Introduction

In this assignment, a cleansed dataset consisting of 424 respondents' information was used to construct different Hierarchical Bayes Multinomial Logit (HB-MNL) models. The intent of this assignment is to help Obee Juan, a product development manager at Star Technologies Company (STC), to estimate preference shares for two different choice scenarios that are different from the initial choice scenarios studied. The conclusion of this analysis will also contain a recommended tablet configuration. For the extra scenarios, a developed model was used to predict the alternative that would be picked by the respondents. Since STC is not in the business of the computer tablet market, it is essential that the company identify a winning combination of tablet specs and price to ensure a successful market debut.

Data Overview

Two key datasets were provided by the Neverending Marketing Insights (NMI) company. One was a choice based conjoint task plan (this consisted of 108 different combinations) and a dataset consisting of 424 respondents' responses for their chosen preference (based on the alternatives presented) as well as the respondents' demographic information. The task plan (i.e., choice set) consisted of six different attributes and a total of 36 choice sets were provided:

- brand (STC, Somesong, Pear, Gaggle)
- price ($199, $299, $399)
- screen (5", 7", 10")
- RAM (8 GB, 16 GB, 32 GB)
- processor (1.5 GHz, 2 GHz, 2.5 GHz)
- interaction between brand and price

As for the survey, the respondents were given three alternatives to pick from within each choice set. In addition, they were also asked to provide demographic information such as age, gender, income level, children, parental status, resident state, and their interest level on the following:

- purchasing a new tablet
- purchasing a new mobile phone
- using cloud storage
- taking an online course

Exploratory Data Analysis

A comprehensive exploratory data analysis was also completed (A small sample of the charts constructed can be found in the Appendix). Recall that the dataset provided for analysis was cleansed. The original dataset consisted of 1024[1] observations, however, the analyzed dataset consisted of only 424 observations (i.e., respondents). It was found that the respondents were equally distributed in terms of gender (see Table 8.1). In terms of age, the majority of the respondents were aged 25 and older (see Figure 8.1). Approximately 44% of the respondents were between the ages of 25 and 44. In terms of income (see Figure 8.2), over 55% of the respondents had an income of $75,000 or less. It was also found that approximately 65% of the respondents had at least one or more child (see Table 8.2).

[1] This value was provided in a sync session to the author of this report and cannot be found in the supporting documentation provided for this assignment

In terms of the interest in purchasing a tablet, it was found that more females were inclined[2] to purchase a tablet (see Figure 8.3). It was also discovered that respondents between the ages of 25 and 44 were more interested to purchase a tablet than older respondents (aged 55 and older). A similar conclusion was found in terms of interest in purchasing a smartphone, interest in using cloud storage. Interestingly, it was found that both genders did not have a strong interest in taking online courses (see Figure 8.4). Furthermore, older respondents were less interested in online courses than younger respondents.

Data Preprocessing

A number of steps were undertaken to prepare the datasets for analysis. This included recoding the responses with the appropriate 'dummy' variables for use in a Multinomial Logit model. Furthermore, a new set of variables were constructed to document the interaction between brand and price. Ultimately, the resultant matrices were incorporated into a nested list so that each respondent's responses could be paired up with the appropriate choice set matrix (see the section R Code in the Appendix). Table 8.3 in the Appendix documents the variables that were constructed. Note that since this is a multinomial logit problem, the following equation syntax was used:

$$E(Y) = log(\frac{p}{1-p}) = \beta_1 X_1 + \beta_2 X_2 + ... + \beta_{14} X_{14} \tag{3.1}$$

Note that β_i is the coefficient and X_i is the dummy variable and i is the variable number. For this analysis, the β coefficient values will be derived.

Since there is a need to create dummy (i.e., indicator) variables, the baseline variables will be screen5 (referring to the 5" screen), ram8 (referring to the 8 GB RAM), proc1.5 (referring to the 1.5 GHz processor), brand.stc (referring to the STC brand), and brand.price.stc (referring to the interaction between price and the STC brand). These variables have been coded as -1 in the appropriate choice set matrix (these values are also known as the preferences). This will enable the ability to derive β coefficients for these baseline variables.

Model Development

Two models were constructed: one without any covariates such as prior ownership of an STC product and another with the covariate (prior ownership of an STC product). Both of these models are HB-MNL[3] models. Furthermore, both models made use of Markov Chain Monte Carlo (MCMC) simulations leveraging 150,000 iterations and keeping every 5th simulated sample. In order to construct these models, the package "bayesm" was used and the function within that package was "rhierMnlDP()". Each model was assessed with a confusion matrix (illustrating the predicted responses versus the actual observed responses) and an area under the curve (AUC) value (assessing the accuracy in terms of sensitivity vs specificity). This discussion can be found in the Model Results & Conclusion section below.

Model 1: No covariate

The intent of the simulated values (using MCMC) is to derive the posterior β values for each predictor and each respondent. The mean of the simulated values is then used to derive the posterior β means. This in turn is then matrix-multiplied with the original choice set matrix, the product is then exponentiated, and then finally divided by the row sums to obtain the most likely (i.e., predicted) choice.

[2] The interest was based on a rating of 1 - 10 where 10 meant most interested and 1 meant least interested

[3] Hierarchical Bayes Multinomial Logistic

One of the requirements during the analysis of MCMC values is to ”make sure that the algorithm produces a Markov chain that converges to the appropriate density (the posterior density) and that mixes well throughout the support of the density” (Lynch, 2007). One way to assess mixing and convergence is to analyze the trace plot for the simulation. Due to the number of retained samples (30,000), a trace plot using the cumulative means was constructed. Note that respondent 23 was arbitrarily chosen to assess the mixing and convergence.

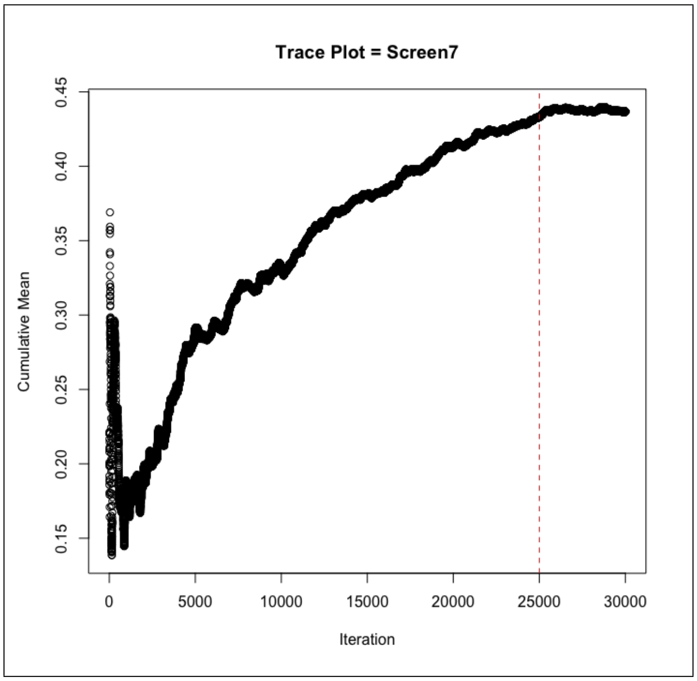


Figure 4.1: Cumulative Mean Trace Plot - Respondent 23, Screen7

In Figure 4.1, the red dotted line suggests that convergence was found around 25,000 iterations. Note how the plot enters a steady-state around the same point. Similar plots were constructed for all predictors for respondent 23. It is essential to note that not every variable converged at the same iteration (see Figure 8.5 - which seems to converge at around iteration 15,000). For the sake of brevity, the burn-in period was assumed to be 25,000 and the remaining 5,000 samples were preserved for analysis.

Typically, the average β values for each respondent was calculated from the simulated β values. Thus, each of the 424 respondents had their own respective HB-MNL model. For simplicity's sake, it was assumed that all respondents are homogeneous. Thus, the overall average β values were calculated. Table 4.1 summarizes the average β coefficient values along with the respective odds ratio.

Table 4.1: Model 1 - Overall Beta Coefficient Summary

Variable	Beta Coefficient	Odds Ratio
screen5	-0.265	0.767
screen7	-0.185	0.831
screen10	0.45	1.57
ram8	-0.731	0.482
ram16	0.0864	1.09
ram32	0.644	1.9
proc1.5	-2.35	0.0951
proc2	1.02	2.78
proc2.5	1.33	3.78
price199	2.62	13.7
price299	0.314	1.37
price399	-2.93	0.0535
brand.stc	0.424	1.53
brand.somesong	-0.202	0.817
brand.pear	0.0822	1.09
brand.gaggle	-0.304	0.738
brand.price.stc	-0.114	0.893
brand.price.somesong	0.0465	1.05
brand.price.pear	0.055	1.06
brand.price.gaggle	0.0122	1.01

The Beta Coefficients in Table 4.1 can be written out as follows (recall that this is the log odds equation):

$$\begin{aligned} E(Y) = log(\frac{p}{1-p}) &= \beta_1 * X_1 + \beta_2 * X_2 + ... + \beta_n * X_n \\ E(Y) = screen5 * X_1 + screen7 * X_2 + ... &+ brand.price.gaggle * X_20 \\ E(Y) &= -0.265 * X_1 + -0.185 * X_2 + ... + 0.0122 * X_20 \end{aligned} \tag{4.1}$$

Results in Table 4.1 can be interpreted fairly easily. For instance, screen 10 is preferred 1.57 times versus no preference. In other words, the probability of preferring a 10” screen is 157% higher than no preference. Note how dominant the price of $199 is at 13.7 times versus no preference.

Another approach is to compare two similarly categorized options together. For instance, the following equation can be used to help understand how much more (or less) a 10” screen is compared to a 5” screen.

$$\begin{aligned} Screen10 - Screen7 = 0.45 - (-0.185) = 0.635 \\ e^{0.635} = oddsratio = 1.89 \end{aligned} \tag{4.2}$$

This means that a 10” screen is preferred 1.89 times more than a 7” screen. A similar approach could be used to compare other options with respective categories.

The variables screen5, ram8, proc1.5, price199, brand.stc, brand.price.stc were not derived directly since the effects coded version of the attributes was developed in the pre-processing stage. The β values for these variables was derived by subtracting the β values for the other respective indicator variables from zero. For instance, the screen5 β coefficient value was derived using the following equation:

$$screen5 = 0 - screen7 - screen10 = 0 - (-0.185) - (0.45) \tag{4.3}$$

The interaction between brand and price is interpreted slightly differently. Essentially, the interaction between price and brand is dependent on the preference. Recall from earlier that the preference can be coded as -1 (preferring the baseline), 0 (preferring the next best choice above the baseline), and 1 (preferring the best choice). For example, if the preference is the baseline and the comparison is between the brands STC and Somesong, then the log odds can be calculated as follows:

$$\begin{aligned} logodds = brand.stc - brand.somesong + (brand.price.stc + brand.price.somesong) * preference \\ logodds = 0.424 - (-0.202) + (-0.114 - 0.0465) * -1 \\ logodds = 0.6935 \end{aligned} \tag{4.4}$$

Taking the exponential of the logodds ($exp(0.7865)$) results in and odd ratio of 2.20. In other words, the STC brand is preferred 2.2 times to Somesong if the price is at $199. Table 8.4 summarizes the odds ratio for the different brands and prices (interaction effects). Note how the brand STC is preferred at all price points to other brands. However, STC is strongly preferred at the lowest price point as compared to other brands at different pricing levels.

Model 1 predicted values can be found in the Appendix under Table 8.6. Note how the predicted number of respondents picking each alternative within each choice set is quite similar to the actual values.

Model 2: Covariate (prior ownership)

For this model, the focus was on understanding how consumer preference may be affected given that particular respondents have previously owned an STC product. Following a similar approach to what was done in Model 1, the only addition was a matrix indicating whether a respondent had owned an STC product or not. In order to maintain consistency with Model 1, this model was also run with 150,000 iterations and keeping every five samples. Furthermore, the last 5,000 samples were kept.

One interesting output from this new model is the ”Delta Draw”. This dataset essentially describes the posterior distributions of the regression coefficients mean-centered on prior ownership of an STC product. Table 4.2 illustrates the average delta-draw values for each of the variables.

Table 4.2: Average Delta Draw Values

Variable	Average
screen7	-0.0462831
screen10	-0.04571877
ram16	0.08536959
ram32	0.02684507
proc2	0.25485435
proc2.5	0.39755898
price299	-0.01887764
price399	-0.68340868
brand.somesong	-0.23349499
brand.pear	0.95924565
brand.gaggle	-0.05984522
brand.price.somesong	-0.17979772
brand.price.pear	0.05361792
brand.price.gaggle	0.16931045

Loosely put, the average values could be construed as the correlations between prior ownership of an STC product and the appropriate variables. Note how the average values for both a 7” and a 10” screen are around -0.04. This suggests that there is not much correlation between screen size and prior ownership of an STC product. On the other hand, note the stronger positive relationship between the two different types of processor and prior ownership of an STC product. Another interesting note is the strong positive relationship (0.96) of the Pear brand with prior ownership of an STC product.

Similar to Table 4.1, Table 4.3 illustrates the average coefficient values for each variable. The interpretation fo this table is made in a similar way as was done for Table 4.1. For instance, the 10" screen is preferred 1.51 times versus no preference. In other words, the probability of preferring a 10" screen is 151 times higher than no preference. Furthermore (leveraging equation 4.2), a 10" screen is preferred 1.78 times more than a 7" screen. Note that the syntax for Equation 4.1 can be used for Table 4.3.

Table 4.3: Model 2 - Overall Beta Coefficient Summary

Variable	Beta Coefficient	Odds Ratio
screen5	-0.248	0.78
screen7	-0.164	0.848
screen10	0.412	1.51
ram8	-0.671	0.511
ram16	0.0934	1.1
ram32	0.578	1.78
proc1.5	-2.33	0.0971
proc2	1.08	2.95
proc2.5	1.25	3.49
price199	2.62	13.8
price299	0.292	1.34
price399	-2.91	0.0543
brand.stc	0.413	1.51
brand.somesong	-0.23	0.795
brand.pear	0.109	1.11
brand.gaggle	-0.292	0.747
brand.price.stc	-0.0836	0.92
brand.price.somesong	0.0973	1.1
brand.price.pear	0.0147	1.01
brand.price.gaggle	-0.0284	0.972

Finally, Table 8.5 summarizes the odds ratio of the different brands and prices (interaction effects) for model 2. Note how the STC brand is preferred 2.28 times to Somesong if the price is at $199. Comparing the odds ratio of Tables 8.4 and 8.5, it is clear to see that both models are very similar in their results.

