BUILDING DESIGNS FOR INDIVIDUAL-LEVEL ESTIMATION: CONSIDERATIONS, IMPLICATIONS AND NEW TOOLS

MEGAN PEITZ
TREVOR OLSEN
NUMERIOUS INC.

ABSTRACT

Choice-based conjoint (CBC) experiments are widely used to understand consumer preferences and willingness to pay for different product features. One important consideration in designing CBC experiments is the balance of attribute levels across the design. Implementing this strategy seeks to give every level an equal chance to influence the respondent's decision in the conjoint design and can work in the majority of cases. However, the authors of this paper were interested in revisiting the work of Huber and Zwerina (1996) to determine if utility balanced designs, a design strategy that trades off on level balance while optimizing which alternatives are paired against each other within tasks, could result in better predictions at the individual level. This paper sets out to explore several different methods of optimizing designs and offers access to an open-source package, built in Julia by the Numerious team, to leverage these different design strategies in the future.

The results from this paper show that utility balanced designs perform well in predicting data from both utility balanced and non-utility balanced designs, and that respondents do not seem to be fatigued by utility balanced designs. This would suggest that utility balanced designs could be a successful strategy depending on the attributes and levels being tested. However, we must caution the user of utility balanced designs as some design strategies may result in sparse data at the interaction level. We also believe that further research is needed to understand the differences in willingness to pay estimates between utility balanced designs and traditional, level balanced designs. It should also be noted that there are several different strategies for creating efficient designs as well as other packages used to generate the designs outside of what is mentioned in this paper. See References for more details.

BACKGROUND

According to Rossi, Allenby, McCulloch (2005) the greatest challenges in marketing are to understand the heterogeneity in preferences. This is why marketing practitioners prefer unit-level hierarchical Bayesian estimates.

To uncover those unit-level estimates, we are often taught to build designs that are level balanced (i.e., within each attribute, each level appears an equal number of times). Implementing this strategy seeks to give every level an equal chance to influence the respondent's decision in the conjoint design and can work in the majority of cases.

But McFadden (1974) shows that the estimated utilities from the model depend not just on which concepts are included in the design, but which concepts are paired against each other. The multinomial logit model (MNL) assumes part-worth utilities are independent of each other (i.e., preference for one level does not depend on the preference for another level). However, certain combinations of attributes and levels can affect the distribution of preferences among respondents. For example, it would not be surprising to see a Ferrari at a $250K price point and a Chevy Volt at $25K. But it wouldn't make much sense to see a Chevy Volt at $125K—so why would we waste observations on combinations that aren't relevant? Because of circumstances like this, we think good designs should not just be a matter of level balance across alternatives, but designs should also be dependent on which alternatives are paired against each other within the CBC tasks.

One could also argue that designs that optimize for the principles above could result in smoothing over the unit-level estimates (Bayesian Shrinkage), muting the individual level preferences and potentially resulting in poorer insights into the true heterogeneity of the marketplace.

So, what is a researcher to do? And is it really that big of a deal if we continue building designs according to these principles?

NON-LEVEL BALANCED AND UTILITY BALANCED DESIGNS

One solution is to use a design that is not level balanced. A non-level balanced design could result in some levels appearing significantly more frequently than others, and some pairs of levels appearing more or less frequently than others. However, this type of design may be useful when the relationships between the attributes and levels are complex, and/or where it's important to test specific interactions between the attributes.

Utility balanced designs are one alternative for a non-level balanced design and it has been shown (Huber and Zwerina, 1996) that designs which include utility balance as one criterion can improve the understanding of aggregate effects.

Utility balanced designs are conjoint designs in which the total utility of each concept shown within a task is as even as possible. Figure 1.1 shows the difference between what a level balanced design might look like versus a utility balanced design.

Figure 1.1: Level Balanced vs. Utility Balanced Designs

Level Balanced
(What most practitioners use today)

Q1. Which of the following would you choose?

