Hassle Costs and Price Discrimination: An Empirical Welfare Analysis: Online Appendix

Guillermo Marshall*

August 20, 2014

Appendix A: Model comparison

In this appendix, I discuss the differences in the estimates of the model that I use in this paper (baseline model) with the estimates of a random coefficient logit model.

Table 7 shows the results of the estimation of both models using the subsample of customers who always purchased Diet Coke products. The table shows that the coefficients on price and the refillable dummy are not statistically different across models. It also shows that most of the interaction effects between the customer characteristics and both the refillable dummy and price preserve the signs reported in Table 5. One exception is the negative coefficient on the interaction between trip frequency and the refillable dummy in the baseline estimates, which is inconsistent with Tables 3 and 5. In spite of this inconsistency, the estimates of both these models preserve the negative relationship between the price coefficient and the taste for refillables that was discussed above (see Figure 3).

An interesting difference between models is how the models rely on unobserved heterogeneity to explain choice heterogeneity, which is reflected in the relative magnitude of coeffi-

*University of Illinois at Urbana-Champaign, Department of Economics, 214 David Kinley Hall, 1407 W. Gregory, Urbana, Illinois 61801 (e-mail: gmarshll@illinois.edu). Acknowledgements: I am grateful to Liran Einav, Igal Hendel, Fernando Luco, Aviv Nevo, Álvaro Parra, Esteban Petruzzello, Tiago Pires, Rob Porter, Alberto Salvo, Anthony Wray, and three anonymous referees for comments and suggestions.

cients σ'_η and σ'_μ across models. In the logit model, unobserved heterogeneity in the refillable coefficient plays almost no role, which suggests that the observed customer heterogeneity together with the product-specific taste shocks are able to accommodate the differences across customers in the willingness to purchase refillables. On the contrary, in the baseline model, the unobserved heterogeneity exhibits a relatively higher variance, as the unobserved heterogeneity takes the role of explaining choice heterogeneity among customers who share the same observables.

Table 7: Simulated maximum likelihood estimates: Diet Coke subsample

	Baseline Model		RC Logit Model	
	Estimate	St. error	Estimate	St. error
Price	-6.5918	0.1117	-6.9483	0.2432
Price * Trip frequency	-0.0510	0.0104	-0.6089	0.0276
Price * Mean expenditure	0.0081	0.0006	0.0043	0.0003
Price * Age	-0.0088	0.0025	0.0004	0.0003
Refillable	-5.9084	0.4330	-6.3439	0.1790
Refillable * Trip frequency	-0.3873	0.1212	0.1062	0.0753
Refillable * Mean expenditure	-0.0100	0.0014	-0.0068	0.0013
Refillable * Age	0.0353	0.0064	0.0270	0.0023
Container size (L)	4.8586	0.0698	5.2818	0.1854
Diet Coke	-0.1216	0.0497	1.7666	0.1918
CF parameter	0.0121	0.0014	8.4983	0.2416
σ'_η	0.1229	0.0162	-0.7674	0.1718
σ'_μ	0.8032	0.1312	-35.2966	22.6623
$-\frac{1}{N}\mathcal{SL}_N(\hat{\theta})$	0.2420		0.2293	
Number of choices	129,749		129,749	

Notes: Estimates are computed using the subsample of customers who always bought Diet Coke. Standard errors were computed using the bootstrap. Mean trip expenditure is the customer's average expenditure across all trips to the store. Prices and trip expenditure are measured in US dollars. Trip frequency is the ratio of weeks in which the customer visited a store over total weeks in the sample period (See Table **??**). The parameters σ'_η and σ'_μ define the standard deviations of the distributions of the random coefficients on price and refillable disamenities: $\sigma_\eta = \exp\{\sigma'_\eta\}$ and $\sigma_\mu = \exp\{\sigma'_\mu\}$. CF parameter is the control function parameter.

