Spring 2019

BUAN 6337: Predictive Analytics using SAS

A Report of

Group Assignment 4 on Crackers Data Analysis

Submitted by Group 7

Vinay Singh 2021441554
Vaibhav Shrivastava 2021434681
Megan Malisani 2021440151
Pragati Mishra 2021434655
Ishan Jain 2021426222
Erhao Liang 2021435949

Under the Guidance of

Prof. Shervin Shahrokhi Tehrani

1. Summary Statistics

There are 3292 obsevations consisting of 3 main brands with rest of them combine as Private: Sunshine, Keebler and Nabisco.

Feature Relevance:

Quantitative variable: PricePrivate, PriceNabisco, PriceKeebler and PriceSunshine .
Qualitative variable: DisplPrivate, DisplKeebler, DisplSunshine, DisplNabisco, FeatPrivate, FeatKeebler, FeatSunshin, and FeatNabisco.

Market Share:

Nabisco has the highest market share- 1792 purchases. So 54.4% market share.
Keebler has lowest market share: 226 purchases. So, 7.2% market share.

The FREQ Procedure

PRIVATE	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	2257	68.56	2257	68.56
1	1035	31.44	3292	100.00

SUNSHINE	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	3053	92.74	3053	92.74
1	239	7.26	3292	100.00

KEEBLER	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	3066	93.13	3066	93.13
1	226	6.87	3292	100.00

NABISCO	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	1500	45.57	1500	45.57
1	1792	54.43	3292	100.00

Average price of brands:

Average Price of Keenler is highest= 1.1259 and average price of private is lowest = 0.6807.

Variable	N	Mean	Std Dev	Minimum	Maximum
PRICEPRIVATE	3292	0.6807290	0.1240652	0.3800000	1.1500000
PRICESUNSHINE	3292	0.9570322	0.1329214	0.4900000	1.2900000
PRICEKEEBLER	3292	1.1259386	0.1063765	0.8800000	1.3900000
PRICENABISCO	3292	1.0792254	0.1447765	0	1.6900001

Display or store feature:

Nabisco provides display and store feature more frequently.

FeatPrivate	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	3137	95.29	3137	95.29
1	155	4.71	3292	100.00

FeatSunshine	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	3168	96.23	3168	96.23
1	124	3.77	3292	100.00

FeatKeebler	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	3152	95.75	3152	95.75
1	140	4.25	3292	100.00

FeatNabisco	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	3007	91.34	3007	91.34
1	285	8.66	3292	100.00

DisplPrivate	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	2967	90.13	2967	90.13
1	325	9.87	3292	100.00

DisplSunshine	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	2868	87.12	2868	87.12
1	424	12.88	3292	100.00

DisplKeebler	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	2942	89.37	2942	89.37
1	350	10.63	3292	100.00

DisplNabisco	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	2172	65.98	2172	65.98
1	1120	34.02	3292	100.00

2. Sampling Dataset

The SAS System

The SURVEYSELECT Procedure

Selection Method	Simple Random Sampling

Input Data Set	CRACKERS
Random Number Seed	2
Sampling Rate	0.8
Sample Size	2634
Selection Probability	0.800122
Sampling Weight	0
Output Data Set	CRACKERS_SAMPLED

3. General Utility Model Equation

Allowing for the effect of price, display, and feature on utility to vary across brands, the utility equations are:

$$U_{ip} = \beta_p + \beta \quad price_{ip} + \beta_{p2} display_{ip} + \beta_{p3} feature_{ip} + \beta_{p4} price_{ip} * feature_{ip} + \epsilon_{ip}$$

$$U_{is} = \beta_s + \beta \quad price_{is} + \beta_{s2} display_{is} + \beta_{s3} feature_{is} + \beta_{s4} price_{is} * feature_{is} + \epsilon_{is}$$

$$U_{ik} = \beta_k + \beta \quad price_{ik} + \beta_{k2} display_{ik} + \beta_{k3} feature_{ik} + \beta_{k4} price_{ik} * feature_{ik} + \epsilon_{ik}$$

$$U_{in} = \beta_n + \beta \quad price_{in} + \beta_{n2} display_{np} + \beta_{n3} feature_{in} + \beta_{n4} price_{in} * feature_{in} + \epsilon_{in}$$

4. Formatted Data set

The data is not formatted as needed for using PROC LOGISTIC or PROC MDC. For each observation (purchase event for a single individual), each potential choice (brand) in the choice set should have its own row, as shown here:

Total rows: 13168 Total columns: 6

	id	brand	brand_chosen	price	feature	displ
1	1	1	0	0.709999979	0	0
2	1	2	0	0.980000019	0	0
3	1	3	0	0.879999995	0	0
4	1	4	1	1.199999928	0	0
5	2	1	0	0.709999979	0	0
6	2	2	0	0.99000001	0	0
7	2	3	0	1.089999914	0	0
8	2	4	1	0.99000001	0	0
9	3	1	0	0.779999912	0	0
10	3	2	1	0.49000001	0	1
11	3	3	0	1.089999914	0	0
12	3	4	0	1.089999914	0	0

5. Multi-Logit Model using Logistic

We have used LOGISTIC to build a multi-logit model and have used Price, Feature, Display metrics along with a interaction variable of Price * Feature.

