The FREQ Procedure

Percent

Table of store by income

	income			
store	**1**	**2**	**3**	**Total**
1	8.00	10.00	4.50	22.50
2	9.50	22.00	21.00	52.50
3	6.00	15.50	3.50	25.00
Total	47 23.50	95 47.50	58 29.00	200 100.00

Figure 3.16 shows that 29 percent of the sample is in the highest income category while 25.5 percent of the sample is in the lowest income category. The majority of the sample chooses to shop in the luxury department store. For all three income groups, the most popular store type is the luxury store. Curiously, middle income individuals constitute the greatest share of consumers shopping at outlets. The same information can be displayed graphically using PROC GCHART. The vbar3d statement asks for a 3d bar chart for each store showing the shopping choices of different

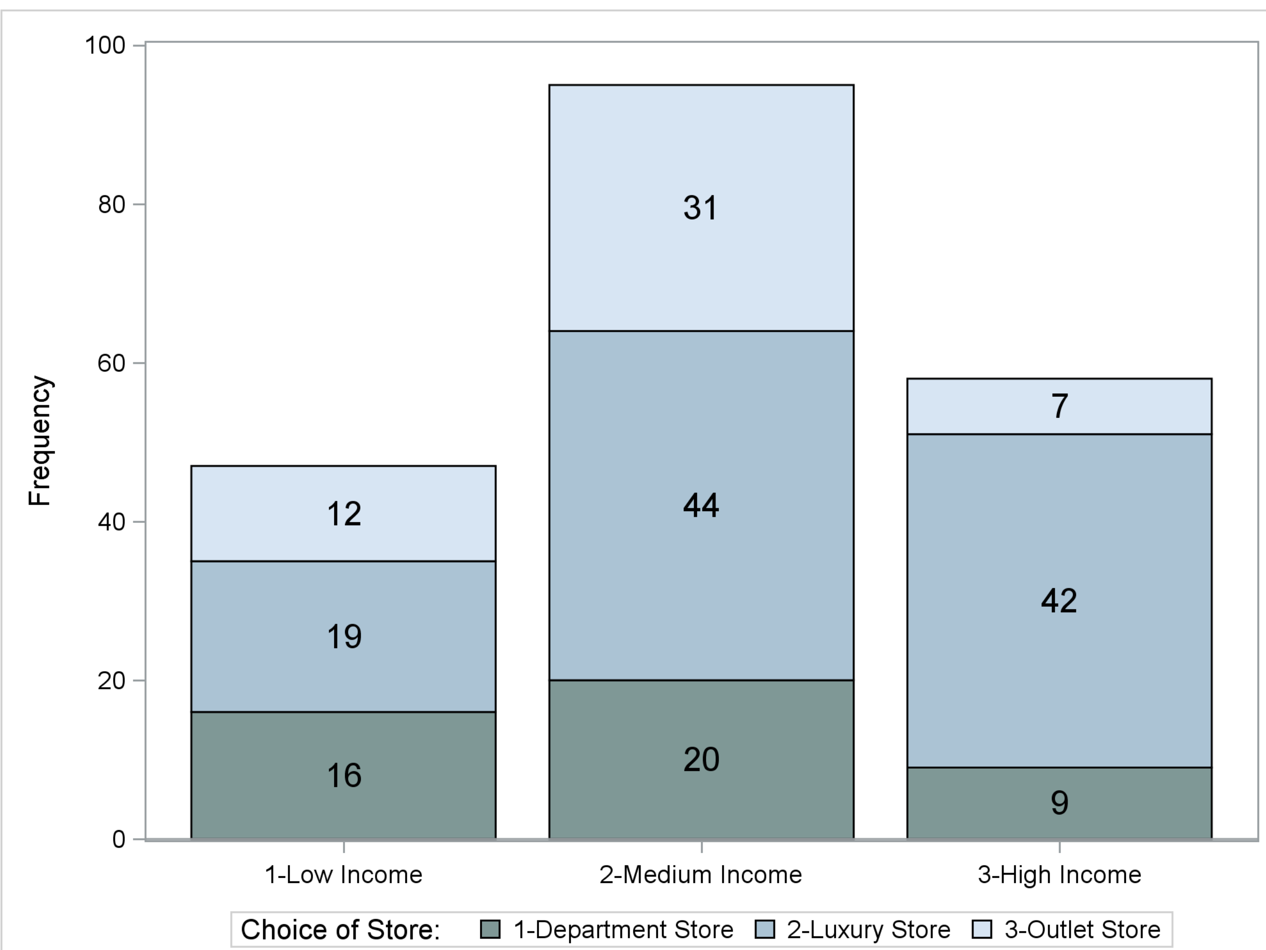
income groups.

```
proc format;
   value incomeform 1='1-Low Income'
                    2='2-Medium Income'
                    3='3-High Income';

   value storeform 1='1-Department Store'
                    2='2-Luxury Store'
                    3='3-Outlet Store';
run;

proc sgplot data = shopping;
   format income incomeform.;
   format store storeform.;
   styleattrs datacolors=('#7f9896' '#abc3d4' '#d7e5f3') datacontrastcolors=(black) ;
   vbar income /group=store seglabel seglabelattrs=(size=12);
   label store = 'Choice of Store:';
   xaxis display=(nolabel);
run;
```