	Option 1	Option 2	Option 3
Brand	Chevrolet	BMW	Ferrari
Price	$100K	$50K	$50K

Frequencies

Brand	
Chevrolet	100
BMW	100
Ferrari	100

Price	
$25K	100
$50K	100
$100K	100

Pairwise Frequencies

	$25K	$50K	$100K
Chevrolet	33	34	33
BMW	33	33	34
Ferrari	34	33	33

Utility Balanced

Q1. Which of the following would you choose?

	Option 1	Option 2	Option 3
Brand	Ferrari	BMW	Chevrolet
Price	$100K	$50K	$50K

Frequencies

Brand	
Chevrolet	118
BMW	95
Ferrari	87

Price	
$25K	107
$50K	108
$100K	85

Pairwise Frequencies

	$25K	$50K	$100K
Chevrolet	80	33	5
BMW	25	55	15
Ferrari	2	20	65

In reviewing Figure 1.1, we can see the imbalance of levels within the utility balanced design. This imbalance can cause apprehension among many researchers who would warn that if the prior understanding is misspecified (i.e., people will spend less for a Ferrari and more for a Chevrolet vs. more for a Ferrari and less for a Chevrolet), then the resulting alternative comparisons will be less efficient than a design built with the prior at 0. However, if using the R idefix package (Traets, F., Sanchez, D.G. and Vandebroek, M. 2020), one can balance the prior utilities from a Bayesian perspective. In this approach, users can specify a full distribution of prior knowledge to balance the utility within tasks. This approach should avoid misspecification since more uncertainty is being incorporated into the design. Therefore, we will field a study with a level balanced design for n=50 completes and capture individual preferences with a hierarchical Bayesian model to seed our utility balanced designs.

Another concern around utility balanced designs is that the CBC tasks become too difficult and respondents become fatigued if the choices are too hard. And if the choices are too hard, respondent error outweighs the added design efficiency. To address this concern, we will ask respondents to rate the designs on measures like easy vs. hard, long vs. short, as well as explore completion time, drop-off rates, percentage of bad actors (i.e., cheaters) and ultimately the error around their responses by examining both within-sample and out-of-sample holdouts.

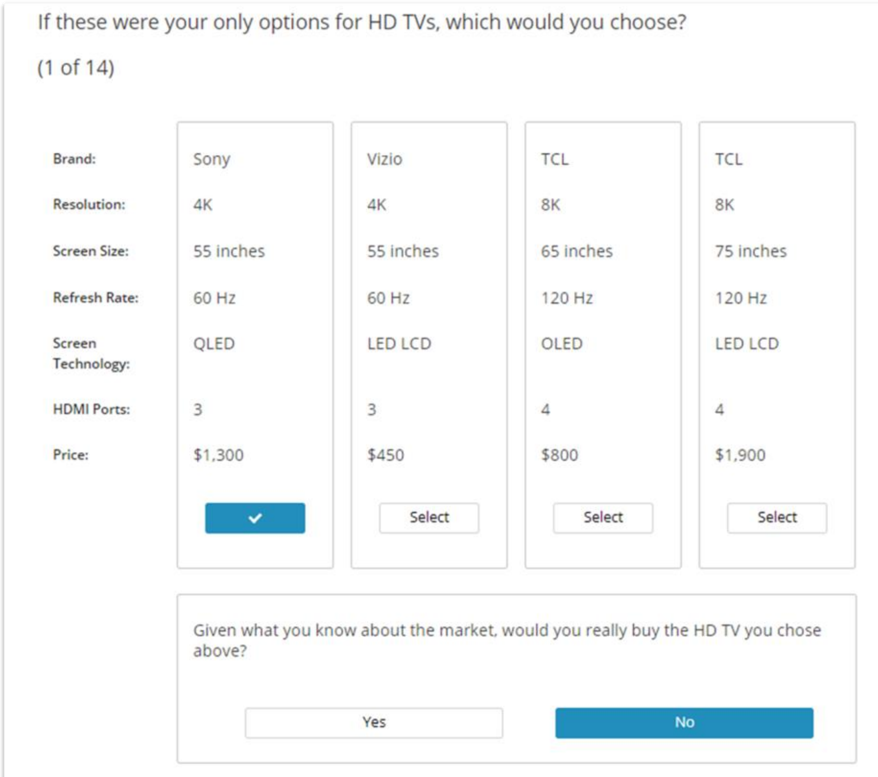
METHODOLOGY

We conducted an online survey about TVs with over 3,500 real respondents. Attributes and levels of the CBC exercise are shown in Figure 2.1 and a screenshot of the conjoint exercise is shown in Figure 2.2.