Appendix B: Estimation

In this appendix, I discuss the code that was used for the estimation of the model in three steps. First, I discuss how the choice data must be organized for it to be compatible with the code. Second, I outline the functioning of the code. Lastly, I make line-by-line comments to the MATLAB m-files used for the estimation.

Data organization

The choice data (i.e., file data.csv in m-file below) include the store trips of all customers. A store trip is defined as a customer–store–time combination. Two examples of store trips are displayed in Table 8. For each store trip, the data list all products that were available as well as the characteristics of these products (i.e., size, brand, price, unobservable estimate, and container type). The total number of rows in the data.csv file is $\sum_{n=1}^{N} |J_n|$, where N is the total number of trips, and $|J_n|$ is the number of products that were available to the customer in trip n.

For each trip, the data also include customer characteristics and a variable that indicates whether the customer chose the outside option or not (i.e., variable Outside in Table 8). If the customer purchases a product (i.e., when Outside is equal to zero), the data is organized so that the product that was purchased by the customer is listed last in the list of products. For instance, in trip number 6 (i.e., last four rows of Table 8), where the variable Outside takes the value zero, the customer purchased a Brand 1–1.5L–nonrefillable product, which is the product that is listed in the last position.

Code

The estimation is performed using two files: mainfulluni_all.m and lik.m. The first file, mainfulluni_all.m, loads the data, defines the variables, and calls the routine that optimizes the likelihood function. The second file, lik.m, evaluates the likelihood function for a given vector of parameters.

The choice probabilities (see Equation 3) used to construct the likelihood function are simulated using random draws from the distributions of the unobserved customer charac-

Table 8: Choice data: data.csv matrix

Variable:	Price	Refillable	Size	Brand	Error	Outside	Age	Mean Expenditure	Trip frequency	Choice Set Card	Trip Code
Column number (in data matrix)	5	8	9	10	11	12	13	15	16	17	18
	1.88	0	1500	1	-0.0060	1	46	310.62	0.2059	4	5
	3.04	0	3000	1	-0.0867	1	46	310.62	0.2059	4	5
	1.63	1	1500	1	0.1111	1	46	310.62	0.2059	4	5
	2.15	1	2500	1	-0.0218	1	46	310.62	0.2059	4	5
	3.31	0	3000	1	0.1116	0	46	310.62	0.2059	4	6
	1.60	1	1500	1	0.0790	0	46	310.62	0.2059	4	6
	2.22	1	2500	1	0.0485	0	46	310.62	0.2059	4	6
	1.96	0	1500	1	0.0210	0	46	310.62	0.2059	4	6

teristics (see Equation 2). For each pair of draws, the code finds the product that gives the customer the highest utility, which is a degenerate random variable (i.e., the indicator function in Equation 3). The choice probabilities are obtained by averaging the product of the degenerate random variable and the probability of purchasing an inside option across all pairs of random variable draws.