Model 2 predicted values can be found in the Appendix under Table 8.6. Note how the predicted number of respondents picking each alternative within each choice set is quite similar to the actual values as well as the predicted results from Model 1.

Model Results & Conclusion

Recall that Table 8.6 in the Appendix provides a summarization of the predicted count of respondents choosing a particular alternative for each choice set. In order to determine how well each model's fit is, two key assessments were used to assess each model: confusion matrix and an area under the curve (AUC) metric. Table 4.4 is the confusion matrix for model 1. A perfect model (which is also a good indicator of an overfitted model) will have non-zero values on diagonal and zero values everywhere else. In reality, the diagonal should have the highest numbers while the non-diagonal parts should have the least number of observations. As an example, 3,679 observations were correctly predicted to be alternative 1 given that they were actually alternative 1. However, the model also predicted that 258 were to be alternative 2 and

315 were to be alternative 3. In reality, both were actually alternative 1. The overall accuracy for the model 1 (using the confusion matrix) is 0.883 or 88.3%. The AUC for alternative 1 is about 0.866 (which is the area under the curve found in the upper left corner of figure 8.6). Overall, the multi-class AUC is at 0.9074 for model 1.

Table 4.4: Model 1: Confusion Matrix

	Actual Response		
Modeled Response	**1**	**2**	**3**
1	3679	419	215
2	234	3754	204
3	291	416	6052

Similarly, Table 4.5 is the confusion matrix for model 2 (which utilizes the covariate of prior STC device ownership). The overall accuracy of the confusion matrix is 0.872 or about 87.2% The individual class AUC's can be found in figure 8.7 in the Appendix.

Table 4.5: Model 2: Confusion Matrix

	Actual Response		
Modeled Response	**1**	**2**	**3**
1	3631	477	217
2	258	3659	226
3	315	453	6028

The multi-class AUC for model 2 is at 0.9001. The key takeaway is that both models are quite similar to each other. Model 2 (with the covariate), however, performs slightly lower (in terms of accuracy) as compared to Model 1.

Another conclusive chart is the parts worth plot (see Figure 4.2). This plot essentially illustrates the average coefficient values (seen in Table 4.1) and plots them. From this plot, it is much more evident to see which coefficients have strong preferences.

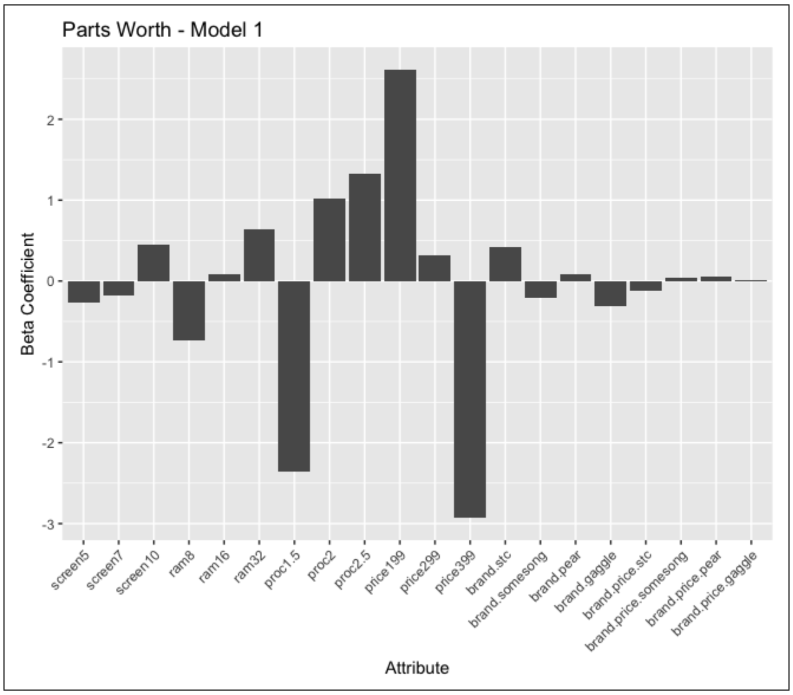


Figure 4.2: Parts Worth - Model 1

In the figure above, it is clear to see that a 10” screen is preferred along with 32 GB RAM, with either a 2 GHz or 2.5 GHz processor and a preferred price point of $199 or $299 as well as a preference for the brand STC. The aforementioned attributes could also serve as potential designs for the new tablet. The parts worth plot for model 2 can be found in the Appendix - see Figure 8.8. Note the similarity of this plot and Figure 4.2. It is also important to note how the interaction between brand and price has very little preference. In other words, price sensitivity does not necessarily vary over the brands in a meaningful way.

In order to calculate the importance of each attribute, a programmer can simply take the absolute value for each coefficient found in Table 4.1 or Table 4.3 and divide it by the sum of the absolute values of all the coefficients within the respective table. This step was not completed for this analysis, but is described here as a potential next step opportunity.

Extra Scenarios

Two additional choice sets were also provided for this analysis. Information about the additional choice sets can be found within the dataset that was provided for this analysis (and is not included in this paper for brevity’s sake). Model 1 was used to help predict which alternative most respondents would pick for both choice set. Recall that Model 1 does not make use of the covariate related to the prior ownership of an STC product.

A voting technique was used to determine the most likely alternative. In other words, the Model 1 approach was used - where 424 models were dynamically leveraged to understand how each of the individual respondents would respond. Then the most preferred alternative for each respondent was leveraged and then tallied to understand how many respondents would pick a particular alternative. This is in contrast to using the overall coefficient means that were documented in Table 4.1. Results of this voting approach can be found in Table 5.1 below.

Table 5.1: Extra Scenario Voting Results

Choice Set	Alternative 1	Alternative 2	Alternative 3
1	92	232	100
2	99	236	89

Note how the majority of the respondents would likely choose alternative 2. Alternative 2 for choice set 1 corresponds to a tablet with a 10" screen, 2 GHz processor, 32 GB RAM, $199 price, and the STC brand. On the other hand, alternative 2 for choice set 2 corresponds to a tablet with a 5" screen, 16 GB RAM, 1.5 GHz processor, $199 price, and the Gaggle brand.

Limitations & Assumptions

For this analysis, some key assumptions had to be made. One key assumption is that the population of respondents is considered to be homogeneous. We also assume that the population of respondents are like-minded and that they have a linear thought process. In reality, there is a very good likelihood that some of the thought processes are not as linear. Furthermore, during the model development phase, not every variable converged after 150,000 iterations. There is also no awareness in this dataset if the respondent was in a hurry or after a certain point was disinterested in the survey.

Another limitation of this study is the lack of training, validation, and test datasets. Typically, the respondents would be randomly split into the three different aforementioned datasets. This would help to improve the model design as well as reduce the likelihood of having an overfitted model. Another key point is the feasibility of certain configurations. For instance, manufacturing costs, research & development costs, etc. are not known and it is difficult to recommend a configuration with full confidence since its feasibility may seem impossible.

In today's competitive tablet market, STC may want to consider updating the configuration options and redoing the assessment. For instance, many modern smartphones sport a display that is 5" or more. STC may want to consider investigating tablets with a 7", 10" or 12" screen. Furthermore, STC may want to explore the operating system (OS) as well. Although Android is a popular option for 3rd party OEM vendors, STC could consider[4] going with its own OS or implementing Windows. Finally, STC may also want to explore refining their terminology regarding memory and expanding the memory size. For instance, STC referred to the disk space as RAM. As of right now, it is technically not feasible to have 32 GB RAM in a portable tablet form. Rather, STC may want to rename the term to disk space such as 32 GB SSD. Furthermore, today's tablets are sporting up to 256 GB or more. STC may want to consider upping the maximum and minimum disk spaces.

Conclusion & Next Steps

Two Hierarchical Bayes Multinomial Logit (HB-MNL) models were developed for this analysis: one utilizing the prior ownership of an STC product and one not. Both models were explored to understand price sensitivity on brand and each attribute assessed to understand the respondents' preference.

Evaluating the different beta coefficient values, it was found (in Model 1) that the interaction between brand and price was virtually negligible. Furthermore, respondents were inclined to prefer a tablet that had a 10" screen, 32 GB RAM, a 2.5 GHz processor, and a $199 price point. From the original 36 choice sets, the most popular configuration chosen consists of a 7" screen, 16 GB RAM, 2.5 GHz processor, price point of $199, and made by STC. If the two extra scenarios are also considered, the preferred configuration stays the same.

[4] Apple's iOS is not an option as it is a proprietary OS that is limited to devices engineered and produced by Apple only.

When utilizing the prior ownership of an STC product (i.e., the covariate), similar results were found. However, when looking at the average delta draw values, it was interesting to note that the respondents (who had owned an STC product previously) preferred the brand Pear. This is a good indicator that STC may wish to invest in advertising and marketing strategies to help convince consumers on the values of owning and using an STC branded tablet versus one from Pear.

As a next step, STC may want to look at gender and household income as covariates. This would enable a deeper understanding whether different genders prefer different types of configurations. Furthermore, it will also help STC understand how different household income levels could affect respondent proclivities. On the other hand, STC could also investigate the development of two different types of models: one that focuses on respondents that have had a prior STC product and another model that focuses on respondents that do not and have never owned an STC product.

References

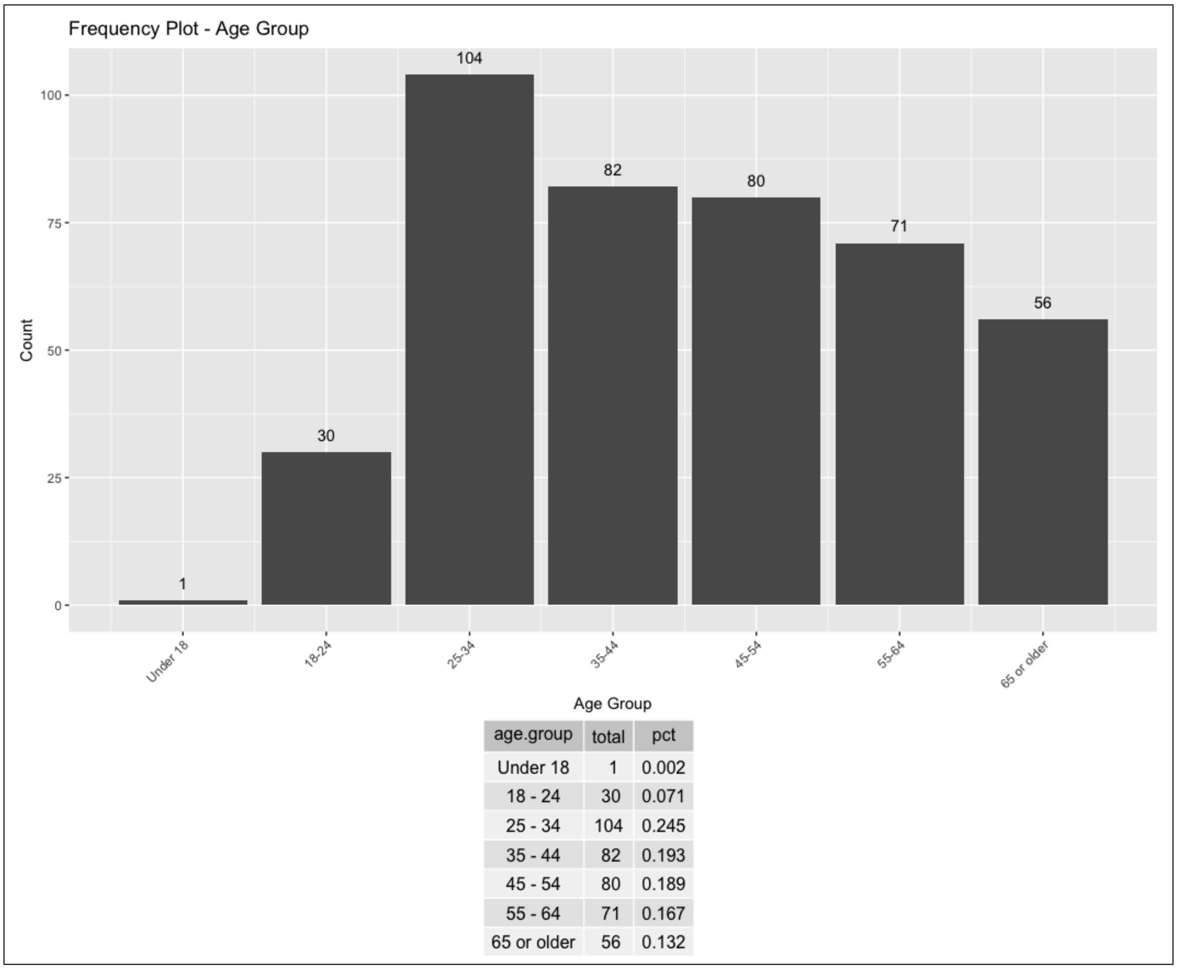
Lynch, S. M. (2007). *Introduction to applied bayesian statistics and estimation for social scientists*. New York: Springer.