Figure 2.1: Conjoint Attributes and Levels

Brand	Resolution	Screen Size	Refresh Rate	Screen Technology	HDMI Ports	Price
Sony	4K	55 inches	60 Hz	LED LCD	3	$450
LG	8K	65 inches	120 Hz	QLED	4	$800
Vizio		75 inches		OLED		$1,300
Samsung						$1,900
TCL						$2,700

Figure 2.2: Conjoint Exercise Screenshot



The conjoint experiment consisted of 14 choice tasks, each containing 4 product profiles and a dual-response none alternative. 12 choice tasks were designed by the algorithm (more details below) and 2 of the choice tasks were fixed, meaning that all respondents saw the same combinations of attributes and levels on two screens.

Respondents were assigned to one of eight of the following cells (Figure 2.3) with approximately n=450 completes per cell. All cells had 300 versions of the design except Cell 6 which is a 1 version design used for out-of-sample holdout validation.

Figure 2.3: Respondent Cells

Cell 1	Lighthouse Balanced and Overlap	Level Balanced
Cell 2	Discover 2.0	Level Balanced
Cell 3	Julia no balance penalty	Level Balanced
Cell 4	Julia balance penalty	Level Balanced
Cell 5A	Julia utility balanced with a balanced penalty (based on N=50 from cell 4)	Not Level Balanced
Cell 5B	Julia utility balanced with a balanced penalty (based on N=50 from cell 4 and utilities are 50% of their original size)	Not Level Balanced
Cell 5C	Julia utility balanced with NO balanced penalty (based on N=50 from cell 4)	Not Level Balanced
Cell 6	*Lighthouse Balanced and Overlap (1 version - used for out-of-sample holdouts only)*	Level Balanced

Sawtooth Software is leveraged to create the designs in Cells 1, 2, and 6. You can read more about the design algorithms used in Lighthouse and Discover on Sawtooth Software's website (www.sawtoothsoftware.com). The Numerious cells were built in Julia (https://julialang.org/), an open-source programming language particularly suited for computational math.

The five Julia cells vary based on whether there is a balance penalty, whether there is a utility prior and whether that utility prior will be scaled.

The goal of the cells that have no balance penalty is to minimize D-error versus those with a balance penalty will trade-off minimizing the D-error in order to obtain more level balance. The three cells that have a utility prior leverage a hierarchical Bayesian model built from the first n=50 to respond to Cell 4. Then, within those utility prior cells, we will either trust the priors entirely and allow them to be 100% of their original size or we will shrink them to 50% of their original size. A high-level overview is below in Figure 2.4.

Figure 2.4: Overview of Julia/Numerious Design Cells

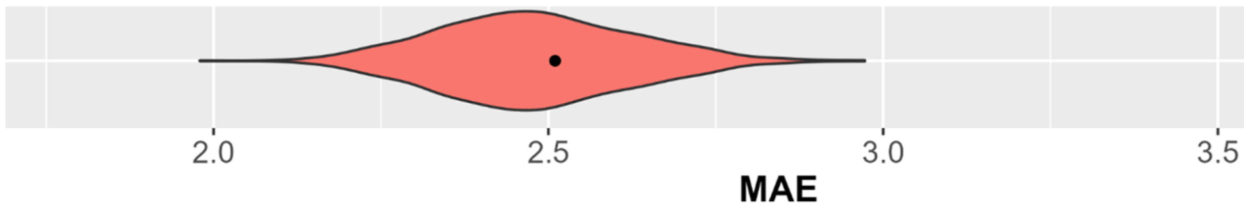
	Balance Penalty	Utility Prior	Utility Scaling
Cell 3	No	0	
Cell 4	Yes	0	
Cell 5A	Yes	Based on N=50 from cell 4	100% of their original size
Cell 5B	Yes	Based on N=50 from cell 4	50% of their original size
Cell 5C	No	Based on N=50 from cell 4	100% of their original size