In what follows, I provide a line-by-line description of both files.

```matlab
%%%%%% File name: mainfulluni_all.m


%%%%%% Description: This file loads the data and optimizes, given an
%%%%%% initial vector, the likelihood function of the model discussed
%%%%%% in the paper.


clear
clc
format long


%%%%%%Loading data
data=dlmread('~/Bebidas/Choices072013/data.csv');
           %data is the file with the data matrix.

load draws %draws is the file with variables: drawsp, ns, drawsr.
           %ns: Number of draws from random variables for numerical
           %integration
           %drawsp: Draws from a N(0,1).
           %drawsr: Draws from a N(0,1).

%%%%%%Declaring global variables
global X K NN N choiceset outside price refillable drawsp ns ...
    drawsr typecode

%%%%%%Defining variables
%%%Variables that vary at the customer level
meanexp=data(:,15); %Mean trip expenditure in USD

age=data(:,13); %Age

shareweek=data(:,16); %Trip frequency

refintensity=data(:,1); %Refillable purchase frequency



%%%%Variables that vary at the customer-trip level
choicesetcardvect=data(:,17); %Cardinality of choice set faced by customer.

outside=data(:,12); %Binary variable that takes value 1
                    %if customer chose outside option



%%%Variables that vary at the product level
price=data(:,5); %Price in USD

cnt_cont=data(:,9)/1000; %Container size (in liters)

refillable=data(:,8); %Refillable container dummy variable

error=data(:,11); %Control function estimate

typecode=data(:,10); %Brand code

dummytype=[typecode==1 typecode==2 typecode==3]+0; %Brand dummies:
                                                   %1 (Diet Coke),
```

```
                                                   %2 (Coke),
                                                   %3 (Coke Zero)


%%%Interaction variables
page=price.*age; %Price * Age

pexp=price.*meanexp; %Price * Mean expenditure

pshare=price.*shareweek; %Price * Trip frequency

rshare=refillable.*shareweek; %Refillable dummy * Trip frequency

rage=refillable.*age; %Refillable dummy * Age

rexp=refillable.*meanexp; %Refillable dummy * Mean expenditure



%%%Auxiliar variables
N=671086; %Number of trips to the store.

NN=size(data,1); %Number of rows in dataset.

choiceset=zeros(N,2); %Matrix of indeces that identify each trip in data.
                      %Column 1: indicates first observation of each trip.
                      %Column 2: Indicates last observation of each trip.
                      %Note: All observations of a trip are consecutive
                      %rows in the data matrix.
choiceset(1,1)=1;
choiceset(1,2)=choicesetcardvect(1,1);
for k=2:N
choiceset(k,1)=choiceset(k-1,2)+1;
choiceset(k,2)=choicesetcardvect(choiceset(k-1,2)+1,1)+choiceset(k,1)-1;
end



%%%%%%Variables included in the model
X=[price pshare pexp page refillable rshare rexp rage cnt_cont ...
    dummytype error];

K=size(X,2)+2; %Number of columns in matrix X



%%%%%%Deleting unused variables
clear data



%%%%%%Likelihood function optimization
Options = optimset('Algorithm', 'interior-point', 'Display','iter', ...
    'MaxIter',5000,'MaxFunEvals',10000,'TolX', 1e-10, 'TolFun', 1e-7);
                %Optimization options

P1=324; %Seed value

randn('state',P1); %Seed for initial point.

beta0=0.01*randn(K,1); %Initial point.

[betahat,fval]=fminsearch(@(beta) lik(beta), beta0,Options); %Optimization
```

```
%Note: I suggest running this code for several initial points.


%%%%%%Optimization output
betahat %Vector of parameters that minimizes likelihood function
fval    %Function value at parametor vector betahat.
```