Appendix

EDA

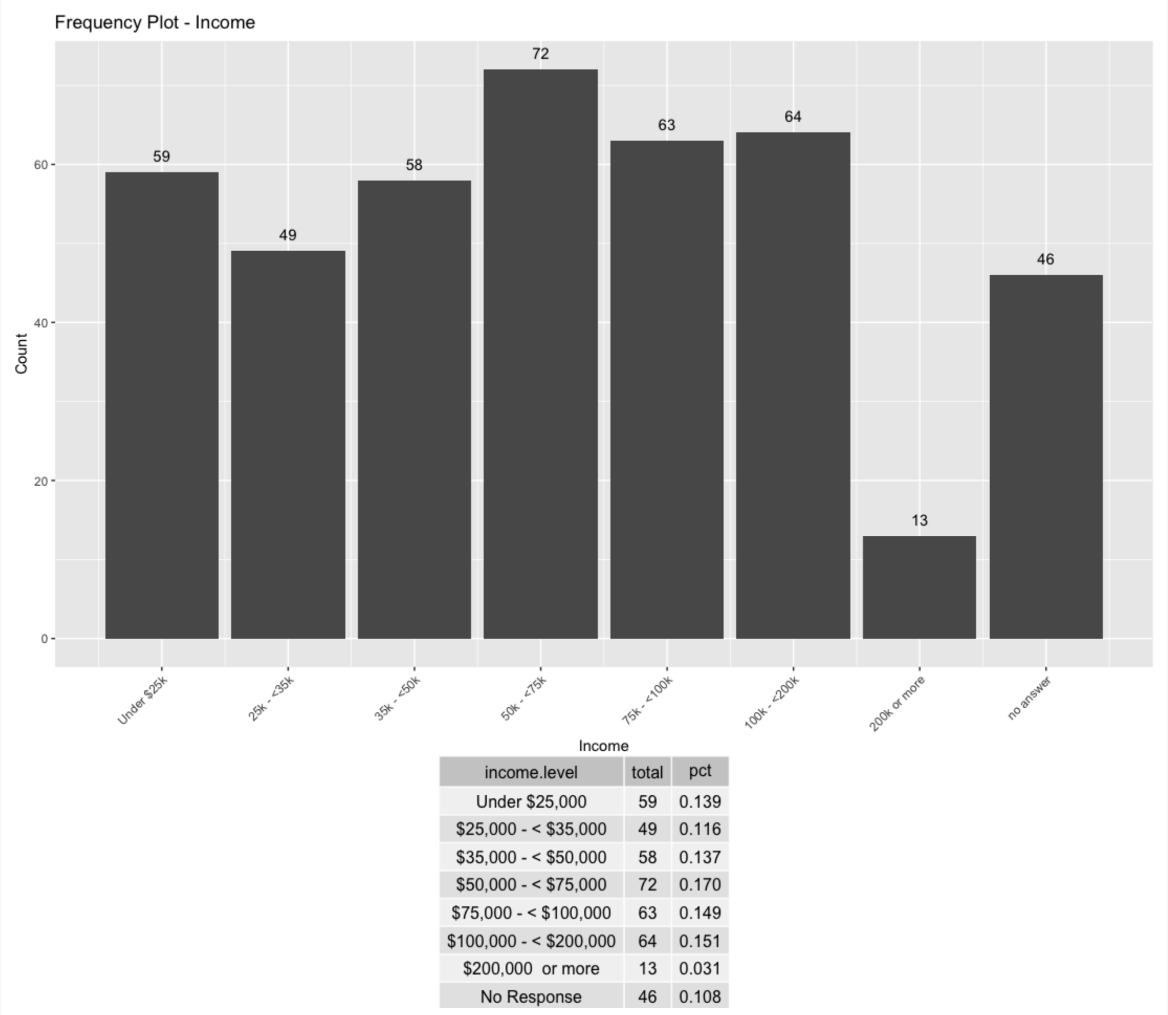
Table 8.1: Frequency Count - Gender

	Count	Percent
Male	214	0.505
Female	210	0.495



age.group	total	pct
Under 18	1	0.002
18 - 24	30	0.071
25 - 34	104	0.245
35 - 44	82	0.193
45 - 54	80	0.189
55 - 64	71	0.167
65 or older	56	0.132

Figure 8.1: Frequency Plot & Table: Age Group



income.level	total	pct
Under $25,000	59	0.139
$25,000 - < $35,000	49	0.116
$35,000 - < $50,000	58	0.137
$50,000 - < $75,000	72	0.170
$75,000 - < $100,000	63	0.149
$100,000 - < $200,000	64	0.151
$200,000 or more	13	0.031
No Response	46	0.108

Figure 8.2: Frequency Plot & Table: Income Level

Table 8.2: Frequency Count - Have Children

Have Kids?	Count	Percent
Yes	277	0.653
No	147	0.347

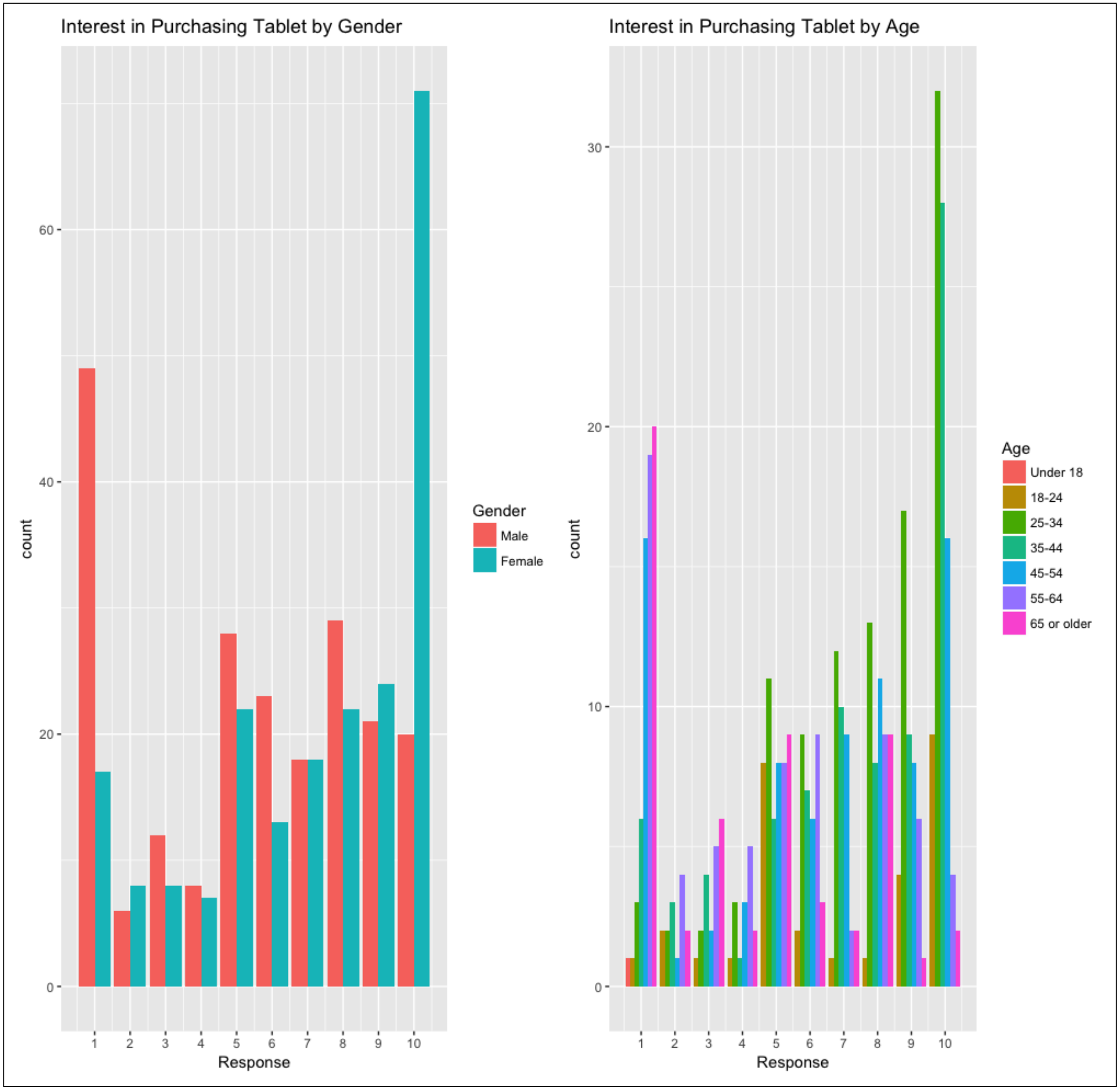


Figure 8.3: Likelihood of Purchasing Tablet by Gender & Age

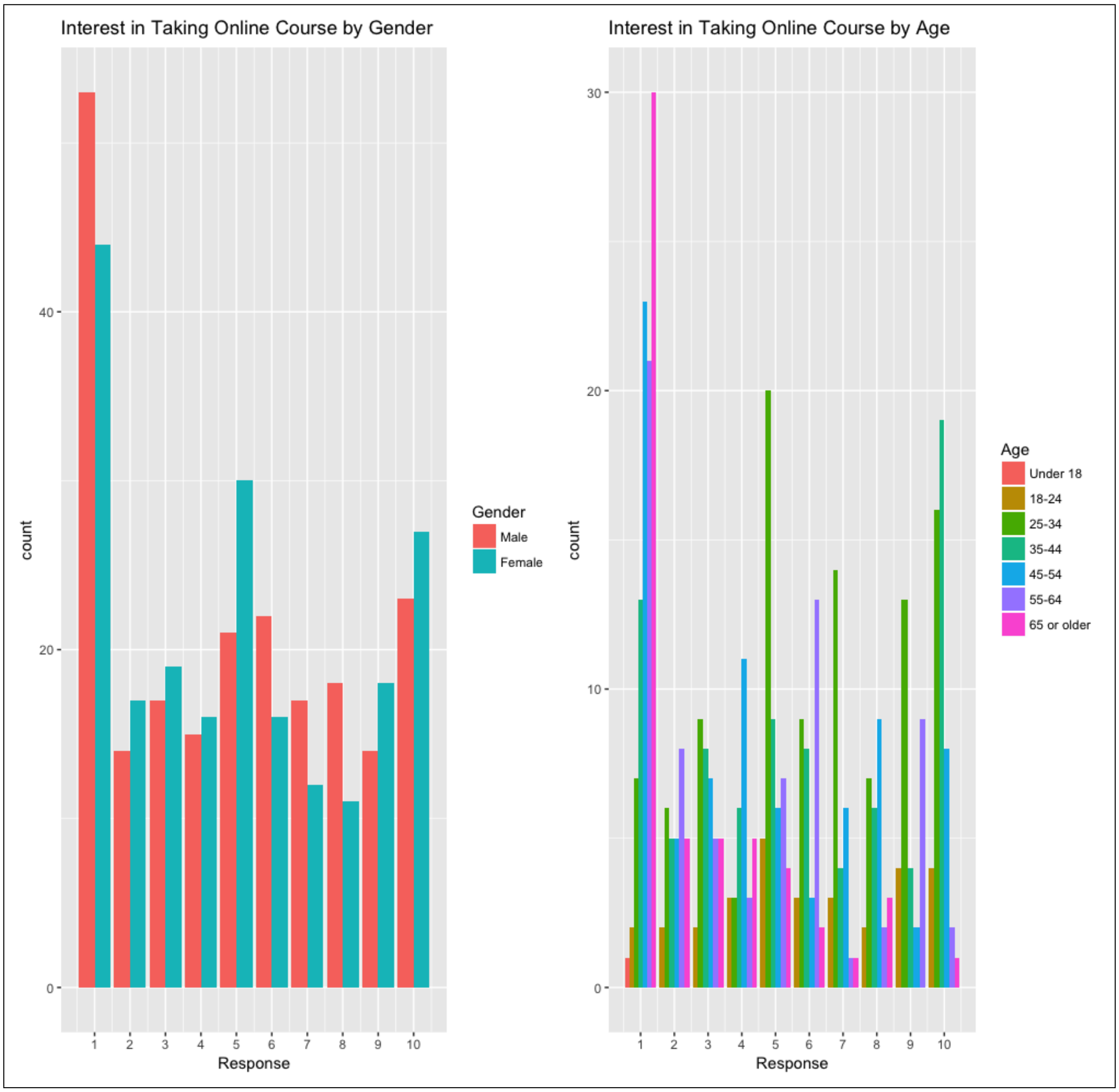


Figure 8.4: Interest in Taking Online Course by Gender & Age

Data Preprocessing

Table 8.3: List of Variables

Variable Name	Variable
screen7	X_1
screen10	X_2
ram16	X_3
ram32	X_4
proc2	X_5
proc2.5	X_6
price299	X_7
price399	X_8
brand.somesong	X_9
brand.pear	X_{10}
brand.gaggle	X_{11}
brand.price.somesong	X_{12}
brand.price.pear	X_{13}
brand.price.gaggle	X_{14}

Model 1

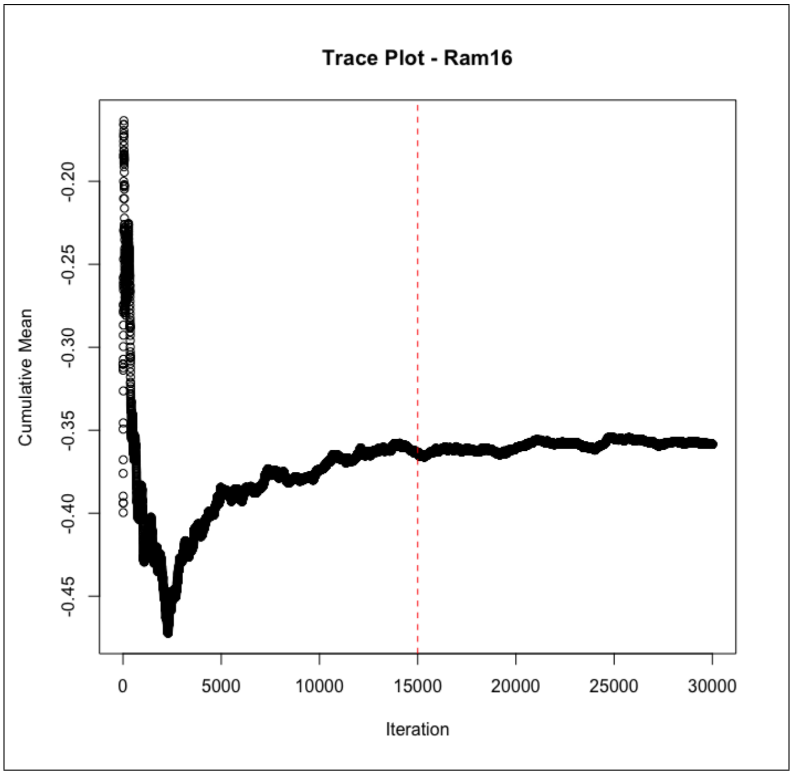


Figure 8.5: Cumulative Mean Trace Plot - Respondent 23, RAM16

Table 8.4: Odds Ratio Table for Interaction Variables in Model 1

Price	Preference	brand.stc	brand.somesong	brand.price.stc	brand.price.somesong	Log Odds	Odds Ratio
199	-1	0.424	-0.202	-0.114	0.0465	0.7865	2.20
299	0	0.424	-0.202	-0.114	0.0465	0.626	1.87
399	1	0.424	-0.202	-0.114	0.0465	0.4655	1.59