RESULTS

To measure the accuracy of the models built from each of the different design cells, we will explore the mean absolute error (MAE) of the models. Calculating the MAE involves assessing the average absolute difference between the predicted values from the hierarchical Bayesian model and the actual values in a set of test data, usually referred to as the holdout tasks. The larger the magnitude, the worse the model does at predicting the actuals—so the smaller MAE the better.

Historically, MAEs have been calculated using the point estimate from the HB model, which is typically calculated by taking the average value of all the draws. However, the whole point of using a Bayesian approach is to capture the uncertainty in the data. Thus, to calculate the MAE for this paper, we will leverage 1,000 draws from the HB model and create 1,000 MAEs. Then we will plot the distribution of the MAEs using a violin plot. For posterity's sake, we will also plot the MAE based on the point estimate on the distribution chart. However, one will see in the following results the risks of only using point estimates to run analysis. In the example below (Figure 3.1), you can see the distribution of the 1,000 MAEs in red and the black dot on the chart represents the point estimate MAE.

Figure 3.1: Example MAE Distribution with Point Estimate MAE Included



Note—The point estimate MAE is calculated based on each individual's point estimate part-worth, which is the average of their posterior distribution. After finding the average of the posterior, we exponentiate each individual's point estimate, simulate the fixed task and report the average probability of choice and then report the difference from the stated frequencies. However, the point estimates can get distorted particularly when constraining price (a non-linear transformation) and the IIA property could also distort where the point estimate lies. Because of these two effects (1. order/sequence of averaging with non-linear transformation [before vs. after and within vs. across draws] and 2. IIA property of logit probabilities) the point estimate MAES (i.e., the black dots) are not required to be within the middle of the distribution and can even be found outside the distribution.

Unconstrained Model Results

First modeling the data unconstrained, we can see that the distribution of MAEs for Cell 1 is further to the left in Figure 4.1 suggesting that it is the top performing design strategy when trying to predict responses from Cell 6. Figure 4.2 shows the likelihood that Cell 1 is better than the other cells (i.e., what percentage of the distribution of MAEs does not overlap with other cells). For example, Cell 1 is 87% likely to be better than Cell 2 and 100% likely to be better than Cell 3 and Cell 4 (i.e., there is no overlap in Cell 1's MAE distribution with Cell 3 and 4). Relative to Cell 1, it does appear that Cells 5b and 5c show promise.