Published with MATLAB® 7.11

```
%%%%%% File name: lik.m


%%%%%% Description: This file evaluates the likelihood function of the
%%%%%% model discussed in this paper, given a vector of parameters beta.


function L =lik(beta)

global X N choiceset outside typecode price ns drawsr drawsp refillable

%%%%%%Preliminaries
sigmap=exp(beta(end-1))*price;    %Price * sigma_p (coefficient on
                                  %unobserved characteristic).

sigmar=exp(beta(end))*refillable; %Refillable dummy * sigma_r (coefficient
                                  %on unobserved characteristic).

D=X*beta(1:(end-2),1);            %X*beta (excluding unobservables).

P=zeros(N,1);                     %Defining vector of probabilities.

for i=1:N,

%Identifying trip in data matrix
t1=choiceset(i,1);

t2=choiceset(i,2);

hK=t2-t1+1; %Choice set cardinality



%Auxiliary variables
houtside=outside(t1:t2,:);   %Indicator for whether the customer chose
                             %the outside option.

hprice=sigmap(t1:t2,:);      %Price * sigma_p (coefficient on unobserved
                             %characteristic).

hrefillable=sigmar(t1:t2,:); %Refillable dummy * sigma_r (coefficient on
                             % unobserved characteristic).

dk=D(t1:t2,:)*ones(1,ns);    %Replicating mean utility vector Xb
                             %for all draws.

d=dk-(hprice*ones(1,ns)).*(ones(hK,1)*drawsp)+ ...
    (hrefillable*ones(1,ns)).*(ones(hK,1)*drawsr);
                             %Adding the random components of price and
                             %refillable coefficients to mean utility
                             %Xbeta.

V1=max(d,[],1);              %Computing highest indirect utility across
                             %products for each draw.

V=exp(V1);                   %Exponential of highest indirect utility
                             %across products for each draw.

Pi=(exp(d(hK,:))==V);        %Indicator for whether hK-th option of choice
                             %set is the one that gives the individual
                             %the highest indirect utility.
```

```
                             %Note: Data is sorted in such a way that if
                             %an inside option is chosen, it is placed in
                             %the last position of the choice set
                             %(i.e., in the hK-th position).

OutW=1+sum(V);               %Denominator of the logit probability of
                             %whether to purchase the preferred inside
                             %option.

Pn=V./OutW;                  %Numerator of the logit probability of
                             %whether to purchase the preferred inside
                             %option.



%Defining probability
if houtside(hK)==0              %If outside option was not chosen,

P(i,1)=mean(Pi,2).*mean(Pn,2);  %Probability of choosing preferred inside
                                %option * probability of hK-th option of
                                %the choice set being the preferred option.
                                %(see equation 3 in paper).

else                            %If outside option was chosen,

P(i,1)=mean(1./OutW,2);         %Probability of purchasing outside option.
end
end


%%%%%%Likelihood function
L=-ones(1,N)*log(P)/N;


end
```

Published with MATLAB® 7.11

Appendix C: Welfare calculations

In this appendix, I discuss the code that was used for the welfare calculations in three steps. First, I discuss the data that was used for these calculations. Second, I outline the functioning of the code. Lastly, I make line-by-line comments to the MATLAB m-files used for the estimation.

Data organization

The welfare calculations make use of two data sets. First, it makes use of the data set that lists all customers and their characteristics, which is called clients.csv. The number of rows in the clients.csv file is equal to the number of customers in the sample. Table 9 displays three rows of this data set. The variables included are the average expenditure per trip, age, and trip frequency.

The second data set is chars.csv, which lists all products and the characteristics of these products. This data set is organized just like Table 1.

Code

The welfare calculations are performed using four files: welfare.m, profit.m, sharenewF.m, and dwelfarenew.m. The first file, welfare.m, loads the data, defines the variables, and calls the rest of the files to i) find the optimal prices before the removal of refillables, ii) find the optimal prices after the removal of refillables, iii) calculate welfare changes with the removal of refillables, and iv) compute the sensitivity of the welfare changes to the distribution of preferences.

sharenewF.m provides the market shares for a given vector of prices, which are computed using the choice probabilities defined in Equation 3. The functioning of sharenewF.m is analogous to that of the function lik.m discussed above. profit.m evaluates the profit function for a given vector of prices. dwelfarenew.m computes the change in customer welfare with the removal of refillables, as defined in Equation 4. dwelfarenew.m makes use of the optimal prices before and after the removal of refillables.

In what follows, I provide a line-by-line description of all four files.