Price	Preference	brand.stc	brand.pear	brand.price.stc	brand.price.pear	Log Odds	Odds Ratio
199	-1	0.424	0.0822	-0.114	0.055	0.5108	1.67
299	0	0.424	0.0822	-0.114	0.055	0.3418	1.41
399	1	0.424	0.0822	-0.114	0.055	0.1728	1.19

Price	Preference	brand.stc	brand.gaggle	brand.price.stc	brand.price.gaggle	Log Odds	Odds Ratio
199	-1	0.424	-0.304	-0.114	0.0122	0.8542	2.35
299	0	0.424	-0.304	-0.114	0.0122	0.728	2.07
399	1	0.424	-0.304	-0.114	0.0122	0.6018	1.83

Model 2

Table 8.5: Odds Ratio Table for Interaction Variables in Model 2

Price	Preference	brand.stc	brand.somesong	brand.price.stc	brand.price.somesong	Log Odds	Odds Ratio
199	-1	0.413	-0.23	-0.0836	0.0973	0.8239	2.28
299	0	0.413	-0.23	-0.0836	0.0973	0.643	1.90
399	1	0.413	-0.23	-0.0836	0.0973	0.4621	1.59

Price	Preference	brand.stc	brand.pear	brand.price.stc	brand.price.pear	Log Odds	Odds Ratio
199	-1	0.413	0.109	-0.0836	0.0147	0.4023	1.50
299	0	0.413	0.109	-0.0836	0.0147	0.304	1.36
399	1	0.413	0.109	-0.0836	0.0147	0.2057	1.23

Price	Preference	brand.stc	brand.gaggle	brand.price.stc	brand.price.gaggle	Log Odds	Odds Ratio
199	-1	0.413	-0.292	-0.0836	-0.0284	0.7602	2.14
299	0	0.413	-0.292	-0.0836	-0.0284	0.705	2.02
399	1	0.413	-0.292	-0.0836	-0.0284	0.6498	1.92

Model Results

Table 8.6: Predicted Results for Models 1 and 2

	Actual Results			Model 1 Predicted Results			Model 2 Predicted Results		
Choice Set	**Alternative 1**	**Alternative 2**	**Alternative 3**	**Alternative 1**	**Alternative 2**	**Alternative 3**	**Alternative 1**	**Alternative 2**	**Alternative 3**
1	93	242	89	86	247	91	87	250	87
2	316	45	63	342	31	51	345	30	49
3	218	180	26	238	166	20	242	163	19
4	125	28	271	111	15	298	110	13	301
5	157	132	135	183	111	130	190	107	127
6	70	83	271	69	61	294	70	53	301
7	81	182	161	73	191	160	77	189	158
8	65	85	274	59	70	295	60	65	299
9	100	21	303	106	12	306	105	7	312
10	25	307	92	32	302	90	32	305	87
11	271	113	40	301	95	28	306	91	27
12	192	217	15	192	219	13	199	214	11
13	77	75	272	61	68	295	54	71	299
14	151	148	125	174	128	122	176	128	120
15	18	140	266	20	122	282	16	115	293
16	65	207	152	47	217	160	46	220	158
17	22	130	272	17	119	288	16	115	293
18	64	68	292	58	62	304	55	64	305
19	41	276	107	35	279	110	34	286	104
20	303	42	79	330	27	67	330	28	66
21	225	147	52	262	114	48	275	99	50
22	114	25	285	105	13	306	109	11	304
23	183	88	153	211	56	157	211	53	160
24	37	88	299	29	77	318	30	75	319
25	71	177	176	53	185	186	51	190	183
26	32	103	289	36	79	309	33	84	307
27	109	19	296	96	8	320	95	8	321
28	34	308	82	36	309	79	32	313	79
29	312	67	45	327	54	43	331	49	44
30	189	209	26	214	188	22	222	181	21
31	88	43	293	78	31	315	71	30	323
32	164	121	139	177	103	144	181	97	146
33	27	114	283	22	90	312	20	87	317
34	44	207	173	33	218	173	26	225	173
35	35	116	273	25	99	300	24	101	299
36	86	36	302	75	26	323	64	26	334