Figure 4.1: Distribution of MAEs for Unconstrained Models when Predicting Cell 6

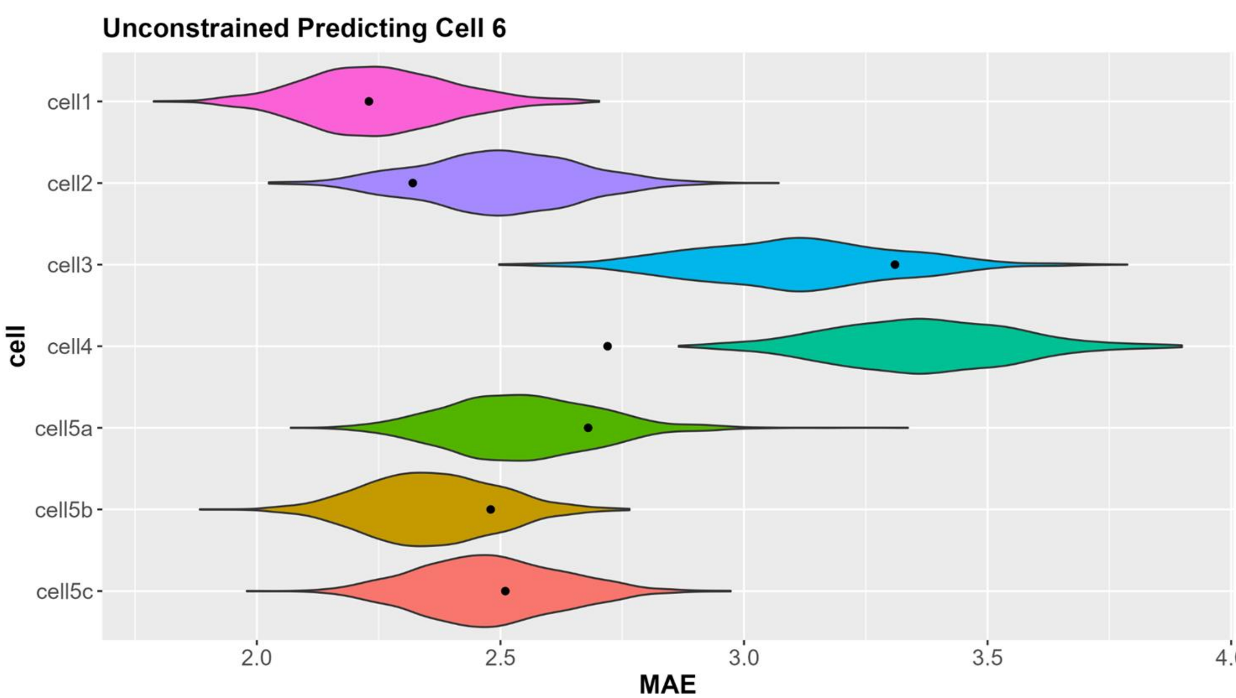


Figure 4.2: Likelihood of a Cell’s MAE Outperforming the Other Cells when Predicting Cell 6

cell	Utility prior	Software	Type	cell1	cell2	cell3	cell4	cell5a	cell5b	cell5c
1	zero	Sawtooth Lighthouse	Balanced and Overlap		87%	100%	100%	92%	71%	86%
2	zero	Sawtooth Discover 2.0	Default	13%		99%	100%	58%	25%	46%
3	zero	Julia	Min D-error	0%	1%		83%	2%	0%	1%
4	zero	Julia	Level Balanced Min D-error	0%	0%	17%		0%	0%	0%
5a	Utilities from 1st 50 Cell 4	Julia	Utility & Level Balanced Min D-error	8%	42%	98%	100%		18%	38%
5b	Utilities from 1st 50 Cell 4	Julia	Utility & 50% Level Balanced Min D-error	29%	75%	100%	100%	82%		72%
5c	Utilities from 1st 50 Cell 4	Julia	Utility & Min D-error	13%	54%	99%	100%	62%	28%	

It should also be noted that if one were to use the point estimate (black dot) instead of the draws, a researcher might come to very different conclusions. For example, they might claim that Cell 4 is significantly better than Cell 3—which, when looking at the draws, we know is not the case. Therefore, practitioners should remain cautious when drawing conclusions based only on the point estimate.

In addition to predicting Cell 6, we can take the data from Cell 1 and predict the two holdouts in Cell 2, 3, 4, 5a, 5b, and 5c. Then we can take the data from Cell 2 and predict the two holdouts in Cell 1, 3, 4, 5a, 5b, and 5c and so on.

Figure 4.3: Distribution of MAEs for Unconstrained Models when Predicting All Other Cells