Figure 3.17 Income Grouped by Shopping Choice



We have not said anything about the relationship between age and shopping choice. Through the use of PROC MEANS, and a BY statement, we can easily obtain summary statistics by store type. The data are first sorted according to store type and then the relevant call to the MEANS procedure is given.

```
proc sort data = shopping;
   by store;
run;

proc means data = shopping;
   var age;
   by store;
run;
```

Figure 3.18 Summary Statistics by Store

The MEANS Procedure

store=1

Analysis Variable : age				
N	Mean	Std Dev	Minimum	Maximum
45	51.3333333	9.3977754	31.0000000	67.0000000

store=2

Analysis Variable : age				
N	Mean	Std Dev	Minimum	Maximum
105	56.2571429	7.9433433	33.0000000	67.0000000

store=3

Analysis Variable : age				
N	Mean	Std Dev	Minimum	Maximum
50	46.7600000	9.3187544	31.0000000	67.0000000

The maximum age shopper at every store category is 67 years of age, but the mean age does vary by category. According to Figure 3.18, the outlet store has the youngest average shopper while the luxury store has the oldest average shopper. The LOGISTIC procedure can be used to estimate a multinomial logistic regression model. The choice variable $store$ and the independent variable $income$ are both categorical variables and should be indicated as such in the CLASS statement. The baseline category for $store$ is the luxury store ($store = 2$). For the dummy variable $income$, we choose $income = 1$ as our baseline case.

```
proc logistic data = shopping;
   class store (ref = "2") income (ref = "1") / param = ref;
   model store = income age / link = glogit;
run;
```

We could also manually create dummy variables for the categorical predictors using a DATA step.

```
data shopping;
set shopping;
   income_1=(income=1);
   income_2=(income=2);
   income_3=(income=3);
run;
```

Since income is now indicated by dummy variables, instead of a single categorical variable, the corresponding CLASS statement can be dropped. However, the MODEL statement has to be adjusted to include the additional predictors. In any event, both of these approaches will produce the same results.

```
proc logistic data = shopping;
   class store (ref = "2") / param = ref;
   model store = income_2 income_3 age / link = glogit;
run;
```

In the output shown in Figure 3.19, we can see that

1. a one unit increase in age is associated with a 0.058 decrease in the relative log odds of choosing a department store versus a luxury store.
2. a one unit increase in age is associated with a 0.1136 decrease in the relative log odds of choosing an outlet store versus a luxury store.
3. moving from the lowest income group ($income = 1$) to the highest income group ($income = 3$) leads to a decrease in relative log odds of shopping in a department store vs. a luxury store.