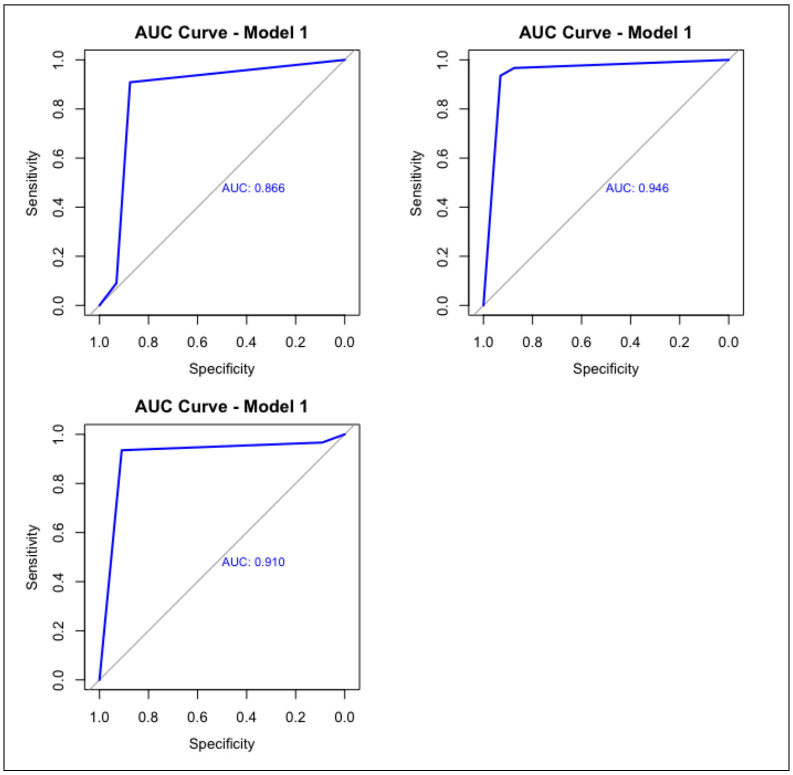


Figure 8.6: AUC Curves - Model 1

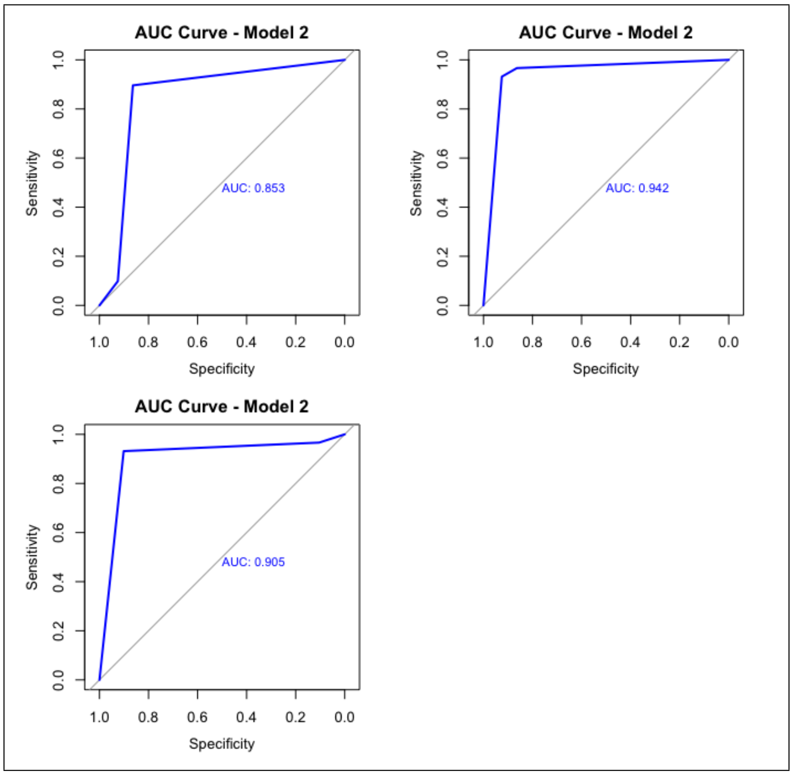


Figure 8.7: AUC Curves - Model 2

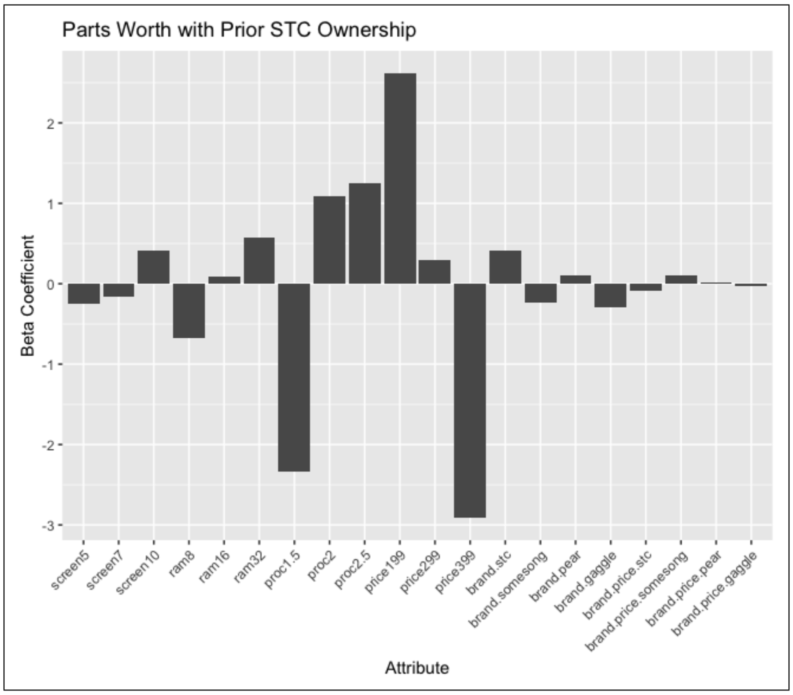


Figure 8.8: Parts Worth - Model 2

R Code

```
# Nikhil Agarwal
# PREDICT 450
# Northwestern University

# load libraries ----
library(tidyverse)
library(bayesm)
library(pROC)
library(gridExtra)

# load input files ----

load(’data/stc-cbc-respondents-v3(1).RData’)
load(’data/efCode.RData’)
cbc <- read_delim(file = ’data/stc-dc-task-cbc-v3.csv’, delim = ’\t’)


# load responsdent file in a new df
response.data <- resp.data.v3

# preview files ----
glimpse(cbc)
glimpse(response.data)

# get NA count
cbc %>%
map_df(function(x) sum(is.na(x))) %>%
gather() %>%
arrange(desc(value))

response.data %>%
map_df(function(x) sum(is.na(x))) %>%
gather() %>%
arrange(desc(value))


# EDA ----

# get total counts for each preferece within each choice set
response.data %>%
select(starts_with(’DCM’), -dcm1_timer) %>%
gather() %>%
group_by(key, value) %>%
summarize(
cnt = n()
) %>%
spread(key = value, value = cnt, fill = 0) %>%
head()

response.data %>%
select(starts_with(’DCM’), -dcm1_timer) %>%
gather() %>%
group_by(value) %>%
summarize(
total.count = n()
) %>%
ungroup()


# Likelihood of purchasing a new tablet
response.data %>%
select(uuid, STT2r1) %>%
ggplot(aes(x = STT2r1)) +
geom_histogram() +
scale_x_continuous(breaks = seq(1,10,1))

response.data %>%
select(D1, STT2r1) %>%
ggplot(aes(x = STT2r1, fill = as.factor(D1))) +
geom_bar(stat = ’count’, position = ’dodge’) +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = ’Response’) +
ggtitle(’Likelihood of Purchasing Tablet by Gender’) +
scale_fill_discrete(
name = ’Gender’,
breaks = c(1,2),
labels = c(’Male’,’Female’)
)

response.data %>%
select(D2, STT2r1) %>%
ggplot(aes(x = STT2r1, fill = as.factor(D2))) +
geom_bar(stat = ’count’, position = ’dodge’) +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = ’Response’) +
ggtitle('Likelihood of Purchasing Tablet by Age') +
scale_fill_discrete(
name = 'Age',
breaks = c(1,2,3,4,5,6,7),
labels = c('Under 18','18-24','25-34','35-44','45-54','55-64','65 or older')
)

a <- response.data %>%
select(D1, STT2r1) %>%
ggplot(aes(x = STT2r1, fill = as.factor(D1))) +
geom_bar(stat = 'count', position = 'dodge') +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = 'Response') +
ggtitle('Interest in Purchasing Tablet by Gender') +
scale_fill_discrete(
name = 'Gender',
breaks = c(1,2),
labels = c('Male','Female')
)

b <- response.data %>%
select(D2, STT2r1) %>%
ggplot(aes(x = STT2r1, fill = as.factor(D2))) +
geom_bar(stat = 'count', position = 'dodge') +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = 'Response') +
ggtitle('Interest in Purchasing Tablet by Age') +
scale_fill_discrete(
name = 'Age',
breaks = c(1,2,3,4,5,6,7),
labels = c('Under 18','18-24','25-34','35-44','45-54','55-64','65 or older')
)

grid.arrange(a,b,ncol = 2)

response.data %>%
select(D3, STT2r1) %>%
ggplot(aes(x = STT2r1, fill = as.factor(D3))) +
geom_bar(stat = 'count', position = 'dodge') +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = 'Response') +
ggtitle('Likelihood of Purchasing Tablet if Parent') +
scale_fill_discrete(
name = 'Parental Status',
breaks = c(1,2),
labels = c('Parent','Not Parent')
)

response.data %>%
select(D5, STT2r1) %>%
ggplot(aes(x = STT2r1, fill = as.factor(D5))) +
geom_bar(stat = 'count', position = 'dodge') +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = 'Response') +
ggtitle('Likelihood of Purchasing Tablet by Income') +
scale_fill_discrete(
name = 'Income',
breaks = c(1,2,3,4,5,6,7,8),
labels = c('Under $25k','25k - <35k','35k - <50k', '50k - <75k','75k - <100k','100k - <200k
    ','200k or more','no answer')
)

response.data %>%
select(uuid, STT2r1) %>%
group_by(
STT2r1
) %>%
summarize(
total.respondents = n(),
pct.of.total = total.respondents / 424
) %>%
ungroup()

# Liklihood of purchasing a new smartphone
response.data %>%
select(uuid, STT2r2) %>%
ggplot(aes(x = STT2r2)) +
geom_histogram() +
scale_x_continuous(breaks = seq(1,10,1))


response.data %>%
select(D1, STT2r2) %>%
ggplot(aes(x = STT2r2, fill = as.factor(D1))) +
geom_bar(stat = 'count', position = 'dodge') +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = 'Response') +
ggtitle('Likelihood of Purchasing Smartphone by Gender') +
scale_fill_discrete(
name = 'Gender',
breaks = c(1,2),
labels = c('Male','Female')
)

response.data %>%
select(D2, STT2r2) %>%
ggplot(aes(x = STT2r2, fill = as.factor(D2))) +
geom_bar(stat = 'count', position = 'dodge') +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = 'Response') +
ggtitle('Likelihood of Purchasing Smartphone by Age') +
scale_fill_discrete(
name = 'Age',
breaks = c(1,2,3,4,5,6,7),
labels = c('Under 18','18-24','25-34','35-44','45-54','55-64','65 or older')
)

response.data %>%
select(D3, STT2r2) %>%
ggplot(aes(x = STT2r2, fill = as.factor(D3))) +
geom_bar(stat = 'count', position = 'dodge') +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = 'Response') +
ggtitle('Likelihood of Purchasing Smartphone if Parent') +
scale_fill_discrete(
name = 'Parental Status',
breaks = c(1,2),
labels = c('Parent','Not Parent')
```

```
)

response.data %>%
select(D5, STT2r2) %>%
ggplot(aes(x = STT2r2, fill = as.factor(D5))) +
geom_bar(stat = ’count’, position = ’dodge’) +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = ’Response’) +
ggtitle(’Likelihood of Purchasing Smartphone by Income’) +
scale_fill_discrete(
name = ’Income’,
breaks = c(1,2,3,4,5,6,7,8),
labels = c(’Under $25k’,’25k - <35k’,’35k - <50k’, ’50k - <75k’,’75k - <100k’,’100k - <200k
    ’,’200k or more’,’no answer’)
)



response.data %>%
select(uuid, STT2r2) %>%
group_by(
STT2r2
) %>%
summarize(
total.respondents = n(),
pct.of.total = total.respondents / 424
) %>%
ungroup()

# using cloud storage
response.data %>%
select(uuid, STT2r3) %>%
ggplot(aes(x = STT2r3)) +
geom_histogram() +
scale_x_continuous(breaks = seq(1,10,1))


response.data %>%
select(D1, STT2r3) %>%
ggplot(aes(x = STT2r3, fill = as.factor(D1))) +
geom_bar(stat = ’count’, position = ’dodge’) +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = ’Response’) +
ggtitle(’Interest in Using Cloud Storage by Gender’) +
scale_fill_discrete(
name = ’Gender’,
breaks = c(1,2),
labels = c(’Male’,’Female’)
)

response.data %>%
select(D2, STT2r3) %>%
ggplot(aes(x = STT2r3, fill = as.factor(D2))) +
geom_bar(stat = ’count’, position = ’dodge’) +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = ’Response’) +
ggtitle(’Interest in Using Cloud Storage by Age’) +
scale_fill_discrete(
name = ’Age’,
breaks = c(1,2,3,4,5,6,7),
labels = c(’Under 18’,’18-24’,’25-34’,’35-44’,’45-54’,’55-64’,’65 or older’)
)

response.data %>%
select(D3, STT2r3) %>%
ggplot(aes(x = STT2r3, fill = as.factor(D3))) +
geom_bar(stat = ’count’, position = ’dodge’) +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = ’Response’) +
ggtitle(’Likelihood of Using Cloud Storage if Parent’) +
scale_fill_discrete(
name = ’Parental Status’,
breaks = c(1,2),
labels = c(’Parent’,’Not Parent’)
)

response.data %>%
select(D5, STT2r3) %>%
ggplot(aes(x = STT2r3, fill = as.factor(D5))) +
geom_bar(stat = ’count’, position = ’dodge’) +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = ’Response’) +
ggtitle(’Likelihood of Using Cloud Storage by Income’) +
scale_fill_discrete(
name = ’Income’,
breaks = c(1,2,3,4,5,6,7,8),
labels = c(’Under $25k’,’25k - <35k’,’35k - <50k’, ’50k - <75k’,’75k - <100k’,’100k - <200k
    ’,’200k or more’,’no answer’)
)



response.data %>%
select(uuid, STT2r3) %>%
group_by(
STT2r3
) %>%
summarize(
total.respondents = n(),
pct.of.total = total.respondents / 424
) %>%
ungroup()

# taking online course
response.data %>%
select(uuid, STT2r4) %>%
ggplot(aes(x = STT2r4)) +
geom_histogram() +
scale_x_continuous(breaks = seq(1,10,1))

response.data %>%
select(D1, STT2r4) %>%
ggplot(aes(x = STT2r4, fill = as.factor(D1))) +
geom_bar(stat = ’count’, position = ’dodge’) +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = ’Response’) +
ggtitle(’Interest in Taking Online Course by Gender’) +
scale_fill_discrete(
name = 'Gender',
breaks = c(1,2),
labels = c('Male','Female')
)

response.data %>%
select(D2, STT2r4) %>%
ggplot(aes(x = STT2r4, fill = as.factor(D2))) +
geom_bar(stat = 'count', position = 'dodge') +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = 'Response') +
ggtitle('Interest in Taking Online Course by Age') +
scale_fill_discrete(
name = 'Age',
breaks = c(1,2,3,4,5,6,7),
labels = c('Under 18','18-24','25-34','35-44','45-54','55-64','65 or older')
)

a <- response.data %>%
select(D1, STT2r4) %>%
ggplot(aes(x = STT2r4, fill = as.factor(D1))) +
geom_bar(stat = 'count', position = 'dodge') +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = 'Response') +
ggtitle('Interest in Taking Online Course by Gender') +
scale_fill_discrete(
name = 'Gender',
breaks = c(1,2),
labels = c('Male','Female')
)

b <- response.data %>%
select(D2, STT2r4) %>%
ggplot(aes(x = STT2r4, fill = as.factor(D2))) +
geom_bar(stat = 'count', position = 'dodge') +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = 'Response') +
ggtitle('Interest in Taking Online Course by Age') +
scale_fill_discrete(
name = 'Age',
breaks = c(1,2,3,4,5,6,7),
labels = c('Under 18','18-24','25-34','35-44','45-54','55-64','65 or older')
)

grid.arrange(a, b, ncol=2)

response.data %>%
select(D3, STT2r4) %>%
ggplot(aes(x = STT2r4, fill = as.factor(D3))) +
geom_bar(stat = 'count', position = 'dodge') +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = 'Response') +
ggtitle('Likelihood of Taking Online Course if Parent') +
scale_fill_discrete(
name = 'Parental Status',
breaks = c(1,2),
labels = c('Parent','Not Parent')
)
```

```
response.data %>%
select(D5, STT2r4) %>%
ggplot(aes(x = STT2r4, fill = as.factor(D5))) +
geom_bar(stat = ’count’, position = ’dodge’) +
scale_x_continuous(breaks = seq(1,10,1)) +
labs(x = ’Response’) +
ggtitle(’Likelihood of Taking Online Course by Income’) +
scale_fill_discrete(
name = ’Income’,
breaks = c(1,2,3,4,5,6,7,8),
labels = c(’Under $25k’,’25k - <35k’,’35k - <50k’, ’50k - <75k’,’75k - <100k’,’100k - <200k
    ’,’200k or more’,’no answer’)
)

response.data %>%
select(uuid, STT2r4) %>%
group_by(
STT2r4
) %>%
summarize(
total.respondents = n(),
pct.of.total = total.respondents / 424
) %>%
ungroup()

# gender breakout
response.data %>%
select(D1) %>%
group_by(D1) %>%
summarize(
total = n(),
pct = n() / 424
) %>%
ungroup()

# age breakout
response.data %>%
select(D2) %>%
group_by(D2) %>%
summarize(
total = n(),
pct = n() / 424
) %>%
ungroup()

response.data %>%
select(D2) %>%
ggplot(aes(x = D2)) +
geom_bar(stat = ’count’) +
geom_text(stat = ’count’, aes(label = ..count..), vjust = -1) +
scale_x_continuous(
breaks = c(1, 2, 3, 4, 5, 6, 7),
labels = c(’Under 18’,’18-24’,’25-34’,’35-44’,’45-54’,’55-64’,’65 or older’)
) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = ’Age Group’, y = ’Count’) +
ggtitle("Frequency Plot - Age Group")
```

```
a <- response.data %>%
select(D2) %>%
group_by(D2) %>%
summarize(
total = n(),
pct = round(n() / 424,3)
) %>%
ungroup() %>%
transmute(
age.group = case_when(
D2 == 1 ~ 'Under 18',
D2 == 2 ~ '18 - 24',
D2 == 3 ~ '25 - 34',
D2 == 4 ~ '35 - 44',
D2 == 5 ~ '45 - 54',
D2 == 6 ~ '55 - 64',
D2 == 7 ~ '65 or older'
),
total,
pct
)

b <- response.data %>%
select(D2) %>%
ggplot(aes(x = D2)) +
geom_bar(stat = 'count') +
geom_text(stat = 'count', aes(label = ..count..), vjust = -1) +
scale_x_continuous(
breaks = c(1, 2, 3, 4, 5, 6, 7),
labels = c('Under 18','18-24','25-34','35-44','45-54','55-64','65 or older')
) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = 'Age Group', y = 'Count') +
ggtitle("Frequency Plot - Age Group")

grid.arrange(b, tableGrob(a, theme = ttheme_default(colhead=list(fg_params = list(parse=TRUE
    ))), rows = NULL), nrow = 2, as.table = T, heights = c(3,1))

# parent?
response.data %>%