Table 9: Customer characteristics: clients.csv matrix

Variable:	Customer code	Mean Expenditure	Trip frequency	Age	Refillable intensity
Column number	1	2	3	4	5
(in datacons matrix)					
	1005910	47.46	0.0789	26	0.5000
	1005924	60.96	0.0541	41	0.0000
	1006001	13.99	0.5231	22	0.1429

```
%%%%%% File name: welfare.m


%%%%%% Description: This file performs the welfare calculations
%%%%%% associated to the removal of refillables.


clear
clear global
format long

global alpha gamma beta upc nprods sizec refillable whprice nind ...
    income tshare age iniprice drawsp drawsr ns

%%%%%%Estimates
beta=[-8.153113663557281
      -2.064620430370368
       0.014952442304718
       0.010739100726440
      -1.003846370385680
       0.709225561634593
      -0.010430475443421
      -0.003731087040642
       0.706286285897152
      -0.040696468598258   %Un-comment for Diet Coke
    %  0.176469009991388   %Un-comment for Coke
    % -0.194002674519435   %Un-comment for Coke Zero
      -0.003258700797337
       1.601512985823921
       1.007093176477466];

betat=beta;
betar=beta;
betar(5)=-10000000;       %Modified vector of estimates that 'eliminates'
                          %refillables from the consideration of customers.


%%%%%%Loading data

%Draws
load draws %draws is the file with variables: drawsp, ns, drawsr.
           %ns: Number of draws from random variables for numerical
           %integration.
           %drawsp: Draws from a N(0,1).
           %drawsr: Draws from a N(0,1).



%Product characteristics
datachars=importdata('chars.csv'); %Replace filename to
                                   %i) chars_zero.csv for Coke Zero,
                                   %ii) chars_regular.csv for Coke Zero.

nprods=size(datachars,1);          %Number of options in choice set.

iniprice=datachars(:,3);           %Median price in sample.

whprice=datachars(:,2);            %Median wholesale price in sample.

sizec=datachars(:,5)/1000;         %Container size in liters.
```

```
refillable=datachars(:,4);          %Refillable container dummy variable.

upc=datachars(:,6);                 %Brand code.


%Customer characteristics
datacons=importdata('clients.csv');
nind=max(size(datacons));          %Number of customers

income=datacons(:,2)/483.78;       %Mean trip expenditure in USD

tshare=datacons(:,3);              %Trip frequency

age=datacons(:,4);                 %Age

retintensity=datacons(:,5);        %Refillable purchase frequency


%%%%%%Price and refillable coefficients
alpha=[ones(nind,1) tshare income age]*betat(1:4,1); %Price coefficient

gamma=[ones(nind,1) tshare income age]*betat(5:8,1); %Refillable
                                                     %coefficient


%%%%%%Solving for prices after removal.

seedlist=324;
Options=optimset('Display','iter','MaxIter',300,'TolX',1e-10, ...
    'TolFun',1e-6);                      %Optimization options.

beta=betar;                              %'Eliminating' refillables.

randn('state',seedlist);
p0=.75*iniprice+[0.1*randn(nprods,1)];  %Initial price.

[pr fr]=fminsearch(@(p) profitr(p),p0,Options); %Optimization

sr=sharenewF(pr);                        %Shares under optimal price vector.


%%%%%%Solving for prices before removal.
beta=betat;                              %'Including refillables.

Options=optimset('Display','iter','MaxIter',300,'TolX',1e-10, ...
    'TolFun',1e-6);                      %Optimization options.

randn('state',seedlist);
p0=iniprice+[0.1*randn(nprods,1)];      %Initial price.

[p f]=fminsearch(@(p) profit(p),p0,Options); %Optimization

ss=sharenewF(p);                         %Shares under optimal price vector.


format short
[abs(p) abs(pr) ss sr refillable sizec] %Displays prices and market shares,
                                         %before and after removal, for all
                                         %products.
```

```
%%%%%%Welfare change
%Profit change (in percent)
profit_change=[fr/f-1 ]*100;


%Customer welfare
[dw2 dwpch2]=dwelfarenew(pr,p);


[profit_change mean(dwpch2)] %Displays profit and customer welfare change
                             %(in percent).



%%%%%%Profit sensitivity to distribution of price and refillable
%%%%%%coefficients.

%Counterfactual distribution of price and refillable coefficients are
%given by (alpha, min(gamma)+rho*(gamma-min(gamma)), where rho is a
%coefficient that can take any non-negative value.

%Defining rho coefficients
Prop=0:0.05:1.2;

nProp=size(Prop,2); %Number of coefficients being used.

F_prop=zeros(nProp,1); %Declaring vector.



%Computing profits under counterfactual distributions.
for i=1:nProp;
prop=Prop(i); %Rho coefficient

gamma=min(gammaR)+(gammaR-min(gammaR))*prop; %Counterfactual distribution
                                             %of refillable coefficient.

Options=optimset('Display','final','MaxIter',300,...
    'TolX',1e-10,'TolFun',1e-6);             %Optimization options.

[p_0 f_0]=fminsearch(@(p) profit_14(p),p,Options);
                                             %Optimization.
F_prop(i,1)=f_0/f;  %Profits under counterfactual distribution /
                    %Profits with estimated distribution.
end

%Vector F_prop is used to construct Figure 6.
```