Figure 3.19 Logit Model Estimates

The LOGISTIC Procedure

Model Information	
Data Set	WORK.SHOPPING
Response Variable	store
Number of Response Levels	3
Model	generalized logit
Optimization Technique	Newton-Raphson

Analysis of Maximum Likelihood Estimates

Parameter		store	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	1	2.8522	1.1664	5.9790	0.0145
Intercept		3	1	5.2182	1.1635	20.1128	<.0001
income	2	1	1	-0.5333	0.4437	1.4444	0.2294
income	2	3	1	0.2914	0.4764	0.3742	0.5407
income	3	1	1	-1.1628	0.5142	5.1137	0.0237
income	3	3	1	-0.9827	0.5956	2.7224	0.0989
age		1	1	-0.0579	0.0214	7.3200	0.0068
age		3	1	-0.1136	0.0222	26.1392	<.0001

Odds Ratio Estimates

Effect	store	Point Estimate	95% Wald Confidence Limits	
income 2 vs 1	1	0.587	0.246	1.400
income 2 vs 1	3	1.338	0.526	3.404
income 3 vs 1	1	0.313	0.114	0.856
income 3 vs 1	3	0.374	0.116	1.203
age	1	0.944	0.905	0.984
age	3	0.893	0.855	0.932

In the logit model, we assume that consumers' tastes for product characteristics, as presented by the coefficients β, do not vary across consumers. A more general model allows β to vary across consumers, that is, $\beta_n \neq \beta_{n'}$ when $n \neq n'$.

The ratio between the probability of choosing product i and j can be written as

$$\frac{P_{ni}}{P_{nj}} = e^{V_{ni}-V_{nj}}$$

This ratio only depends on the characteristics of product j and i, but does not depend on characteristics of any other products. This property called the Independence of Irrelevant Alternatives (IIA) property.

While the IIA property is realistic in some choice situations, this type of substitution patterns is restrictive. Consider the case where a new product enters the market. If the characteristics of this new product are very similar to those of product j but not so close to product i, the relative odd of choosing product i and j should be affected.

The restriction of the IIA property is also reflected by the cross price elasticity that we derived in the previous section. Recall the cross price elasticity of product i derived from the logit model does not depend on the characteristics of product i. This implies that when there is a change in the price of product j, the demand for all other products will change by the same percentage. This is not realistic in many cases because the demand for close substitutes of product j should be affected more than those that are not close substitutes of product j.

3.5.3.2 An Example with Aggregate Level Data

In the previous section, we showed how consumer demand can be estimated when we observe choices made by each individual. It is sometimes difficult for researchers to collect micro level data in reality. Aggregate level data, however, is more widely available. We only need to observe choices made by consumers at the aggregate level, that is, the market share of products. This section contains a brief introduction to the estimation of discrete choice models when only aggregate level data is available. We first discuss model specification and identification and then present an example to show how estimation can be realized in SAS.

Suppose there are J products in the market and each individual consumer buys at most one unit of product. The utility function of consumer i for product j is specified as

$$\begin{aligned} U_{ij} &= X_j\beta - \alpha p_j + \eta_j + \epsilon_{ij} \\ &= \delta_j + \epsilon_{ij} \end{aligned} \tag{3.47}$$

where the mean utility level of product j can be written as

$$\delta_j = X_j\beta - \alpha p_j + \eta_j \tag{3.48}$$

X_j is a vector of characteristics of product j, where $j = 0, 1, 2..., J$. p_j is the price of product j and α is the regression coefficient of price. β is a vector of regression coefficients for X_j. For example, X_j can be fuel efficiency, number of seats, etc, for a car. Unobserved time-invariant product characteristics are denoted by η_j. η_j can be thought of as the mean of the consumers' valuation of the unobserved product characteristics of product j. ϵ_{ij} is i.i.d. across products and consumers and follows the extreme value distribution.

Since consumer i chooses the product that yields the highest utility, the probability for consumer i to choose product j can be written as

$$\begin{aligned} s_{ij} &= \text{Prob}\left(U_{ij} > U_{il}, l = 0, 1, ..., j-1, j+1, ..., J\right) \\ &= \frac{e^{\delta_{ij}}}{e^{\delta_0} + \sum_{k=1}^{J} e^{\delta_k}} \end{aligned} \tag{3.49}$$

where $\delta_0 = 0$ for the outside option, product 0. The outside option is a failure to purchase any of the available alternatives, thus spending the entire income of the consumer on goods outside of the analysis. Integrating over individual consumers, a formula for the market share of each product can be obtained.