<u>Model-Fit Statistics</u>

Model Fit Statistics		
Criterion	Without Covariates	With Covariates
AIC	7302.999	5301.795
SC	7302.999	5417.996
-2 Log L	7302.999	5269.795

<u>Model performance compared to the null model</u>

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	2033.2040	16	<.0001
Score	1931.3418	16	<.0001
Wald	1316.8636	16	<.0001

Parameter Estimates

- We can see from the Logit model that result for the Intercept of Nabisco, Sunshine are significant with the intercept 1.6671 and -0.8448 respectively. It is insignificant for Keebler at Pr>0.0085 level.
- Price is a significant variable in the model and has an intercept of -2.9949 for all the crackers
- Feature variable is only significant for Nabisco brand with intercept of 8.3761.
- All the brand variables if they are on display are insignificant Pr>>0.0001. The intercept for Keebler, Nabisco, Sunshine and Private cracker while on display are 9.1745, 8.37, 1.44 and 2.12 respectively
- If the interaction of Price and Feature takes place, then the results are only consistent for Nabisco brand with the intercept of -7.4445

Analysis of Conditional Maximum Likelihood Estimates

Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
cracker_choice	Keebler	1	-0.3728	0.1417	6.9200	0.0085
cracker_choice	Nabisco	1	1.6671	0.1168	203.6067	<.0001
cracker_choice	Sunshine	1	-0.8448	0.1170	52.1311	<.0001
cracker_choice	Private	0	0	.	.	.
price		1	-2.9949	0.2425	152.5672	<.0001
feature*cracker_choi	Keebler	1	9.1745	3.8512	5.6751	0.0172
feature*cracker_choi	Nabisco	1	8.3761	1.9827	17.8464	<.0001
feature*cracker_choi	Sunshine	1	1.4477	1.2261	1.3941	0.2377
feature*cracker_choi	Private	1	2.1290	2.0448	1.0840	0.2978
displ*cracker_choice	Keebler	1	0.3339	0.2361	2.0005	0.1573
displ*cracker_choice	Nabisco	1	0.1294	0.0880	2.1652	0.1412
displ*cracker_choice	Sunshine	1	0.4357	0.1890	5.3165	0.0211
displ*cracker_choice	Private	1	-0.2728	0.1724	2.5036	0.1136
price*featur*cracker	Keebler	1	-8.3964	3.8483	4.7603	0.0291
price*featur*cracker	Nabisco	1	-7.4445	1.8708	15.8350	<.0001
price*featur*cracker	Sunshine	1	-0.9238	1.7107	0.2916	0.5892
price*featur*cracker	Private	1	-3.7874	3.9808	0.9052	0.3414

Explanation of Parameter Estimates

1
- Keebler and Sunshine generate lower Utility than Private cracker if everything else is same
- Nabisco generate higher Utility than Private crackers if everything else is same

2 If price increases, the purchase probabilities will drop for all the brands

3 If there is a feature for a product in store, the purchase probability will increase, Keebler store feature has the strongest effect than Nabisco, Sunshine and Private cracker brands

4 If there is a display for a product in store, the purchase probability will increase for all the brand except for Private Cracker. Keebler has the strongest effect of Store Display when compared to other brands.

5 Interaction effect of Price and Feature indicate that the purchase probability will decrease if the price of cracker is increased and is on feature for each particular crackers brand

Analysis of Conditional Maximum Likelihood Estimates

	Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
1	cracker_choice	Keebler	1	-0.3728	0.1417	6.9200	0.0085
	cracker_choice	Nabisco	1	1.6671	0.1168	203.6067	<.0001
	cracker_choice	Sunshine	1	-0.8448	0.1170	52.1311	<.0001
	cracker_choice	Private	0	0	.	.	.
2	price		1	-2.9949	0.2425	152.5672	<.0001
3	feature*cracker_choi	Keebler	1	9.1745	3.8512	5.6751	0.0172
	feature*cracker_choi	Nabisco	1	8.3761	1.9827	17.8464	<.0001
	feature*cracker_choi	Sunshine	1	1.4477	1.2261	1.3941	0.2377
	feature*cracker_choi	Private	1	2.1290	2.0448	1.0840	0.2978
4	displ*cracker_choice	Keebler	1	0.3339	0.2361	2.0005	0.1573
	displ*cracker_choice	Nabisco	1	0.1294	0.0880	2.1652	0.1412
	displ*cracker_choice	Sunshine	1	0.4357	0.1890	5.3165	0.0211
	displ*cracker_choice	Private	1	-0.2728	0.1724	2.5036	0.1136
5	price*featur*cracker	Keebler	1	-8.3964	3.8483	4.7603	0.0291
	price*featur*cracker	Nabisco	1	-7.4445	1.8708	15.8350	<.0001
	price*featur*cracker	Sunshine	1	-0.9238	1.7107	0.2916	0.5892
	price*featur*cracker	Private	1	-3.7874	3.9808	0.9052	0.3414