select(D3) %>%
group_by(D3) %>%
summarize(
total = n(),
pct = n() / 424
) %>%
ungroup()

# children
response.data %>%
select(starts_with('D4r')) %>%
head

response.data %>%
select(starts_with('D4r')) %>%
gather(key = 'question', value = 'response') %>%
group_by(question, response) %>%
summarize(
count = n()
```

```
) %>%
ungroup() %>%
spread(key = response, value = count) %>%
mutate(
pct_0 = ‘0‘ / 424,
pct_1 = ‘1‘ / 424,
pct_na = ‘<NA>‘ / 424
)

response.data %>%
select(starts_with(’D4r’)) %>%
mutate(
have.kids = case_when(
D4r1 + D4r2 + D4r3 + D4r4 + D4r5 + D4r6 >= 1 ~ ’Yes’,
TRUE ~ ’No’
)
) %>%
group_by(have.kids) %>%
summarize(
obs.count = n(),
obs.pct = n() / 424
) %>%
head

# income

response.data %>%
select(D5) %>%
ggplot(aes(x = D5)) +
geom_histogram()


response.data %>%
select(D5) %>%
group_by(D5) %>%
summarize(
total = n(),
pct = n() / 424
) %>%
ungroup()

response.data %>%
ggplot(aes(x = D5)) +
geom_bar(stat = ’count’) +
geom_text(stat = ’count’, aes(label = ..count..), vjust = -1) +
scale_x_continuous(
breaks = c(1, 2, 3, 4, 5, 6, 7, 8),
labels = c(’Under $25k’,’25k - <35k’,’35k - <50k’, ’50k - <75k’,’75k - <100k’,’100k - <200k
    ’,’200k or more’,’no answer’)
) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = ’Income’, y = ’Count’) +
ggtitle("Frequency Plot - Income")

a <- response.data %>%
select(D5) %>%
group_by(D5) %>%
summarize(
total = n(),
pct = round(n() / 424,3)
) %>%
ungroup() %>%
transmute(
income.level = case_when(
D5 == 1 ˜ ’Under $25,000’,
D5 == 2 ˜ ’$25,000 - < $35,000’,
D5 == 3 ˜ ’$35,000 - < $50,000’,
D5 == 4 ˜ ’$50,000 - < $75,000’,
D5 == 5 ˜ ’$75,000 - < $100,000’,
D5 == 6 ˜ ’$100,000 - < $200,000’,
D5 == 7 ˜ ’$200,000 or more’,
D5 == 8 ˜ ’No Response’
),
total,
pct
)

b <- response.data %>%
ggplot(aes(x = D5)) +
geom_bar(stat = ’count’) +
geom_text(stat = ’count’, aes(label = ..count..), vjust = -1) +
scale_x_continuous(
breaks = c(1, 2, 3, 4, 5, 6, 7, 8),
labels = c(’Under $25k’,’25k - <35k’,’35k - <50k’, ’50k - <75k’,’75k - <100k’,’100k - <200k
    ’,’200k or more’,’no answer’)
) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = ’Income’, y = ’Count’) +
ggtitle("Frequency Plot - Income")

grid.arrange(b, tableGrob(a, theme = ttheme_default(colhead=list(fg_params = list(parse=TRUE
    ))), rows = NULL), nrow = 2, as.table = T, heights = c(3,1))

# state of residence

response.data %>%
select(D6) %>%
ggplot(aes(x = D6)) +
geom_bar(stat = ’count’) +
scale_x_continuous(breaks = seq(1,53,1))

response.data %>%
select(D1, D6) %>%
ggplot(aes(x = D6, fill = as.factor(D1))) +
geom_bar(stat = ’count’, position = ’dodge’) +
scale_x_continuous(breaks = seq(1,53,1))

response.data %>%
select(D6) %>%
group_by(D6) %>%
summarize(
total = n(),
pct = n() / 424
) %>%
ungroup()
```

```
# create the proper dummies ----

cbc2 <- cbc %>%
transmute(
choice.set,
choice.ID,
# screen7 = ifelse(screen == 0, -1, screen),
# screen10 = ifelse(screen == 0, -1, screen)
screen7 = case_when(
screen == 0 ~ -1,
screen == 1 ~ 1,
TRUE ~ 0
),
screen10 = case_when(
screen == 0 ~ -1,
screen == 2 ~ 1,
TRUE ~ 0
),
ram16 = case_when(
RAM == 0 ~ -1,
RAM == 1 ~ 1,
TRUE ~ 0
),
ram32 = case_when(
RAM == 0 ~ -1,
RAM == 2 ~ 1,
TRUE ~ 0
),
proc2 = case_when(
processor == 0 ~ -1,
processor == 1 ~ 1,
TRUE ~ 0
),
proc2.5 = case_when(
processor == 0 ~ -1,
processor == 2 ~ 1,
TRUE ~ 0
),
price299 = case_when(
price == 0 ~ -1,
price == 1 ~ 1,
TRUE ~ 0
),
price399 = case_when(
price == 0 ~ -1,
price == 2 ~ 1,
TRUE ~ 0
),
brand.somesong = case_when(
brand == 0 ~ -1,
brand == 1 ~ 1,
TRUE ~ 0
),
brand.pear = case_when(
brand == 0 ~ -1,
brand == 2 ~ 1,
TRUE ~ 0
),
brand.gaggle = case_when(
brand == 0 ~ -1,
brand == 3 ~ 1,
TRUE ~ 0
),
scaled.price = scale(price, scale = F),
brand.price.somesong = brand.somesong * scaled.price,
brand.price.pear = brand.pear * scaled.price,
brand.price.gaggle = brand.gaggle * scaled.price
)

y.data <- as.matrix(response.data %>%
select(starts_with('DCM'), -dcm1_timer))

z.owner <- response.data %>%
transmute(
zowner = 1 * (!is.na(vList3))
)

# create a list for bayesm pkg ----

cbc2 <- as.matrix(cbc2 %>% select(-choice.ID, -choice.set, -scaled.price))
#det(t(cbc2) %*% cbc2) # check to make sure my data frame method is working as expected



lgtdata2 = NULL # a starter placeholder for your list
for (i in 1:424) {
lgtdata2[[i]]=list(y=y.data[i,],X=cbc2)
}
length(lgtdata2)
str(lgtdata2)

#lgtdata2[[3]] %>% View

# trial run ----
lgt100 <- lgtdata2[1:100]
mcmctest <- list(R = 5000, keep = 5)
data.list <- list(lgtdata = lgt100, p = 3)

set.seed(1234)
dummy.test <- rhierMnlDP(Data = data.list, Mcmc = mcmctest)

names(dummy.test)
dim(dummy.test$betadraw)

plot(1:length(dummy.test$betadraw[1,1,]), dummy.test$betadraw[1,1,])
plot(density(dummy.test$betadraw[1,1,701:1000], width = 2))
abline(v = mean(dummy.test$betadraw[1,1,701:1000]), col = 'red') # mean of beta 1 using the
    last 300 obs.
abline(v = 0, col = 'blue', lty = 2) # no preference
# now that we know that the mean is greater than 0, we know that beta1 (see the second
    number within the brackets) is more preferred than not liking it (which happens to be
    the 7" screen)
pnorm(0, mean = mean(dummy.test$betadraw[1,1,701:1000]), sd = sd(dummy.test$betadraw
    [1,1,701:1000]), lower.tail = F)
# area to the right of 0 in the plot happens to be 0.979. in other words, 98% like, 2% don't
```

```
quantile(dummy.test$betadraw[1,1,701:1000], probs = seq(0, 1, by= 0.1))
summary(dummy.test$betadraw[1,1,701:1000])


plot.bayesm.hcoef(dummy.test$betadraw)

apply(dummy.test$betadraw[,,701:1000],c(2),mean)
apply(dummy.test$betadraw[,,701:1000],c(1,2),mean)

betameans.overall <- apply(dummy.test$betadraw[,,701:1000],c(2),mean)
descrip.quantiles <- apply(dummy.test$betadraw[,,701:1000],2,quantile,probs=c
    (0.05,0.10,0.25,0.5 ,0.75,0.90,0.95))

# Model 1 - no covariate ----
mcmctest.final <- list(R = 150000, keep = 5)
data.list.final <- list(lgtdata = lgtdata2, p = 3)

set.seed(1234)
logit.model.1 <- rhierMnlDP(Data = data.list.final, Mcmc = mcmctest.final)

# let's plot the hcoef betadraws
plot.bayesm.hcoef(logit.model.1$betadraw)
plot.bayesm.mat(logit.model.1$loglike)
plot.bayesm.nmix(logit.model.1$nmix)

# choose person 23
# plot person 23's beta values for var 1
logit.model.1$betadraw[23,1,]

p23.betadraws <- tibble(
screen7 = logit.model.1$betadraw[23,1,],
screen10 = logit.model.1$betadraw[23,2,],
ram16 = logit.model.1$betadraw[23,3,],
ram32 = logit.model.1$betadraw[23,4,],
proc2 = logit.model.1$betadraw[23,5,],
proc2.5 = logit.model.1$betadraw[23,6,],
price299 = logit.model.1$betadraw[23,7,],
price399 = logit.model.1$betadraw[23,8,],
brand.somesong = logit.model.1$betadraw[23,9,],
brand.pear = logit.model.1$betadraw[23,10,],
brand.gaggle = logit.model.1$betadraw[23,11,],
brand.price.somesong = logit.model.1$betadraw[23,12,],
brand.price.pear = logit.model.1$betadraw[23,13,],
brand.price.gaggle = logit.model.1$betadraw[23,14,]
)

p23.betadraws %>%
ggplot(aes(x=screen7)) +
geom_density()


p23.betadraws %>%
ggplot(aes(x = seq(1,nrow(.),1), y = screen7)) +
geom_line() +
labs(x = "Iteration", y = 'Beta Values - Screen7')

p23.betadraws %>%
ggplot(aes(x = seq(1,nrow(.),1), y = screen7)) +
geom_point() +
labs(x = "Iteration", y = ’Beta Values - Screen7’)

plot(x = seq(1,30000,1), y = cumsum(p23.betadraws$screen7) / seq_along(p23.betadraws$screen7
    ), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace Plot = Screen7’)
abline(v = 25000, col = ’red’, lty=2)
# seems like at 25000, it goes steady state
plot(x = seq(1,30000,1), y = cumsum(p23.betadraws$screen10) / seq_along(p23.
    betadraws$screen10), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace Plot -
     Screen10’)
plot(x = seq(1,30000,1), y = cumsum(p23.betadraws$ram16) / seq_along(p23.betadraws$ram16),
    xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace Plot - Ram16’)
abline(v = 15000, col = ’red’, lty = 2)
plot(x = seq(1,30000,1), y = cumsum(p23.betadraws$ram32) / seq_along(p23.betadraws$ram32),
    xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace Plot - Ram32’)
plot(x = seq(1,30000,1), y = cumsum(p23.betadraws$proc2) / seq_along(p23.betadraws$proc2),
    xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace Plot - Processor2’)
plot(x = seq(1,30000,1), y = cumsum(p23.betadraws$proc2.5) / seq_along(p23.betadraws$proc2
    .5), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace Plot - Processor2.5’)
plot(x = seq(1,30000,1), y = cumsum(p23.betadraws$price299) / seq_along(p23.
    betadraws$price299), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace Plot -
     Price299’)
plot(x = seq(1,30000,1), y = cumsum(p23.betadraws$price399) / seq_along(p23.
    betadraws$price399), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace Plot -
     Price399’)
plot(x = seq(1,30000,1), y = cumsum(p23.betadraws$brand.somesong) / seq_along(p23.
    betadraws$brand.somesong), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace
    Plot - Somesong’)
plot(x = seq(1,30000,1), y = cumsum(p23.betadraws$brand.pear) / seq_along(p23.
    betadraws$brand.pear), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace Plot
     - Pear’)
plot(x = seq(1,30000,1), y = cumsum(p23.betadraws$brand.gaggle) / seq_along(p23.
    betadraws$brand.gaggle), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace
    Plot - Gaggle’)
plot(x = seq(1,30000,1), y = cumsum(p23.betadraws$brand.price.somesong) / seq_along(p23.
    betadraws$brand.price.somesong), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’
    Trace Plot - Price & Somesong’)
plot(x = seq(1,30000,1), y = cumsum(p23.betadraws$brand.price.pear) / seq_along(p23.
    betadraws$brand.price.pear), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’
    Trace Plot - Price & Pear’)
plot(x = seq(1,30000,1), y = cumsum(p23.betadraws$brand.price.gaggle) / seq_along(p23.
    betadraws$brand.price.gaggle), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’
    Trace Plot - Price & Gaggle’)



minVal = 25001
maxVal = 30000

p23.betadraws.ss <- tibble(
screen7 = logit.model.1$betadraw[23,1,minVal:maxVal],
screen10 = logit.model.1$betadraw[23,2,minVal:maxVal],
ram16 = logit.model.1$betadraw[23,3,minVal:maxVal],
ram32 = logit.model.1$betadraw[23,4,minVal:maxVal],
proc2 = logit.model.1$betadraw[23,5,minVal:maxVal],
proc2.5 = logit.model.1$betadraw[23,6,minVal:maxVal],
price299 = logit.model.1$betadraw[23,7,minVal:maxVal],
price399 = logit.model.1$betadraw[23,8,minVal:maxVal],
brand.somesong = logit.model.1$betadraw[23,9,minVal:maxVal],
brand.pear = logit.model.1$betadraw[23,10,minVal:maxVal],
brand.gaggle = logit.model.1$betadraw[23,11,minVal:maxVal],
brand.price.somesong = logit.model.1$betadraw[23,12,minVal:maxVal],
brand.price.pear = logit.model.1$betadraw[23,13,minVal:maxVal],
brand.price.gaggle = logit.model.1$betadraw[23,14,minVal:maxVal]
)

p23.betadraws.ss %>%
ggplot(aes(x=screen7)) +
geom_density() +
geom_vline(data = p23.betadraws.ss, aes(xintercept = mean(screen7)), col = 'red') +
geom_vline(xintercept = 0, col = 'blue', linetype = 2)

# the following gives area under the curve for the simulated beta1 samples for each
    predictor for candidate 23 in relation to 0 (which implies no preference). The first
    one comes out to be 0.847. This means that the area under the curve is 0.847. This can
    be interpreted as 84% of the simulated beta1 values prefer a screen size of 7" versus
    no preference
pnorm(0, mean = mean(p23.betadraws.ss$screen7), sd = sd(p23.betadraws.ss$screen7), lower.
    tail = F)
pnorm(0, mean = mean(p23.betadraws.ss$screen10), sd = sd(p23.betadraws.ss$screen10), lower.
    tail = F)
pnorm(0, mean = mean(p23.betadraws.ss$ram16), sd = sd(p23.betadraws.ss$ram16), lower.tail =
    F)
pnorm(0, mean = mean(p23.betadraws.ss$ram32), sd = sd(p23.betadraws.ss$ram32), lower.tail =
    F)
pnorm(0, mean = mean(p23.betadraws.ss$proc2), sd = sd(p23.betadraws.ss$proc2), lower.tail =
    F)