$$\begin{aligned} s_j &= \frac{\sum_i s_{ij}}{\sum_k \sum_i s_{ik}} \\ &= N * \frac{e^{\delta_{ij}}}{1+\sum_{k=1}^{J} e^{\delta_k}} \\ &= \frac{e^{\delta_j}}{1+\sum_{k=1}^{J} e^{\delta_k}} \end{aligned} \tag{3.50}$$

Next we convert the demand functions to obtain estimates of the mean utility level, δ_j as a function of the market share, s_j

$$\begin{aligned} \frac{s_j}{s_0} &= \frac{e^{\delta_j}}{1+\sum_{k=1}^{J} e^{\delta_k}} \left(\frac{e^{\delta_0}}{1+\sum_{k=1}^{J} e^{\delta_k}} \right)^{-1} \\ &= e^{\delta_j} \\ \ln(s_j) - \ln(s_0) &= \delta_j \end{aligned} \tag{3.51}$$

Therefore, the demand estimation can be specified as

$$\ln(s_j) - \ln(s_0) = X_j \beta - \alpha p_j + \eta_j \tag{3.52}$$

Similar to the previous section, we can derive the own price elasticity and cross price elasticity of a product as follows.

$$\frac{\partial s_j}{\partial p'_j} = \frac{e^{\delta_j} e^{\delta'_j}}{\left[1+\sum_{k=1}^{J} e^{\delta_k}\right]^2} \tag{3.53}$$

where j and j' can be any two choices. The own price elasticity of product j can be calculated as

$$e_{jj} = \frac{\partial s_j}{\partial p_j} \frac{p_j}{s_j} = -\alpha p_j (1 - s_j) \tag{3.54}$$

Similarly, the cross price elasticity can be calculated as

$$e_{jj'} = \frac{\partial s_j}{\partial p'_j} \frac{p'_j}{s_j} = \alpha s_{j'} p_{j'} \tag{3.55}$$

The own price elasticity of demand is negative, which is consistent with law of demand. The cross price elasticity of demand is also negative. Because product j does not enter this formula, a change in price of product j has the same effect on all the other products. It may not be appropriate in some cases because closer competitors of product j should have a larger impact. This is the IIA critique that simple logit models suffer.

The difficulty in estimating demand models lies in endogeneity. Some of the product characteristics are unobserved (to the researcher) and these characteristics may be correlated with the independent variables. For example, we want to estimate demand for cereal. Researchers can observe how much sugar, fat, and calories per 100 gram of each brand. But researchers cannot observe the taste of each brand. Intuitively, taste should be positively correlated with price. Therefore, estimating this equation by ordinary least squares (OLS) would yield biased estimates.

To deal with endogeneity, we need to find instrumental variables (IVs). A good IV should be uncorrelated with the unobserved product characteristic, η_j, but highly correlated with the endogenous variable, in our case, p_j. The first strategy is to assume product characteristics, X_j are exogenous and to use X_j (or functions of X_j) as IVs. X_j is obviously correlated with p_j because firms set their prices based on these characteristics. However, for some products, the unobserved characteristics, η_j are correlated with X_j. For example, how good the cereal tastes may depend on the amount of sugar it contains.

The second strategy is to use the average characteristics of all competing products as IVs to control for p_j. For example, we can use the average price of all other products which should be positively correlated with p_j because the price of product j is likely to increase if all its competitors increase their prices. This IV is unlikely to be correlated with the unobserved characteristics of product j. For instance, the taste of cereal is not likely to be associated with the average price of its rivals.

In this example, we estimate consumer preferences for cereal using aggregate level data. The data set includes product information of the top selling 50 brands of cereal in 1992, such as the market share of each brand, average retail price, level of sugar, fat, and calories. Two dummy variables indicate whether the brand targets families and whether the flavor is child or adult oriented. The market shares are shares of total cereal purchased during 1992. For simplicity, we assume that the outside option is the composite basket of all other brands in the market.