6. Multi Logit Model using PROC MDC

Parameter Estimates using MDC commands

We have reproduced the above Multi Logit model using PROC MDC and the result are consistent with the PROC LOGISTIC command.

Logit Model using MDC commands

The MDC Procedure

Conditional Logit Estimates

Parameter Estimates						
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Parameter Label
CRACKER_CHOICEPrivate	0	0	0			
CRACKER_CHOICESunshine	1	-0.8448	0.1170	-7.22	<.0001	
CRACKER_CHOICEKeebler	1	-0.3728	0.1417	-2.63	0.0085	
CRACKER_CHOICENabisco	1	1.6671	0.1168	14.27	<.0001	
price	1	-2.9949	0.2425	-12.35	<.0001	
CRACKER_CHOICEPrivateFEATURE	1	2.1290	2.0448	1.04	0.2978	
CRACKER_CHOICESunshineFEATURE	1	1.4477	1.2261	1.18	0.2377	
CRACKER_CHOICEKeeblerFEATURE	1	9.1745	3.8512	2.38	0.0172	
CRACKER_CHOICENabiscoFEATURE	1	8.3761	1.9827	4.22	<.0001	
CRACKER_CHOICEPrivateDISPL	1	-0.2728	0.1724	-1.58	0.1136	
CRACKER_CHOICESunshineDISPL	1	0.4357	0.1890	2.31	0.0211	
CRACKER_CHOICEKeeblerDISPL	1	0.3339	0.2361	1.41	0.1573	
CRACKER_CHOICENabiscoDISPL	1	0.1294	0.0880	1.47	0.1412	
CRACKER_CHOICEPrivate2	1	-3.7874	3.9808	-0.95	0.3414	
CRACKER_CHOICESunshine2	1	-0.9238	1.7107	-0.54	0.5892	
CRACKER_CHOICEKeebler2	1	-8.3964	3.8483	-2.18	0.0291	
CRACKER_CHOICENabisco2	1	-7.4445	1.8708	-3.98	<.0001	
Restrict1	1	5.0749E-7	12.2611	0.00	1.0000*	Linear EC [1]

<u>Goodness of Fit measures</u>

Model Fit Summary	
Dependent Variable	brand_chosen
Number of Observations	2634
Number of Cases	10536
Log Likelihood	-2635
Log Likelihood Null (LogL(0))	-3651
Maximum Absolute Gradient	7.28365E-7
Number of Iterations	5
Optimization Method	Newton-Raphson
AIC	5302
Schwarz Criterion	5396

Goodness-of-Fit Measures		
Measure	Value	Formula
Likelihood Ratio (R)	2033.2	2 * (LogL - LogL0)
Upper Bound of R (U)	7303	- 2 * LogL0
Aldrich-Nelson	0.4356	R / (R+N)
Cragg-Uhler 1	0.5379	1 - exp(-R/N)
Cragg-Uhler 2	0.5737	(1-exp(-R/N)) / (1-exp(-U/N))
Estrella	0.5953	1 - (1-R/U)^(U/N)
Adjusted Estrella	0.588	1 - ((LogL-K)/LogL0)^(-2/N*LogL0)
McFadden's LRI	0.2784	R / U
Veall-Zimmermann	0.5928	(R * (U+N)) / (U * (R+N))
N = # of observations, K = # of regressors		

7. PROBIT Model using PROC MDC

Model Fit Statistics

- While comparing the Model Fit summary, we see that PROBIT model perform better than Logit Model since it has less AIC, though it is very minute difference

PROBIT

Model Fit Summary	
Dependent Variable	brand_chosen
Number of Observations	2634
Number of Cases	10536
Log Likelihood	-2635
Log Likelihood Null (LogL(0))	-3651
Maximum Absolute Gradient	7.28365E-7
Number of Iterations	5
Optimization Method	Newton-Raphson
AIC	5302
Schwarz Criterion	5396