Published with MATLAB® 7.11

```
%%%%%% File name: profit.m

%%%%%% Description: This file evaluates the profit function given
%%%%%% a vector of prices.

function p=profit(price)

global whprice

share=sharenewF(price); %Computing market shares at price vector.

p=-(price-whprice)'*share; %Profits

end
```

Published with MATLAB® 7.11

```
%%%%%% File name: sharenewF.m


%%%%%% Description: This file computes the market shares given
%%%%%% a vector of prices.


function share =sharenewF(price)

global  beta nprods sizec refillable nind income tshare age ns drawsr drawsp

%%%%%%Preliminaries
CharsI=kron([ones(nind,1) tshare income age],ones(nprods,1));

CharsP1=kron(ones(nind,1),[price refillable]);

X1=[(CharsP1(:,1)*ones(1,4)).*CharsI (CharsP1(:,2)*ones(1,4)).*CharsI];
                                  %Constructing matrix of variables
                                  %included in indirect utility function.

D=(X1*beta(1:8,1))+beta(9)*kron(ones(nind,1),sizec)+beta(10);
                                  %X*beta (excluding unobservables).

sigmap=exp(beta(end-1))*price;    %Price * sigma_p (coefficient on
                                  %unobserved characteristic).

sigmar=exp(beta(end))*refillable; %Refillable dummy * sigma_r (coefficient
                                  %on unobserved characteristic).

P=zeros(nprods,nind);             %Defining vector of probabilities.


for i=1:nind,
%Identifying trip in data matrix
t1=(i-1)*nprods+1;

t2=i*nprods;

hK=nprods; %Choice set cardinality



%Auxiliary variables
dk=D(t1:t2,:)*ones(1,ns);   %Replicating mean utility vector Xb
                            %for all draws.

d=dk-(sigmap*ones(1,ns)).*(ones(hK,1)*drawsp)+...
    (sigmar*ones(1,ns)).*(ones(hK,1)*drawsr);
                            %Adding the random components of price and
                            %refillable coefficients to mean utility
                            %Xbeta.

dmax=max(d,[],1);           %Computing highest indirect utility across
                            %products, for each draw.

V=exp(dmax);                %Exponential of highest indirect utility
                            %across products for each draw.

Pi=(d==(ones(hK,1)*dmax));  %Indicator for which product in the choice
                            %set is the one that gives the individual
                            %the highest indirect utility.
```

```
OutW=1+V;                    %Denominator of the logit probability of
                             %whether to purchase the preferred inside
                             %option.

Pn=V./OutW;                  %Numerator of the logit probability of
                             %whether to purchase the preferred inside
                             %option.

%Defining probability
P(:,i)=mean(Pi,2).*(ones(hK,1)*mean(Pn,2));
                             %Probability of choosing preferred inside
                             %option * probability of product j being the
                             %preferred option.
                             %(see equation 3 in paper).

end


%%%%%%Market shares
share=sum(P,2)/nind;


end
```