3.5.3.3 Model Specification

The utility specification for individual i of choosing brand j can be written as

$$u_{ij} = X_j\beta - \alpha p_j + \eta_j + \epsilon_{ij} \tag{3.56}$$

where X_j are characteristics of brand j, η_j is unobserved characteristics of product j. ϵ_{ij} is i.i.d. type I extreme value. The mean utility level of product j can then be written as

$$\delta_j = X_j\beta - \alpha p_j + \eta_j$$

The estimation equation is then

$$\ln(s_j) - \ln(s_0) = X_j\beta - \alpha p_j + \eta_j$$

where s_0 is the market share of the outside option. As discussed in the previous section, the price variable, p_j is endogenous and OLS estimation can lead to bias. In the estimation, we will use different sets of IVs to estimate consumer demand and potentially correct for the endogeneity bias. We begin by reading the data into SAS using the IMPORT statement.

```
proc import out = WORK.cereal datafile= "Cereal1"
   dbms = xls replace;
   getnames = yes;
run;
```

The market share of the outside option is 24.29 percent and the average price of the outside option is 2.68. We construct the dependent variable and deflate the prices using a DATA step.

```
data cereal;
set cereal;
   Y = log(Mkt_share) - log(24.29);
   price = Avg_Trans_Price/2.68;
run;
```

We can start with OLS estimation implement in PROC REG.

```
proc reg data=cereal;
   model Y = FAM_Dummy Kids_Dummy Cals Fat Sugar price;
run;
```

Although OLS gives the correct sign for the price parameter, endogeneity could be an issue. To improve our estimation, we use several sets of IVs and compare these results with the OLS results.

Figure 3.20 OLS Parameter Estimates

The REG Procedure
Model: MODEL1
Dependent Variable: Y

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	-2.80294	0.60455	-4.64	<.0001
Fam_Dummy	Fam_Dummy	1	0.57541	0.17768	3.24	0.0023
Kids_Dummy	Kids_Dummy	1	0.07034	0.20915	0.34	0.7383
Cals	Cals	1	0.00246	0.00275	0.90	0.3753
Fat	Fat	1	0.01125	0.05168	0.22	0.8288
Sugar	Sugar	1	-0.04231	0.01556	-2.72	0.0094
price		1	-0.21037	0.37430	-0.56	0.5770

IV Method 1

The first set of IVs are constructed using average characteristics of all the other brands produced by the same company. These IVs are likely to be correlated with the price variable but unlikely to be correlated with the unobserved product characteristics. To obtain the sum of the characteristics for each company, we use PROC SUMMARY and the CLASS statement. This instructs PROC SUMMARY to compute the statistics by company. After obtaining the output data set sum, this data is merged with the original data set.

```
proc summary data = cereal;
   var price cals fat sugar;
   class company_id;
   output out = sum_company
   n = count_company
   sum=;
run;
```

```
proc print data = sum_company;
run;

/*Generate sum of characteristics variables*/
data sum_company;
set sum_company;
   if _TYPE_ = 0 then DELETE;
   rename price = sum_price_company
                cals = sum_cals_company
                fat = sum_fat_company
                sugar = sum_sugar_company
                count_identifier = count_company;
run;

/*Merge sum table to main table*/
data cereal;
   merge cereal sum_company(keep=company_id count_company sum_price_company
       sum_cals_company sum_fat_company sum_sugar_company);
   by company_id;
run;

/*Generate averages of characteristics of other products
  produced by the same company*/
data cereal;
set cereal;
   ave_price_company = (sum_price_company -price) / (count_company - 1);
   ave_cals_company = (sum_cals_company -cals) / (count_company - 1);
   ave_fat_company = (sum_fat_company - fat) / (count_company - 1);
   ave_sugar_company = (sum_sugar_company - sugar) / (count_company - 1);
run;

proc print data = cereal;
run;
```

There are a number of ways to implement two stage least squares (2SLS) in SAS; in this instance we use PROC SYSLIN. The 2SLS option statement specifies the 2SLS method. The ENGOGENOUS statement specifies which regressor is endogenous. The first stage predicted values are substituted for this variable. Note that the dependent variable, market share, is endogenous. But the dependent variable is not used as a regressor in this type of model, and hence should not be included in the ENDOGENOUS statement. The INSTRUMENT statement specifies the instrumental variables that are used to control for the right-hand side endogenous variable. In our case, we should include the average characteristics of all the other brands produced by the same company, including ave_price_company, ave_cals_company, ave_fat_company, and ave_sugar_company. We also need to include the other independent variables in the second stage, including Cals, Fat, Sugar, FAM_Dummy, and KIDS_Dummy.

```
/*2SLS using average characteristics of other products produced by same company*/
proc syslin data = cereal 2sls ;
   endogenous price ;
   instruments ave_price_company ave_cals_company ave_fat_company
                      ave_sugar_company FAM_Dummy  Kids_Dummy Cals Fat Sugar;
   model Y = FAM_Dummy Kids_Dummy Cals Fat Sugar price;
run ;
```