Ques 7

LOGIT

Model Fit Summary	
Dependent Variable	brand_chosen
Number of Observations	2634
Number of Cases	10536
Log Likelihood	-2632
Log Likelihood Null (LogL(0))	-3651
Maximum Absolute Gradient	0.06861
Number of Iterations	90
Optimization Method	Dual Quasi-Newton
AIC	5306
Schwarz Criterion	5430
Number of Simulations	100
Starting Point of Halton Sequence	11

Ques 6

Goodness of Fit Measures

Goodness-of-Fit Measures

Measure	Value	Formula
Likelihood Ratio (R)	2038.9	2 * (LogL - LogL0)
Upper Bound of R (U)	7303	- 2 * LogL0
Aldrich-Nelson	0.4363	R / (R+N)
Cragg-Uhler 1	0.5389	1 - exp(-R/N)
Cragg-Uhler 2	0.5748	(1-exp(-R/N)) / (1-exp(-U/N))
Estrella	0.5965	1 - (1-R/U)^(U/N)
Adjusted Estrella	0.5893	1 - ((LogL-K)/LogL0)^(-2/N*LogL0)
McFadden's LRI	0.2792	R / U
Veall-Zimmermann	0.5937	(R * (U+N)) / (U * (R+N))
N = # of observations, K = # of regressors		

Ques 7

Goodness-of-Fit Measures

Measure	Value	Formula
Likelihood Ratio (R)	2033.2	2 * (LogL - LogL0)
Upper Bound of R (U)	7303	- 2 * LogL0
Aldrich-Nelson	0.4356	R / (R+N)
Cragg-Uhler 1	0.5379	1 - exp(-R/N)
Cragg-Uhler 2	0.5737	(1-exp(-R/N)) / (1-exp(-U/N))
Estrella	0.5953	1 - (1-R/U)^(U/N)
Adjusted Estrella	0.588	1 - ((LogL-K)/LogL0)^(-2/N*LogL0)
McFadden's LRI	0.2784	R / U
Veall-Zimmermann	0.5928	(R * (U+N)) / (U * (R+N))
N = # of observations, K = # of regressors		

Ques 6

Like-hood Ratio are different in two results, Probit model is better than Multi-logit model on goodness-of-fit.

Difference: Logit vs Probit model

- Multi-Logit model suffers from IIA issue to capture asymmetric switching patterns
- Estimates under multinomial PROBIT model are different from conditional LOGIT model since all the intercept are bit increased for Probit Model
- In Logit model, five significant variables are in the result, which are Intercept for Sunshine& Nabisco, price, Nabisco when featured and when Nabisco when featured and price is increased.
- However, we can see in the Probit model, only Nabisco and price are significant parameter.
- Estimates for Intercept of Keebler brand has a positive effect compared to -ve intercept in Logit model
- Other Estimates of PROBIT model have consistent sign but increased value

PROBIT Model using MDC commands

The MDC Procedure

Multinomial Probit Estimates

Parameter Estimates						
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|	Parameter Label
CRACKER_CHOICEPrivate	0	0	0			
CRACKER_CHOICESunshine	1	-0.4672	0.3484	-1.34	0.1799	
CRACKER_CHOICEKeebler	1	0.2625	0.3673	0.71	0.4747	
CRACKER_CHOICENabisco	1	1.6545	0.3006	5.50	<.0001	
price	1	-2.6182	0.4121	-6.35	<.0001	
CRACKER_CHOICEPrivateFEATURE	1	1.7677	1.9531	0.91	0.3654	
CRACKER_CHOICESunshineFEATURE	1	1.1725	1.1719	1.00	0.3171	
CRACKER_CHOICEKeeblerFEATURE	1	4.6624	2.6488	1.76	0.0784	
CRACKER_CHOICENabiscoFEATURE	1	6.8948	1.9342	3.56	0.0004	
CRACKER_CHOICEPrivateDISPL	1	-0.2656	0.1819	-1.46	0.1442	
CRACKER_CHOICESunshineDISPL	1	0.2250	0.1668	1.35	0.1774	
CRACKER_CHOICEKeeblerDISPL	1	0.1617	0.1529	1.06	0.2903	
CRACKER_CHOICENabiscoDISPL	1	0.0908	0.0790	1.15	0.2507	
CRACKER_CHOICEPrivate2	1	-2.9178	3.7479	-0.78	0.4363	
CRACKER_CHOICESunshine2	1	-0.8477	1.5730	-0.54	0.5900	
CRACKER_CHOICEKeebler2	1	-4.3079	2.6163	-1.65	0.0996	
CRACKER_CHOICENabisco2	1	-6.1862	1.7958	-3.44	0.0006	
STD_1	1	1.4471	0.3864	3.75	0.0002	
STD_2	1	1.1337	0.3654	3.10	0.0019	
RHO_21	1	-0.4053	0.5582	-0.73	0.4678	
RHO_31	1	-0.0865	0.4985	-0.17	0.8623	
RHO_32	1	0.4856	0.2121	2.29	0.0220	
Restrict1	1	0.0137	.	.	.*	Linear EC [1]

8. Probit Logit Model Selection

- We can compare the likelihood among four models below, which is indicating the goodness-of-fit. The highest one is Probit model in question 7, which doesn't have any RESTRICT when coding.
- Yes, there is IIA property associated with the Logit Model since the intercept are not consistent for Logit model when compared to all the other model shown below.