pnorm(0, mean = mean(p23.betadraws.ss$proc2.5), sd = sd(p23.betadraws.ss$proc2.5), lower.
    tail = F)
pnorm(0, mean = mean(p23.betadraws.ss$price299), sd = sd(p23.betadraws.ss$price299), lower.
    tail = F)
pnorm(0, mean = mean(p23.betadraws.ss$price399), sd = sd(p23.betadraws.ss$price399), lower.
    tail = F)
pnorm(0, mean = mean(p23.betadraws.ss$brand.somesong), sd = sd(p23.betadraws.ss$brand.
    somesong), lower.tail = F)
pnorm(0, mean = mean(p23.betadraws.ss$brand.pear), sd = sd(p23.betadraws.ss$brand.pear),
    lower.tail = F)
pnorm(0, mean = mean(p23.betadraws.ss$brand.gaggle), sd = sd(p23.betadraws.ss$brand.gaggle),
     lower.tail = F)
pnorm(0, mean = mean(p23.betadraws.ss$brand.price.somesong), sd = sd(p23.betadraws.ss$brand.
    price.somesong), lower.tail = F)
pnorm(0, mean = mean(p23.betadraws.ss$brand.price.pear), sd = sd(p23.betadraws.ss$brand.
    price.pear), lower.tail = F)
pnorm(0, mean = mean(p23.betadraws.ss$brand.price.gaggle), sd = sd(p23.betadraws.ss$brand.
    price.gaggle), lower.tail = F)


# get the mean beta values for all respondents and the 14 predictors
model1.avgBetaVal <- as.data.frame(apply(logit.model.1$betadraw[,,minVal:maxVal],c(1,2),mean
    ))

model1.avgBetaVal <- model1.avgBetaVal %>%
rename(
screen7 = V1,
screen10 = V2,
ram16 = V3,
ram32 = V4,
proc2 = V5,
proc2.5 = V6,
price299 = V7,
price399 = V8,
brand.somesong = V9,
brand.pear = V10,
brand.gaggle = V11,
brand.price.somesong = V12,
brand.price.pear = V13,
brand.price.gaggle =V14
)

summary(model1.avgBetaVal)

min.values <- model1.avgBetaVal %>%
summarise_all(min) %>%
gather(key = variable, value = minimum.value)
max.values <- model1.avgBetaVal %>%
summarize_all(max) %>%
gather(key = variable, value = maximum.value)
q05.values <- model1.avgBetaVal %>%
summarize_all(funs(quantile), probs = 0.05) %>%
gather(key = variable, value = p05.value)
q10.values <- model1.avgBetaVal %>%
summarize_all(funs(quantile), probs = 0.10) %>%
gather(key = variable, value = p10.value)
q25.values <- model1.avgBetaVal %>%
summarize_all(funs(quantile), probs = 0.25) %>%
gather(key = variable, value = p25.value)
median.values <- model1.avgBetaVal %>%
summarize_all(median) %>%
gather(key = variable, value = median.value)
q75.values <- model1.avgBetaVal %>%
summarize_all(funs(quantile), probs = 0.75) %>%
gather(key = variable, value = p75.value)
q90.values <- model1.avgBetaVal %>%
summarize_all(funs(quantile), probs = 0.90) %>%
gather(key = variable, value = p90.value)
q95.values <- model1.avgBetaVal %>%
summarize_all(funs(quantile), probs = 0.95) %>%
gather(key = variable, value = p95.value)

descriptive.stats.model1.avgBetaVals <- min.values %>%
inner_join(max.values, by = ’variable’) %>%
inner_join(median.values, by = ’variable’) %>%
inner_join(q05.values, by = ’variable’) %>%
inner_join(q10.values, by = ’variable’) %>%
inner_join(q25.values, by = ’variable’) %>%
inner_join(q75.values, by = ’variable’) %>%
inner_join(q90.values, by = ’variable’) %>%
inner_join(q95.values, by = ’variable’)

dim(model1.avgBetaVal)


# Model 1 Fit Statistics (no covariate) ----
dim(model1.avgBetaVal)
dim(cbc2)

model1.xbetas <- cbc2 %*% t(as.matrix(model1.avgBetaVal))
```

```
# convert to 3 columns across
model1.xbetas <- matrix(model1.xbetas, ncol = 3, byrow = T)

# take the exponential (remember, it’s a log-odds model) of the matrix
model1.xbetas <- exp(model1.xbetas)

# convert to dataframe
model1.xbetas <- as.data.frame(model1.xbetas)
model1.xbetas <- model1.xbetas %>%
rename(
’choice1’ = V1,
’choice2’ = V2,
’choice3’ = V3
) %>%
mutate(
sum.of.row = rowSums(.)
) %>%
transmute(
choice1 = choice1 / sum.of.row,
choice2 = choice2 / sum.of.row,
choice3 = choice3 / sum.of.row
)

model1.choice <- max.col(model1.xbetas)

responses <- as.vector(t(y.data))

table(model1.choice, responses) # confusion matrix

par(mfrow=c(2,2))
auc(multiclass.roc(responses, model1.choice, print.auc = T, levels = c(1,2,3), col = ’blue’,
     percent = F, plot = T, main = ’AUC Curve - Model 1’)) # AUC
par(mfrow=c(1,1))

mean(logit.model.1$loglike) # -2loglike test

model1.predicted.values <- as.data.frame(apply(t(matrix(model1.choice, nrow = 36, ncol =
    424)), 2, function(x){tabulate(na.omit(x))}))

model1.predicted.values %>%
mutate(
alternative = c(1,2,3)
) %>%
gather(key = choice.set, value, starts_with(’V’)) %>%
spread(key = alternative, value = value) %>%
mutate(
choice.set = case_when(
choice.set == ’V1’ ˜ 1,
choice.set == ’V2’ ˜ 2,
choice.set == ’V3’ ˜ 3,
choice.set == ’V4’ ˜ 4,
choice.set == ’V5’ ˜ 5,
choice.set == ’V6’ ˜ 6,
choice.set == ’V7’ ˜ 7,
choice.set == ’V8’ ˜ 8,
choice.set == ’V9’ ˜ 9,
choice.set == ’V10’ ˜ 10,
choice.set == ’V11’ ˜ 11,
choice.set == ’V12’ ˜ 12,
choice.set == ’V13’ ~ 13,
choice.set == ’V14’ ~ 14,
choice.set == ’V15’ ~ 15,
choice.set == ’V16’ ~ 16,
choice.set == ’V17’ ~ 17,
choice.set == ’V18’ ~ 18,
choice.set == ’V19’ ~ 19,
choice.set == ’V20’ ~ 20,
choice.set == ’V21’ ~ 21,
choice.set == ’V22’ ~ 22,
choice.set == ’V23’ ~ 23,
choice.set == ’V24’ ~ 24,
choice.set == ’V25’ ~ 25,
choice.set == ’V26’ ~ 26,
choice.set == ’V27’ ~ 27,
choice.set == ’V28’ ~ 28,
choice.set == ’V29’ ~ 29,
choice.set == ’V30’ ~ 30,
choice.set == ’V31’ ~ 31,
choice.set == ’V32’ ~ 32,
choice.set == ’V33’ ~ 33,
choice.set == ’V34’ ~ 34,
choice.set == ’V35’ ~ 35,
choice.set == ’V36’ ~ 36
)
) %>%
rename(
alt.1 = ’1’,
alt.2 = ’2’,
alt.3 = ’3’
) %>%
arrange(choice.set) %>%
mutate(
alt.1.pct = round(alt.1 / rowSums(.,dims = 1),3),
alt.2.pct = round(alt.2 / rowSums(.,dims = 1),3),
alt.3.pct = round(alt.3 / rowSums(.,dims = 1),3)
) %>%
head(n=36)

# actual data
response.data %>%
select(starts_with(’DCM’), -dcm1_timer) %>%
gather() %>%
group_by(key, value) %>%
summarize(
cnt = n()
) %>%
ungroup() %>%
spread(key = value, value = cnt, fill = 0) %>%
rename(
choice.set = key,
alt.1 = ‘1‘,
alt.2 = ‘2‘,
alt.3 = ‘3‘
) %>%
mutate(
choice.set = case_when(
choice.set == ’DCM1_1’ ~ 1,
choice.set == ’DCM1_2’ ~ 2,
choice.set == ’DCM1_3’ ~ 3,
choice.set == ’DCM1_4’ ~ 4,
choice.set == ’DCM1_5’ ~ 5,
choice.set == ’DCM1_6’ ~ 6,
choice.set == ’DCM1_7’ ~ 7,
choice.set == ’DCM1_8’ ~ 8,
choice.set == ’DCM1_9’ ~ 9,
choice.set == ’DCM1_10’ ~ 10,
choice.set == ’DCM1_11’ ~ 11,
choice.set == ’DCM1_12’ ~ 12,
choice.set == ’DCM1_13’ ~ 13,
choice.set == ’DCM1_14’ ~ 14,
choice.set == ’DCM1_15’ ~ 15,
choice.set == ’DCM1_16’ ~ 16,
choice.set == ’DCM1_17’ ~ 17,
choice.set == ’DCM1_18’ ~ 18,
choice.set == ’DCM1_19’ ~ 19,
choice.set == ’DCM1_20’ ~ 20,
choice.set == ’DCM1_21’ ~ 21,
choice.set == ’DCM1_22’ ~ 22,
choice.set == ’DCM1_23’ ~ 23,
choice.set == ’DCM1_24’ ~ 24,
choice.set == ’DCM1_25’ ~ 25,
choice.set == ’DCM1_26’ ~ 26,
choice.set == ’DCM1_27’ ~ 27,
choice.set == ’DCM1_28’ ~ 28,
choice.set == ’DCM1_29’ ~ 29,
choice.set == ’DCM1_30’ ~ 30,
choice.set == ’DCM1_31’ ~ 31,
choice.set == ’DCM1_32’ ~ 32,
choice.set == ’DCM1_33’ ~ 33,
choice.set == ’DCM1_34’ ~ 34,
choice.set == ’DCM1_35’ ~ 35,
choice.set == ’DCM1_36’ ~ 36
)
) %>%
arrange(choice.set) %>%
mutate(choice.set = as.character(choice.set)) %>%
View



# Model 1 Overall Summary ----

# get the overall beta values
model1.avgBetaValOverall <- tibble(variable = c(’screen7’, ’screen10’,’ram16’,’ram32’,’proc2
    ’,’proc2.5’,’price299’,’price399’,’brand.somesong’,’brand.pear’,’brand.gaggle’,’brand.
    price.somesong’,’brand.price.pear’,’brand.price.gaggle’), coefficient = apply(logit.
    model.1$betadraw[,,minVal:maxVal],c(2),mean))

model1.coefficient.table <- model1.avgBetaValOverall %>%
spread(key = variable, value = coefficient) %>%
transmute(
screen5 = 0 - screen7 - screen10,
screen7,
screen10,
ram8 = 0 - ram16 - ram32,
ram16,
ram32,
proc1.5 = 0 - proc2 - proc2.5,
proc2,
proc2.5,
price199 = 0 - price299 - price399,
price299,
price399,
brand.stc = 0 - brand.somesong - brand.pear - brand.gaggle,
brand.somesong,
brand.pear,
brand.gaggle,
brand.price.stc = 0 - brand.price.somesong - brand.price.pear - brand.price.gaggle,
brand.price.somesong,
brand.price.pear,
brand.price.gaggle
) %>%
gather(key = variable, value = coefficient) %>%
mutate(
odds.ratio = exp(coefficient)
)

model1.coefficient.table

model1.coefficient.table %>%
select(variable, coefficient) %>%
ggplot(aes(x = variable, y = coefficient)) +
geom_bar(stat = 'identity') +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
scale_x_discrete(limits = c('screen5','screen7','screen10','ram8','ram16','ram32','proc1
    .5','proc2','proc2.5','price199','price299','price399','brand.stc','brand.somesong','
    brand.pear','brand.gaggle', 'brand.price.stc','brand.price.somesong','brand.price.pear
    ','brand.price.gaggle')) +
labs(x = 'Attribute', y = 'Beta Coefficient') +
ggtitle('Parts Worth - Model 1')







# Model 2 - with covariate ----
owner.status = as.matrix(scale(z.owner$zowner, scale = F))
mcmctest.model2 <- list(R = 150000, keep = 5)
data.list.model2 <- list(lgtdata = lgtdata2, p = 3, Z = owner.status)

set.seed(1234)
logit.model.2 <- rhierMnlDP(Data = data.list.model2, Mcmc = mcmctest.model2)

table(logit.model.2$Istardraw)


dim(logit.model.2$Deltadraw)

model2.deltadraw <-  as.data.frame(logit.model.2$Deltadraw[minVal:maxVal,])

model2.deltadraw <- model2.deltadraw %>%
rename(
screen7 = V1,
screen10 = V2,
ram16 = V3,
ram32 = V4,
proc2 = V5,
proc2.5 = V6,
price299 = V7,
price399 = V8,
brand.somesong = V9,
brand.pear = V10,
brand.gaggle = V11,
brand.price.somesong = V12,
brand.price.pear = V13,
brand.price.gaggle =V14
)


mean.values <- model2.deltadraw %>%
summarize_all(mean) %>%
gather(key = variable, value = mean.value)
min.values <- model2.deltadraw %>%
summarise_all(min) %>%
gather(key = variable, value = minimum.value)
max.values <- model2.deltadraw %>%
summarize_all(max) %>%
gather(key = variable, value = maximum.value)
q05.values <- model2.deltadraw %>%
summarize_all(funs(quantile), probs = 0.05) %>%
gather(key = variable, value = p05.value)
q10.values <- model2.deltadraw %>%
summarize_all(funs(quantile), probs = 0.10) %>%
gather(key = variable, value = p10.value)
q25.values <- model2.deltadraw %>%
summarize_all(funs(quantile), probs = 0.25) %>%
gather(key = variable, value = p25.value)
median.values <- model2.deltadraw %>%
summarize_all(median) %>%
gather(key = variable, value = median.value)
q75.values <- model2.deltadraw %>%
summarize_all(funs(quantile), probs = 0.75) %>%
gather(key = variable, value = p75.value)
q90.values <- model2.deltadraw %>%
summarize_all(funs(quantile), probs = 0.90) %>%
gather(key = variable, value = p90.value)
q95.values <- model2.deltadraw %>%
summarize_all(funs(quantile), probs = 0.95) %>%
gather(key = variable, value = p95.value)

descriptive.stats.model2.DeltaDraw <- mean.values %>%
inner_join(min.values, by = 'variable') %>%
inner_join(max.values, by = 'variable') %>%
inner_join(median.values, by = 'variable') %>%
inner_join(q05.values, by = 'variable') %>%
inner_join(q10.values, by = 'variable') %>%
inner_join(q25.values, by = 'variable') %>%
inner_join(q75.values, by = 'variable') %>%
inner_join(q90.values, by = 'variable') %>%
inner_join(q95.values, by = 'variable')


# p23.betadraws.model2 <- tibble(
#   screen7 = logit.model.2$betadraw[23,1,],
#   screen10 = logit.model.2$betadraw[23,2,],
#   ram16 = logit.model.2$betadraw[23,3,],
#   ram32 = logit.model.2$betadraw[23,4,],
#   proc2 = logit.model.2$betadraw[23,5,],
#   proc2.5 = logit.model.2$betadraw[23,6,],
#   price299 = logit.model.2$betadraw[23,7,],
#   price399 = logit.model.2$betadraw[23,8,],
#   brand.somesong = logit.model.2$betadraw[23,9,],
#   brand.pear = logit.model.2$betadraw[23,10,],
#   brand.gaggle = logit.model.2$betadraw[23,11,],
#   brand.price.somesong = logit.model.2$betadraw[23,12,],
#   brand.price.pear = logit.model.2$betadraw[23,13,],
#   brand.price.gaggle = logit.model.2$betadraw[23,14,]
# )
#
#
# plot(x = seq(1,30000,1), y = cumsum(p23.betadraws.model2$screen7) / seq_along(p23.
    betadraws.model2$screen7), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace
    Plot = Screen7’)