Figure 3.21 Estimates from PROC SYSLIN

The SYSLIN Procedure
Two-Stage Least Squares Estimation

Model	Y
Dependent Variable	Y

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variable Label
Intercept	1	-4.17869	0.977925	-4.27	0.0001	Intercept
Fam_Dummy	1	0.684948	0.201118	3.41	0.0014	Fam_Dummy
Kids_Dummy	1	0.043367	0.227200	0.19	0.8495	Kids_Dummy
Cals	1	0.004429	0.003154	1.40	0.1674	Cals
Fat	1	-0.00024	0.056357	-0.00	0.9966	Fat
Sugar	1	-0.04811	0.017142	-2.81	0.0075	Sugar
price	1	0.817732	0.677375	1.21	0.2340	

Another way to implement 2SLS estimation is to use the MODEL procedure. In the PARAMETERS statement, the parameters to be estimated are specified. In this case, they are the constant term, parameters for Cals, Fat, Sugar, FAM_Dummy, and KIDS_Dummy. The EXOGENOUS statement specifies the exogenous independent variables and the ENDOGENOUS statement specifies the endogenous variables. The estimation method selected comes after the slash in the FIT statement. The INSTRUMENTS statement follows the FIT statement and in this case we include the average characteristics of all the other brands produced by the same manufacturer, as well as all the exogenous variables as instruments with the _EXOG_keyword. PROC MODEL and PROC SYSLIN return the same parameter estimates.

```
proc model data=cereal;
   parameters b0 a1 a2 a3 a4 a5 a6;
   exogenous FAM_Dummy Kids_Dummy Cals Fat Sugar;
   endogenous price;
   Y = b0 + a1 * FAM_Dummy + a2 * Kids_Dummy
            + a3 * Cals + a4 * Fat + a5 * Sugar + a6 * price;
   fit Y / 2sls;
   instruments _exog_ ave_price_company ave_cals_company ave_fat_company
   ave_sugar_company;
run;
```

Figure 3.22 Estimates from PROC MODEL

The MODEL Procedure

Model Summary	
Model Variables	7
Endogenous	1
Exogenous	5
Parameters	7
Equations	1
Number of Statements	1

Model Variables	Fam_Dummy Kids_Dummy Cals Fat Sugar price Y
Parameters	b0 a1 a2 a3 a4 a5 a6
Equations	Y

Figure 3.22 *continued*

Nonlinear 2SLS Parameter Estimates

Parameter	Estimate	Approx Std Err	t Value	Approx Pr > |t|
b0	-4.17869	0.9779	-4.27	0.0001
a1	0.684948	0.2011	3.41	0.0014
a2	0.043367	0.2272	0.19	0.8495
a3	0.004429	0.00315	1.40	0.1674
a4	-0.00024	0.0564	-0.00	0.9966
a5	-0.04811	0.0171	-2.81	0.0075
a6	0.817732	0.6774	1.21	0.2340

IV Method 2

The second set of IVs can be constructed by using the average characteristics of products produced by rivals. The construction of similar to the first method.

```
/*Calculate the sum of all variables*/
proc summary data = cereal;
  var price cals fat sugar;
  output out = sum n=count sum = sum_price sum_cals sum_fat sum_sugar;
run;

proc print data = sum;
run;

/*Merge sum of characteristics to every observation*/
data cereal;
   if _N_ = 1 then set sum(keep = count sum_price sum_cals sum_fat sum_sugar);
set cereal;
   ave_price_rival = (sum_price - sum_price_company) / (count - count_company);
   ave_cals_rival = (sum_cals - sum_cals_company) / (count - count_company);
   ave_fat_rival = (sum_fat - sum_fat_company) / (count - count_company);
   ave_sugar_rival = (sum_sugar - sum_sugar_company) / (count - count_company);
run;

proc print data = cereal;
run;
```