PROBIT

Goodness-of-Fit Measures		
Measure	Value	Formula
Likelihood Ratio (R)	2038.9	2 * (LogL - LogL0)
Upper Bound of R (U)	7303	- 2 * LogL0
Aldrich-Nelson	0.4363	R / (R+N)
Cragg-Uhler 1	0.5389	1 - exp(-R/N)
Cragg-Uhler 2	0.5748	(1-exp(-R/N)) / (1-exp(-U/N))
Estrella	0.5965	1 - (1-R/U)^(U/N)
Adjusted Estrella	0.5893	1 - ((LogL-K)/LogL0)^(-2/N*LogL0)
McFadden's LRI	0.2792	R / U
Veall-Zimmermann	0.5937	(R * (U+N)) / (U * (R+N))
N = # of observations, K = # of regressors		

Result in question 7

Goodness-of-Fit Measures		
Measure	Value	Formula
Likelihood Ratio (R)	2037.2	2 * (LogL - LogL0)
Upper Bound of R (U)	7303	- 2 * LogL0
Aldrich-Nelson	0.4361	R / (R+N)
Cragg-Uhler 1	0.5386	1 - exp(-R/N)
Cragg-Uhler 2	0.5745	(1-exp(-R/N)) / (1-exp(-U/N))
Estrella	0.5962	1 - (1-R/U)^(U/N)
Adjusted Estrella	0.5889	1 - ((LogL-K)/LogL0)^(-2/N*LogL0)
McFadden's LRI	0.279	R / U
Veall-Zimmermann	0.5934	(R * (U+N)) / (U * (R+N))
N = # of observations, K = # of regressors		

Result in question 8 - I

Goodness-of-Fit Measures		
Measure	Value	Formula
Likelihood Ratio (R)	2035.1	2 * (LogL - LogL0)
Upper Bound of R (U)	7303	- 2 * LogL0
Aldrich-Nelson	0.4359	R / (R+N)
Cragg-Uhler 1	0.5382	1 - exp(-R/N)
Cragg-Uhler 2	0.5741	(1-exp(-R/N)) / (1-exp(-U/N))
Estrella	0.5957	1 - (1-R/U)^(U/N)
Adjusted Estrella	0.5884	1 - ((LogL-K)/LogL0)^(-2/N*LogL0)
McFadden's LRI	0.2787	R / U
Veall-Zimmermann	0.5931	(R * (U+N)) / (U * (R+N))
N = # of observations, K = # of regressors		

Result in question 8 - II

Goodness-of-Fit Measures		
Measure	Value	Formula
Likelihood Ratio (R)	2030.8	2 * (LogL - LogL0)
Upper Bound of R (U)	7303	- 2 * LogL0
Aldrich-Nelson	0.4353	R / (R+N)
Cragg-Uhler 1	0.5375	1 - exp(-R/N)
Cragg-Uhler 2	0.5733	(1-exp(-R/N)) / (1-exp(-U/N))
Estrella	0.5948	1 - (1-R/U)^(U/N)
Adjusted Estrella	0.5875	1 - ((LogL-K)/LogL0)^(-2/N*LogL0)
McFadden's LRI	0.2781	R / U
Veall-Zimmermann	0.5924	(R * (U+N)) / (U * (R+N))
N = # of observations, K = # of regressors		

Result in question 8 - III

9. Analysis for Nabisco

Normal Model

There are 418 Nabisco customer who predicted probability of more than 0.5 in the test dataset.

Nabisco Market Share = 418 /2632 = 15.88%

Ist Model: 10% Price Decrease

We decreased the Price of Nabisco by 10% and re analyzed the Market Share of the model.
Market Share = 548/2632 = 20 .8 %

There is increase of 4.92% market share using strategy 1.

2nd Model: Always on Feature

We put Nabisco always on feature and re-analyzed the Market Share of the model.
Market Share = 655/2632 = 24 .8 %

There is increase of 9% market share using strategy 2.

Conclusion: Since the Market Share increase more if we put Nabisco on Feature by 9%. Therefore, the brand Manager should implement second strategy.

Note: see code below

10. Conclusion:

Dataset has 3 main brands with all other brands combines as private. Dataset has (i)price of each 4 categories, (ii) display and store feature. By analyzing dataset, Nabisco has highest market share, and display and store feature have been used more frequently for Nabisco.