Published with MATLAB® 7.11

```
%%%%%% File name: dwelfarenew.m


%%%%%% Description: This file computes how customer welfare changes when
%%%%%% moving from vector price to price1.



function [dw dwPCH] =dwelfarenew(price1,price)

global  beta nprods sizec refillable nind income tshare age ns drawsr drawsp

%%%%%%Preliminaries

%%%%%%Auxiliar variables: Under vector price
CharsI=kron([ones(nind,1) tshare income age],ones(nprods,1));

CharsP1=kron(ones(nind,1),[price refillable]);

X1=[(CharsP1(:,1)*ones(1,4)).*CharsI (CharsP1(:,2)*ones(1,4)).*CharsI];
                                  %Constructing matrix of variables
                                  %included in indirect utility function.

D=(X1*beta(1:8,1))+beta(9)*kron(ones(nind,1),sizec)+beta(10);
                                  %X*beta (excluding unobservables).

sigmap=exp(beta(end-1))*price;    %Price * sigma_p (coefficient on
                                  %unobserved characteristic).

sigmar=exp(beta(end))*refillable; %Refillable dummy * sigma_r (coefficient
                                  %on unobserved characteristic).



%%%%%%Auxiliar variables: Under vector price1
CharsP1=kron(ones(nind,1),[price1 refillable]);

X1=[(CharsP1(:,1)*ones(1,4)).*CharsI (CharsP1(:,2)*ones(1,4)).*CharsI];
                                  %Constructing matrix of variables
                                  %included in indirect utility function.

D1=(X1*beta(1:8,1))+beta(9)*kron(ones(nind,1),sizec)+beta(10);
                                  %X*beta (excluding unobservables).


%%%%%%Declaring vectors
dw=zeros(nind,1);
dwPCH=dw;



%%%%%%Computing welfare measures
for i=1:nind,
%Identifying trip in data matrix
t1=(i-1)*nprods+1;

t2=i*nprods;

hK=nprods; %Choice set cardinality

%Computing welfare measure: Under vector price
```

```
dk=D(t1:t2,:)*ones(1,ns);   %Replicating mean utility vector Xb
                            %for all draws.

d=dk-(sigmap*ones(1,ns)).*(ones(hK,1)*drawsp)+...
    (sigmar*ones(1,ns)).*(ones(hK,1)*drawsr);
                            %Adding the random components of price and
                            %refillable coefficients to mean utility
                            %Xbeta.

dmax=max(d,[],1);           %Computing highest indirect utility across
                            %products, for each draw.

V=exp(dmax);                %Exponential of highest indirect utility
                            %across products for each draw.

OutW=sum(log(1+V),2)/ns;    %Welfare measure under vector price
                            %Note: See equation 4 in paper.



%Computing welfare measure: Under vector price1
dk=D1(t1:t2,:)*ones(1,ns);  %Replicating mean utility vector Xb
                            %for all draws.

sigmap=exp(beta(end-1))*price1;
                            %Price * sigma_p (coefficient on
                            %unobserved characteristic).


d1=dk-(sigmap*ones(1,ns)).*(ones(hK,1)*drawsp)+...
    (sigmar*ones(1,ns)).*(ones(hK,1)*drawsr);
                            %Adding the random components of price and
                            %refillable coefficients to mean utility
                            %Xbeta.

dmax=max(d1,[],1);          %Computing highest indirect utility across
                            %products, for each draw.

V0=exp(dmax);               %Exponential of highest indirect utility
                            %across products for each draw.

OutW0=sum(log(1+V0),2)/ns;  %Welfare measure under vector price
                            %Note: See equation 4 in paper.



%Computing welfare changes: From price to price1
dw(i,1)=-OutW+OutW0;
dwPCH(i,1)=(-OutW+OutW0)/OutW;

end
end
```

Published with MATLAB® 7.11