Both PROC SYSLIN and PROC MODEL can be used to implement the estimation and, again, produce the same results.

```
proc syslin data = cereal 2sls ;
   endogenous price ;
   instruments ave_price_rival ave_cals_rival ave_fat_rival ave_sugar_rival
                      FAM_Dummy Kids_Dummy Cals Fat Sugar ;
   model Y = FAM_Dummy Kids_Dummy Cals Fat Sugar price;
run ;

proc model data=cereal;
   parameters b0 a1 a2 a3 a4 a5 a6;
   exogenous FAM_Dummy Kids_Dummy Cals Fat Sugar;
       endogenous price;
   Y = b0 + a1 * FAM_Dummy + a2 * Kids_Dummy
            + a3 * Cals + a4 * Fat + a5 * Sugar + a6 * price;
```

```
   fit Y / 2sls;
   instruments _exog_ ave_price_rival ave_cals_rival ave_fat_rival ave_sugar_rival;
run;
```

Figure 3.23 Logit Estimates Using Rival Char. IV

The MODEL Procedure

Model Variables	Fam_Dummy Kids_Dummy Cals Fat Sugar price Y
Parameters	b0 a1 a2 a3 a4 a5 a6
Equations	Y

Nonlinear 2SLS Parameter Estimates

Parameter	Estimate	Approx Std Err	t Value	Approx Pr > |t|
b0	-3.61667	0.8683	-4.17	0.0001
a1	0.640199	0.1893	3.38	0.0015
a2	0.054387	0.2158	0.25	0.8022
a3	0.003625	0.00296	1.22	0.2273
a4	0.004453	0.0535	0.08	0.9340
a5	-0.04574	0.0162	-2.82	0.0073
a6	0.397737	0.5942	0.67	0.5069

Lastly, the average characteristics of all other products could be used as IVs.

```
data cereal;
set cereal;
   ave_price_other = (sum_price - price) / (count - 1);
   ave_cals_other = (sum_cals - cals) / (count - 1);
   ave_fat_other = (sum_fat - fat) / (count - 1);
   ave_sugar_other = (sum_sugar - sugar) / (count - 1);
run;

/*Using average characteristics of all other products as IVs to run 2SLS*/
proc syslin data = cereal 2sls ;
endogenous price ;
instruments ave_price_other ave_cals_other ave_fat_other ave_sugar_other
                     FAM_Dummy Kids_Dummy Cals  Fat Sugar ;
model Y = FAM_Dummy Kids_Dummy Cals Fat Sugar price;
run ;
```

Figure 3.24 Logit Estimates Using Avg. Char. IV

The SYSLIN Procedure
Two-Stage Least Squares Estimation

Model	Y
Dependent Variable	Y

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Variable Label
Intercept	1	-2.80294	0.604554	-4.64	<.0001	Intercept
Fam_Dummy	1	0.575408	0.177680	3.24	0.0023	Fam_Dummy
Kids_Dummy	1	0.070342	0.209147	0.34	0.7383	Kids_Dummy
Cals	1	0.002461	0.002747	0.90	0.3753	Cals
Fat	1	0.011246	0.051680	0.22	0.8288	Fat
Sugar	1	-0.04231	0.015556	-2.72	0.0094	Sugar
price	1	-0.21037	0.374297	-0.56	0.5770	

3.6 Conclusion

We have only examined some of the functional forms in use for demand analysis. We would be remiss if we did not note some of the forms that we have not discussed. Several of these forms are used in later chapters on the derived demand of producers. We have not treated the translog model, the generalized Leontief model, the fourier flexible form, the miniflex laurent, the normalized quadratic, or the asymptotically ideal model. Many discrete choice models have also been omitted. However, the important concepts surrounding applied demand analysis have been addressed. Likewise, the use of several SAS procedures has been demonstrated.

Ready to take your SAS® and JMP® skills up a notch?

Be among the first to know about new books, special events, and exclusive discounts.
support.sas.com/newbooks

Share your expertise. Write a book with SAS.
support.sas.com/publish

sas.com/books
for additional books and resources.

sas
THE POWER TO KNOW®