As per Logit model, price is significant for all category. Feature is significant only for Nabisco, display feature is insignificant for all brand and price-feature factor is significant for Nabisco only.

So, feature has positive effect on sales of Nabisco. Price has negative effect on sales of all brand.

As a brand manager of Nabisco, we would recommend decreasing price, and increasing store feature and price-feature factor to increase market share.

As a brand manager of Keeble, we would recommend decreasing price, increasing feature and display.

As a brand manager of Sunshine, we would recommend decreasing price, increasing feature and display.

As a brand manager of Private, we would recommend decreasing price, increasing feature and decrease display.

APPENDIX

```
/*Homework:4*/

LIBNAME HW4 'H:\My SAS Files\HW4';

PROC IMPORT OUT= HW4.Crackers
            DATAFILE= "H:\My SAS Files\HW4\crackers.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;

DATA Crackers;
SET HW4.Crackers;
RUN;

/*1.  Based on the previous lectures, provide a professional summary statistic of this
dataset.
Your summary should briefly explain the status of crackers brands in the market, based on
the above dataset?
For example: which brand has the highest market share? Is there any difference between
the average prices of these brands?
Which brand does provide display or feature in store more frequently? And, …  */

proc freq data=Crackers;
tables PRIVATE SUNSHINE KEEBLER NABISCO;
run;

PROC MEANS DATA=Crackers;
VAR PRICEPRIVATE PRICESUNSHINE PRICEKEEBLER PRICENABISCO;
RUN;

proc freq data=Crackers;
tables FeatPrivate FeatSunshine FeatKeebler FeatNabisco DisplPrivate DisplSunshine
DisplKeebler DisplNabisco;
run;
```

```
/*2*/

proc surveyselect
data= Crackers
out=crackers_sampled
outall samprate=0.8
seed=2;
run;


data crackers_training crackers_test;
set crackers_sampled;
if selected=1 then
output crackers_training;
else
output crackers_test;
run;
/*3*//*no code*

//*4*//*need numeric choice in order to convert format*/
data crackers_training;

set crackers_training;

if (Private=1) & (Sunshine=0) & (Keebler=0) & (Nabisco=0) then Choice=1;
if (Private=0) & (Sunshine=1) & (Keebler=0) & (Nabisco=0) then Choice=2;
if (Private=0) & (Sunshine=0) & (Keebler=1) & (Nabisco=0) then Choice=3;
if (Private=0) & (Sunshine=0) & (Keebler=0) & (Nabisco=1) then Choice=4;

run;

data crackers_test;


set crackers_test;

if (Private=1) & (Sunshine=0) & (Keebler=0) & (Nabisco=0) then Choice=1;
if (Private=0) & (Sunshine=1) & (Keebler=0) & (Nabisco=0) then Choice=2;
if (Private=0) & (Sunshine=0) & (Keebler=1) & (Nabisco=0) then Choice=3;
if (Private=0) & (Sunshine=0) & (Keebler=0) & (Nabisco=1) then Choice=4;

run;

/*converting data*/

data crackers_training(keep=id brand brand_chosen price feature displ);
set crackers_training;
array price_vector{4} priceprivate pricesunshine pricekeebler pricenabisco;
array feature_vector{4} featprivate featsunshine featkeebler featnabisco;
array display_vector{4} displprivate displsunshine displkeebler displnabisco;
retain id 0;
id + 1;
do i=1 to 4;
brand=i;
brand_chosen=(choice=i);
price=price_vector{i};
```

```
feature=feature_vector{i};
displ=display_vector{i};
output;
end;
run;

data crackers_test(keep=id brand brand_chosen price feature displ);
set crackers_test;
array price_vector{4} priceprivate pricesunshine pricekeebler pricenabisco;
array feature_vector{4} featprivate featsunshine featkeebler featnabisco;
array display_vector{4} displprivate displsunshine displkeebler displnabisco;
retain id 0;
id + 1;
do i=1 to 4;
brand=i;
brand_chosen=(choice=i);
price=price_vector{i};
feature=feature_vector{i};
displ=display_vector{i};
output;
end;
run;


data crackers_training;

set crackers_training;

if brand='1' then cracker_choice='Private ';
if brand='2' then cracker_choice='Sunshine';
if brand='3' then cracker_choice='Keebler';
if brand='4' then cracker_choice='Nabisco';

drop brand;

run;


data crackers_test;

set crackers_test;

if brand='1' then cracker_choice='Private ';
if brand='2' then cracker_choice='Sunshine';
if brand='3' then cracker_choice='Keebler';
if brand='4' then cracker_choice='Nabisco';

drop brand;

run;


/* Ques 5: Estimate the logit model on the training sample using PROC LOGISTIC and report
the estimation results.
Please set the Private intercept equal to zero as the based. You need to report model
parameters, significance.
Also, provide bullet points on the meaning of the estimations results in terms of
consumers' utility by purchasing a brand. */
```

```
proc logistic data=crackers_training;
 strata id;
 class cracker_choice (ref = 'Private') / param=glm;
 model brand_chosen (event='1') = cracker_choice price cracker_choice*feature