# # seems like at 25000, it goes steady state
# plot(x = seq(1,30000,1), y = cumsum(p23.betadraws.model2$screen10) / seq_along(p23.
    betadraws.model2$screen10), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace
     Plot - Screen10’)
# plot(x = seq(1,30000,1), y = cumsum(p23.betadraws.model2$ram16) / seq_along(p23.betadraws.
    model2$ram16), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace Plot - Ram16
    ’)
# plot(x = seq(1,30000,1), y = cumsum(p23.betadraws.model2$ram32) / seq_along(p23.betadraws.
    model2$ram32), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace Plot - Ram32
    ’)
# plot(x = seq(1,30000,1), y = cumsum(p23.betadraws.model2$proc2) / seq_along(p23.betadraws.
    model2$proc2), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace Plot -
    Processor2’)
# plot(x = seq(1,30000,1), y = cumsum(p23.betadraws.model2$proc2.5) / seq_along(p23.
    betadraws.model2$proc2.5), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace
    Plot - Processor2.5’)
# plot(x = seq(1,30000,1), y = cumsum(p23.betadraws.model2$price299) / seq_along(p23.
    betadraws.model2$price299), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace
     Plot - Price299’)
# plot(x = seq(1,30000,1), y = cumsum(p23.betadraws.model2$price399) / seq_along(p23.
    betadraws.model2$price399), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’Trace
     Plot - Price399’)
# plot(x = seq(1,30000,1), y = cumsum(p23.betadraws.model2$brand.somesong) / seq_along(p23.
    betadraws.model2$brand.somesong), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main =
    ’Trace Plot - Somesong’)
# plot(x = seq(1,30000,1), y = cumsum(p23.betadraws.model2$brand.pear) / seq_along(p23.
    betadraws.model2$brand.pear), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’
    Trace Plot - Pear’)
# plot(x = seq(1,30000,1), y = cumsum(p23.betadraws.model2$brand.gaggle) / seq_along(p23.
    betadraws.model2$brand.gaggle), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main = ’
    Trace Plot - Gaggle’)
# plot(x = seq(1,30000,1), y = cumsum(p23.betadraws.model2$brand.price.somesong) / seq_along
    (p23.betadraws.model2$brand.price.somesong), xlab = ’Iteration’, ylab = ’Cumulative
    Mean’, main = ’Trace Plot - Price & Somesong’)
# plot(x = seq(1,30000,1), y = cumsum(p23.betadraws.model2$brand.price.pear) / seq_along(p23
    .betadraws.model2$brand.price.pear), xlab = ’Iteration’, ylab = ’Cumulative Mean’, main
     = ’Trace Plot - Price & Pear’)
# plot(x = seq(1,30000,1), y = cumsum(p23.betadraws.model2$brand.price.gaggle) / seq_along(
    p23.betadraws.model2$brand.price.gaggle), xlab = ’Iteration’, ylab = ’Cumulative Mean’,
     main = 'Trace Plot - Price & Gaggle')


model2.avgBetaVal <- as.data.frame(apply(logit.model.2$betadraw[,,minVal:maxVal],c(1,2),mean
    ))

model2.avgBetaVal <- model2.avgBetaVal %>%
rename(
screen7 = V1,
screen10 = V2,
ram16 = V3,
ram32 = V4,
proc2 = V5,
proc2.5 = V6,
price299 = V7,
price399 = V8,
brand.somesong = V9,
brand.pear = V10,
brand.gaggle = V11,
brand.price.somesong = V12,
brand.price.pear = V13,
brand.price.gaggle =V14
)

summary(model2.avgBetaVal)

min.values <- model2.avgBetaVal %>%
summarise_all(min) %>%
gather(key = variable, value = minimum.value)
max.values <- model2.avgBetaVal %>%
summarize_all(max) %>%
gather(key = variable, value = maximum.value)
q05.values <- model2.avgBetaVal %>%
summarize_all(funs(quantile), probs = 0.05) %>%
gather(key = variable, value = p05.value)
q10.values <- model2.avgBetaVal %>%
summarize_all(funs(quantile), probs = 0.10) %>%
gather(key = variable, value = p10.value)
q25.values <- model2.avgBetaVal %>%
summarize_all(funs(quantile), probs = 0.25) %>%
gather(key = variable, value = p25.value)
median.values <- model2.avgBetaVal %>%
summarize_all(median) %>%
gather(key = variable, value = median.value)
q75.values <- model2.avgBetaVal %>%
summarize_all(funs(quantile), probs = 0.75) %>%
gather(key = variable, value = p75.value)
q90.values <- model2.avgBetaVal %>%
summarize_all(funs(quantile), probs = 0.90) %>%
gather(key = variable, value = p90.value)
q95.values <- model2.avgBetaVal %>%
summarize_all(funs(quantile), probs = 0.95) %>%
gather(key = variable, value = p95.value)

descriptive.stats.model2.avgBetaVals <- min.values %>%
inner_join(max.values, by = 'variable') %>%
inner_join(median.values, by = 'variable') %>%
inner_join(q05.values, by = 'variable') %>%
inner_join(q10.values, by = 'variable') %>%
inner_join(q25.values, by = 'variable') %>%
inner_join(q75.values, by = 'variable') %>%
inner_join(q90.values, by = 'variable') %>%
inner_join(q95.values, by = 'variable')
```

```
# Model 2 - Fit Statistics (with covariate) ----

model2.xbetas <- cbc2 %*% t(as.matrix(model2.avgBetaVal))

# convert to 3 columns across
model2.xbetas <- matrix(model2.xbetas, ncol = 3, byrow = T)

# take the exponential (remember, it's a log-odds model) of the matrix
model2.xbetas <- exp(model2.xbetas)

# convert to dataframe
model2.xbetas <- as.data.frame(model2.xbetas)
model2.xbetas <- model2.xbetas %>%
rename(
'choice1' = V1,
'choice2' = V2,
'choice3' = V3
) %>%
mutate(
sum.of.row = rowSums(.)
) %>%
transmute(
choice1 = choice1 / sum.of.row,
choice2 = choice2 / sum.of.row,
choice3 = choice3 / sum.of.row
)

model2.choice <- max.col(model2.xbetas)

responses <- as.vector(t(y.data))

table(model2.choice, responses) # confusion matrix

par(mfrow=c(2,2))
auc(multiclass.roc(responses, model2.choice, print.auc = T, levels = c(1,2,3), col = 'blue',
     percent = F, plot = T, main = 'AUC Curve - Model 2')) # AUC
par(mfrow=c(1,1))
auc(multiclass.roc(responses, model2.choice, plot = T, main = 'AUC Curve - Model 2')) # AUC

mean(logit.model.2$loglike) # -2loglike test

model2.predicted.values <- as.data.frame(apply(t(matrix(model2.choice, nrow = 36, ncol =
    424)), 2, function(x){tabulate(na.omit(x))}))

model2.predicted.values %>%
mutate(
alternative = c(1,2,3)
) %>%
gather(key = choice.set, value, starts_with(’V’)) %>%
spread(key = alternative, value = value) %>%
mutate(
choice.set = case_when(
choice.set == ’V1’ ˜ 1,
choice.set == ’V2’ ˜ 2,
choice.set == ’V3’ ˜ 3,
choice.set == ’V4’ ˜ 4,
choice.set == ’V5’ ˜ 5,
choice.set == ’V6’ ˜ 6,
choice.set == ’V7’ ˜ 7,
choice.set == ’V8’ ˜ 8,
choice.set == ’V9’ ˜ 9,
choice.set == ’V10’ ˜ 10,
choice.set == ’V11’ ˜ 11,
choice.set == ’V12’ ˜ 12,
choice.set == ’V13’ ˜ 13,
choice.set == ’V14’ ˜ 14,
choice.set == ’V15’ ˜ 15,
choice.set == ’V16’ ˜ 16,
choice.set == ’V17’ ˜ 17,
choice.set == ’V18’ ˜ 18,
choice.set == ’V19’ ˜ 19,
choice.set == ’V20’ ˜ 20,
choice.set == ’V21’ ˜ 21,
choice.set == ’V22’ ˜ 22,
choice.set == ’V23’ ˜ 23,
choice.set == ’V24’ ˜ 24,
choice.set == ’V25’ ˜ 25,
choice.set == ’V26’ ˜ 26,
choice.set == ’V27’ ˜ 27,
choice.set == ’V28’ ˜ 28,
choice.set == ’V29’ ˜ 29,
choice.set == ’V30’ ˜ 30,
choice.set == ’V31’ ˜ 31,
choice.set == ’V32’ ˜ 32,
choice.set == ’V33’ ˜ 33,
choice.set == ’V34’ ˜ 34,
choice.set == ’V35’ ˜ 35,
choice.set == ’V36’ ˜ 36
)
) %>%
rename(
alt.1 = ’1’,
alt.2 = ’2’,
alt.3 = ’3’
) %>%
arrange(choice.set) %>%
mutate(
alt.1.pct = round(alt.1 / rowSums(.,dims = 1),3),
alt.2.pct = round(alt.2 / rowSums(.,dims = 1),3),
alt.3.pct = round(alt.3 / rowSums(.,dims = 1),3)
) %>%
head(n=36)
```

```
# Model 2 Overall Summary (with covariate) ----

# get the overall beta values
model2.avgBetaValOverall <- tibble(variable = c(’screen7’, ’screen10’,’ram16’,’ram32’,’proc2
    ’,’proc2.5’,’price299’,’price399’,’brand.somesong’,’brand.pear’,’brand.gaggle’,’brand.
    price.somesong’,’brand.price.pear’,’brand.price.gaggle’), coefficient = apply(logit.
    model.2$betadraw[,,minVal:maxVal],c(2),mean))

model2.coefficient.table <- model2.avgBetaValOverall %>%
spread(key = variable, value = coefficient) %>%
transmute(
screen5 = 0 - screen7 - screen10,
screen7,
screen10,
ram8 = 0 - ram16 - ram32,
ram16,
ram32,
proc1.5 = 0 - proc2 - proc2.5,
proc2,
proc2.5,
price199 = 0 - price299 - price399,
price299,
price399,
brand.stc = 0 - brand.somesong - brand.pear - brand.gaggle,
brand.somesong,
brand.pear,
brand.gaggle,
brand.price.stc = 0 - brand.price.somesong - brand.price.pear - brand.price.gaggle,
brand.price.somesong,
brand.price.pear,
brand.price.gaggle
) %>%
gather(key = variable, value = coefficient) %>%
mutate(
odds.ratio = exp(coefficient)
)

model2.coefficient.table


model2.coefficient.table %>%
select(variable, coefficient) %>%
ggplot(aes(x = variable, y = coefficient)) +
geom_bar(stat = ’identity’) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
scale_x_discrete(limits = c(’screen5’,’screen7’,’screen10’,’ram8’,’ram16’,’ram32’,’proc1
    .5’,’proc2’,’proc2.5’,’price199’,’price299’,’price399’,’brand.stc’,’brand.somesong’,’
    brand.pear’,’brand.gaggle’, ’brand.price.stc’,’brand.price.somesong’,’brand.price.pear
    ’,’brand.price.gaggle’)) +
labs(x = ’Attribute’, y = ’Beta Coefficient’) +
ggtitle(’Parts Worth with Prior STC Ownership’)


# Extra Scenarios ----
# Load scenario data

extra.scenario.raw <- read_csv(’data/extra-scenarios-v3.csv’)

extra.scenario.processed <- read_csv(’data/extra-scenarios(1).csv’)

# rename the columsn in processed
extra.scenario.processed <- extra.scenario.processed %>%
rename(
screen7 = V1,
screen10 = V2,
ram16 = V3,
ram32 = V4,
proc2 = V5,
proc2.5 = V6,
price299 = V7,
price399 = V8,
brand.somesong = V9,
brand.pear = V10,
brand.gaggle = V11,
brand.price.somesong = V12,
brand.price.pear = V13,
brand.price.gaggle =V14
)

# preview dataframes
extra.scenario.raw %>%
head

extra.scenario.processed %>%
head

# convert the extra scenarios to a matrix format for prediction purposes

extra.scenario.matrix <- as.matrix(extra.scenario.processed)

# use model 1 for predictions (a pooled approach)

extra.scenario.coefficients <- extra.scenario.matrix %*% as.matrix(model1.avgBetaValOverall
    %>% select(coefficient), ncol = 1, byow = T)

extra.scenario.xbetas<- matrix(extra.scenario.coefficients, ncol=3, byrow = T)

extra.scenario.xbetas <- exp(extra.scenario.xbetas)

extra.scenario.xbetas <- as.data.frame(extra.scenario.xbetas)

extra.scenario.xbetas <- extra.scenario.xbetas %>%
rename(
’choice1’ = V1,
’choice2’ = V2,
’choice3’ = V3
) %>%
mutate(
sum.of.row = rowSums(.)
) %>%
transmute(
choice1 = choice1 / sum.of.row,
choice2 = choice2 / sum.of.row,
choice3 = choice3 / sum.of.row
)

# use model 1 for predictions (individualized)

extra.scenario.ind.xbetas <- extra.scenario.matrix %*% t(as.matrix(model1.avgBetaVal))

extra.scenario.ind.xbetas <- matrix(extra.scenario.ind.xbetas, ncol = 3, byrow = T)

extra.scenario.ind.xbetas <- exp(extra.scenario.ind.xbetas)

extra.scenario.ind.xbetas <- as.data.frame(extra.scenario.ind.xbetas)

extra.scenario.ind.xbetas <- extra.scenario.ind.xbetas %>%
rename(
'choice1' = V1,
'choice2' = V2,
'choice3' = V3
) %>%
mutate(
sum.of.row = rowSums(.)
) %>%
transmute(
choice1 = choice1 / sum.of.row,
choice2 = choice2 / sum.of.row,
choice3 = choice3 / sum.of.row
)

extra.scenario.ind.xbetas

extra.scenario.ind.choice <- max.col(extra.scenario.ind.xbetas)

table(extra.scenario.ind.choice[1:424]) # for the 1st choiceset
table(extra.scenario.ind.choice[425:848]) # for the 2nd choiceset

# use model 2 for predictions (individualized)

extra.scenario.ind.xbetas2 <- extra.scenario.matrix %*% t(as.matrix(model2.avgBetaVal))

extra.scenario.ind.xbetas2 <- matrix(extra.scenario.ind.xbetas2, ncol = 3, byrow = T)

extra.scenario.ind.xbetas2 <- exp(extra.scenario.ind.xbetas2)

extra.scenario.ind.xbetas2 <- as.data.frame(extra.scenario.ind.xbetas2)

extra.scenario.ind.xbetas2 <- extra.scenario.ind.xbetas2 %>%
rename(
'choice1' = V1,
'choice2' = V2,
'choice3' = V3
) %>%
mutate(
sum.of.row = rowSums(.)
) %>%
transmute(
choice1 = choice1 / sum.of.row,
choice2 = choice2 / sum.of.row,
choice3 = choice3 / sum.of.row
)
```

```
extra.scenario.ind.xbetas2

extra.scenario.ind.choice2 <- max.col(extra.scenario.ind.xbetas2)

table(extra.scenario.ind.choice2[1:424]) # for the 1st choiceset
table(extra.scenario.ind.choice2[425:848]) # for the 2nd choiceset
```