cracker_choice*displ cracker_choice*price*feature;

 title 'Logit Model using Logistic commands';

run;

/*Ques 6: Reproduce your results in Q5 using PROC MDC (HINT: See the SAS code posted for
the lecture for examples of replicating the
results with PROC MDC. You can use the same dataset format. Refer to the SAS manual for
more details about PROC MDC.
You will need to use "type=clogit" to estimate a multinomial model, and "nchoice=4" to
indicate there are four alternatives for each
choice occasion.


In PROC MDC, using the CLASS statement for a categorical variable with N levels will
create N dummy
variables, each for one level of the categorical variable. Use the restrict statement to
set the coefficient for one of the dummy variables
to zero - effectively omitting this dummy variable. You will need to do this for the main
effects and any interaction effects that
involve the variable used in the CLASS statement - refer to the SAS code for the lecture
for an example. Also, you cannot insert
the interaction effect price ×feature directly into PROC MDC. First define a new variable
as price ×feature,  then insert it into model.

*/

data crackers_training1;
set crackers_training;
price_feature=price*feature;
run;

proc mdc data = crackers_training1;
  id id;
  class  cracker_choice;
  model brand_chosen = cracker_choice price cracker_choice*feature cracker_choice*displ
cracker_choice*price_feature /  type = clogit
nchoice = 4;
restrict cracker_choicePrivate=0; /* mdc does not allow flexible options to code class
variable. Need to explicitly force reference levels to zero */
title 'Logit Model using MDC commands';

run;



/*7.Nowuse the model in Q5, and estimate the PROBIT model using PROC MDC.
Do the estimation results look similar to Q6? Explain the differences.*/
/* Multinomial Probit with out restricting errors (it follows the basic seeting in SAS)
*/
proc mdc data = crackers_sampled_mdc;
id id;
class cracker_purchase;
```

```
model Brand_purchase =cracker_purchase price cracker_purchase*feature
cracker_purchase*displ cracker_purchase*pricefeature /  type = mprobit nchoice = 4;
restrict cracker_purchasePrivate=0;
/* mdc does not allow flexible options to code class variable. Need to explicitly force
reference levels to zero */
run;
ODS RTF close;



data crackers_training1;
set crackers_training;
price_feature=price*feature;
run;

proc mdc data = crackers_training1;
  id id;
  class  cracker_choice;
  model brand_chosen = cracker_choice price cracker_choice*feature cracker_choice*displ
cracker_choice*price_feature /  type = mprobit
nchoice = 4;
restrict cracker_choicePrivate=0; /* mdc does not allow flexible options to code class
variable. Need to explicitly force reference levels to zero */
title 'PROBIT Model using MDC commands';

run;


/*8.In this Question, you need to estimate three new versions of the PROBIT model used in
Q7.
I.Estimate the model where the error terms are normally distributed, independently and
identically.
This is equivalent that the covariance matrix is the identity matrix.II.
Estimate the model where the error terms are normally and independently distributed, but
allow for existence of heteroscedasticity.
This is equivalent that the covariance matrix is a diagonal matrix.III.Estimate the model
where the error terms are normally distributed,
there is no existence of heteroscedasticity, and the errors are correlated. This is
equivalent that the covariance matrix is a matrix
such that (1) all diagonal elements are equal to 1, and (2) there are non-zero
correlations among alternatives
(i.e., the off-diagonal elements are non-zero). (HINT: See the SAS code posted for the
lecture about PROC MDC.
Use the restrict statement to set the right parameters to zero to generate the above
models.
Note that all the above four models (Q7 and three models in Q8) are nested
model.Therefore, you can use the Loglikelihood to choose the
best model that fits into data.Now, based on your answer, is there any evidence against
IIA property in this data, which is existed by
using logit model in Q6?*//*I:Multinomial Probit allowing errors to be correlated, but
restricting them to be unit variance*/
proc mdc data = crackers_sampled_mdc;
id id;
class cracker_purchase;
model Brand_purchase =cracker_purchase price cracker_purchase*feature
cracker_purchase*displ cracker_purchase*pricefeature / type = mprobit nchoice = 4
unitvariance= (1 2 3 4);
/*diagonal elements equal to each other*/
```

```
restrict cracker_purchasePrivate=0; /* mdc does not allow flexible options to code class
variable.
Need to explicitly force reference levels to zero */
run;
ODS RTF close;/*II:multinomial Probitrestricting errors to be uncorrelated, but allowing
heteroschedasticity*/
proc mdc data = crackers_sampled_mdc;
id id;class cracker_purchase;
model Brand_purchase =cracker_purchase price cracker_purchase*feature
cracker_purchase*displ cracker_purchase*pricefeature/  type = mprobit nchoice = 4 ;
restrict cracker_purchasePrivate=0;
/* mdc does not allow flexible options to code class variable. Need to explicitly force
reference levels to zero */
restrict rho_21=0;
restrict rho_31=0;
restrict rho_32=0;
/* Allows heteroschedasticity, the variance of each predictor does not need to be the
same; error terms uncorrelated, then covariance
must be equal to zero,And because SAS automatically defaults the RHO of the bottom row to
zero (rho_41 ~ rho_43),
you only need to define the first two rows separately.*/
run;
ODS RTF close;
/*III:Multinomial Probit restricting errors to be iid*/
proc mdc data = crackers_sampled_mdc;
id id;
class cracker_purchase;
model Brand_purchase =cracker_purchaseprice cracker_purchase*feature
cracker_purchase*displ cracker_purchase*pricefeature /type = mprobit nchoice = 4
unitvariance= (1 2 3 4);
restrict cracker_purchasePrivate=0;
/* mdc does not allow flexible options to code class variable. Need to explicitly force
reference levels to zero */
restrict rho_21=0;
restrict rho_31=0;
restrict rho_32=0;
run;
ODS RTF close;
/*9.    Now re-consider your model in Q6, i.e., the logit model. First, find the
predicted probability for each brand on your test dataset.
Nabisco brand manager believes if the predicted probability of busying Nabisco is greater
than 50%, the consumer will buy it for sure.
Use the predicted probabilities and 50% threshold to find Nabisco market share based on
test dataset. */

/*Now, the Nabisco brand manager has two strategies to increase its market share. I-
reducing their price by 10% or II-having feature in store
always. Now, based on test dataset and your model in Q6, find how far the Nabisco's
market share will increase by implementing one of these two
strategies? If we assume the implementation cost of both strategies are equal, which one
will be more profitable?
(Hint: You can follow the threshold 50% to find who buys Nabisco under the above
strategies in test dataset.) */


/*Taking the training data set and adding a new column selected which is equal to 1*/
data crackers_training;
set crackers_training;
selected=1;
```

```
run;


*Taking the test data and making the field brand_chosen as null*/
data crackers_test;
set crackers_test;
brand_chosen = .;
selected=0;
price_feature=price*feature;
run;

/*Aggregating training and test data*/
data extdata;
set crackers_training crackers_test;
run;

/*Using proc mdc data and getting the predicting probabilities*/
proc mdc data = extdata;
id id;
class cracker_choice;
model brand_chosen = cracker_choice price cracker_choice*feature cracker_choice*displ
cracker_choice*price_feature /  type = clogit nchoice = 4;
restrict cracker_choicePrivate=0;
output out= probdata pred = p;
run;

/*Getting the data for Nabisco for the probability greater than 0.5*/
data probdata_nabisco;
set probdata;
where cracker_choice = 'Nabisco' and p >= 0.50 and brand_chosen=.;
run;

9.I
/*Reducing the price by 10% and creating new data set with price reduced by 10$*/
data crackers_test_price;
set crackers_test;
if cracker_choice = 4 then price = price -(0.1 * price);
run;

/*Aggregating test and training data set*/
data extdata_price;
set crackers_training1 crackers_test_price;
run;

/*Predicting probabilities for new price data set*/
proc mdc data = extdata_price;
id id;
class cracker_choice;
model brand_chosen = cracker_choice price feature displ price_feature /  type =
clogitcovest = hess  nchoice = 4;
restrict cracker_choicePrivate=0;
output out= probdata_price pred = p;
run;

/*Getting the data for Nabisco for the probability greater than 0.5*/
data probdata_nabisco;
set probdata;
where cracker_choice = 'Nabisco' and p >= 0.50 and brand_chosen=.;
run;
```

```
9.II
/*Creating new data set with feature for Nebisco*/
data crackers_test_feature;
set crackers_test;
if cracker_choice = 4 then feature = 1;
run;

/*Aggregating training and test data*/
data extdata_feature;
set crackers_training1 crackers_test_feature;
run;

/*Predicting probabilities for feature data set*/
proc mdc data = extdata_feature;
id id;
class cracker_choice;
model brand_chosen = cracker_choice price feature displ price_feature /  type =
clogitcovest = hess  nchoice = 4;
restrict cracker_choicePrivate=0;
output out= probdata_feature pred = p;
run;

/*Getting the data for Nabisco for the probability greater than 0.5*/
data probdata_nabisco;
set probdata;
where cracker_choice = 'Nabisco' and p >= 0.50 and ;
run;
```