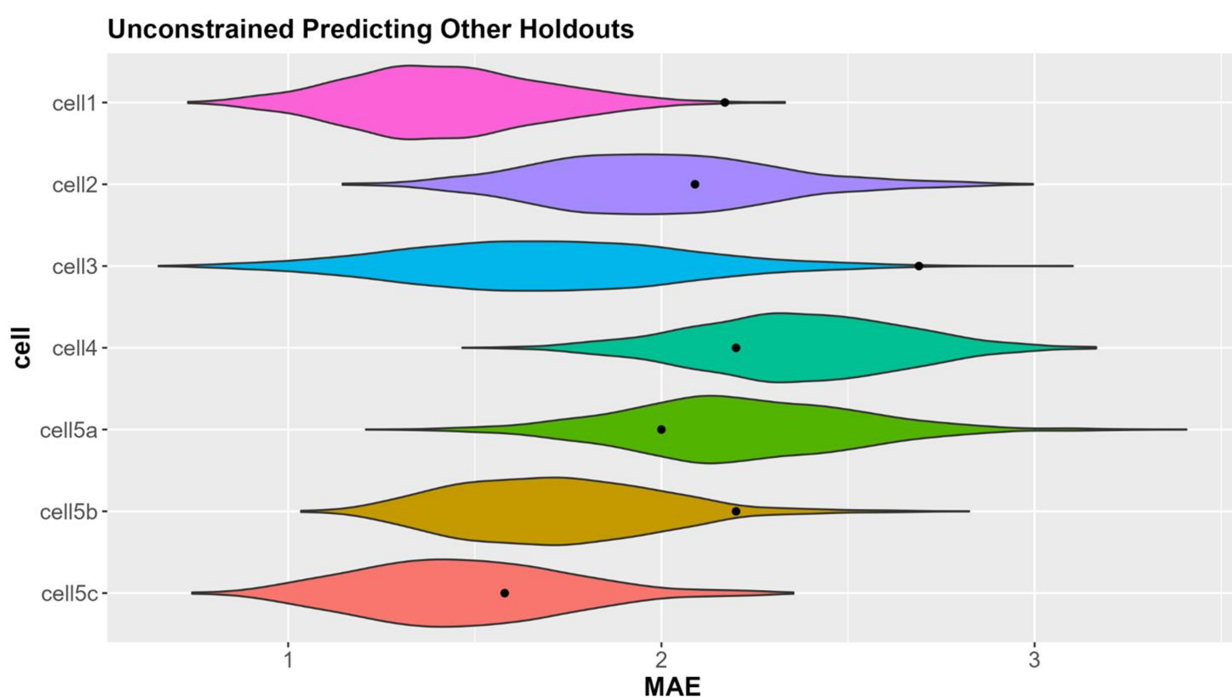


Figure 4.4: Likelihood of a Cell's MAE Outperforming the Other Cells when Predicting All Other Cells

cell	Utility prior	Software	Type	cell1	cell2	cell3	cell4	cell5a	cell5b	cell5c
1	zero	Sawtooth Lighthouse	Balanced and Overlap		92%	72%	99%	97%	78%	55%
2	zero	Sawtooth Discover 2.0	Default	8%		27%	81%	69%	25%	11%
3	zero	Julia	Min D-error	28%	73%		92%	86%	52%	32%
4	zero	Julia	Level Balanced Min D-error	1%	19%	8%		35%	5%	1%
5a	Utilities from 1st 50 Cell 4	Julia	Utility & Level Balanced Min D-error	3%	31%	14%	65%		12%	4%
5b	Utilities from 1st 50 Cell 4	Julia	Utility & 50% Level Balanced Min D-error	22%	75%	48%	95%	88%		27%
5c	Utilities from 1st 50 Cell 4	Julia	Utility & Min D-error	45%	89%	68%	99%	96%	73%	

Figure 4.3 and 4.4 show Cell 1 continuing to outperform the other cells with Cell 4 being the worst at predicting out-of-sample data. (However, if only examining point estimates, one would conclude that Cell 5c is significantly better than Cell 1 when predicting the holdouts in all other cells).

Constrained Model Results

Given that practitioners may use constraints in their model to avoid unrealistic utility estimates (i.e., high prices preferred to low prices vs. low prices preferred to high prices), we also wanted to explore the results when price is constrained to be negative. It is important to note that the application of constraints in a conjoint model should be carefully considered as constraints introduce assumptions or biases into the analysis. Constraints should align with the underlying business context and be based on informed judgments. Proper validation and sensitivity analysis should also be conducted to ensure that the imposed constraints do not overly restrict the model or compromise its predictive power.

Figure 5.1: Distribution of MAEs for Constrained Models when Predicting Cell 6

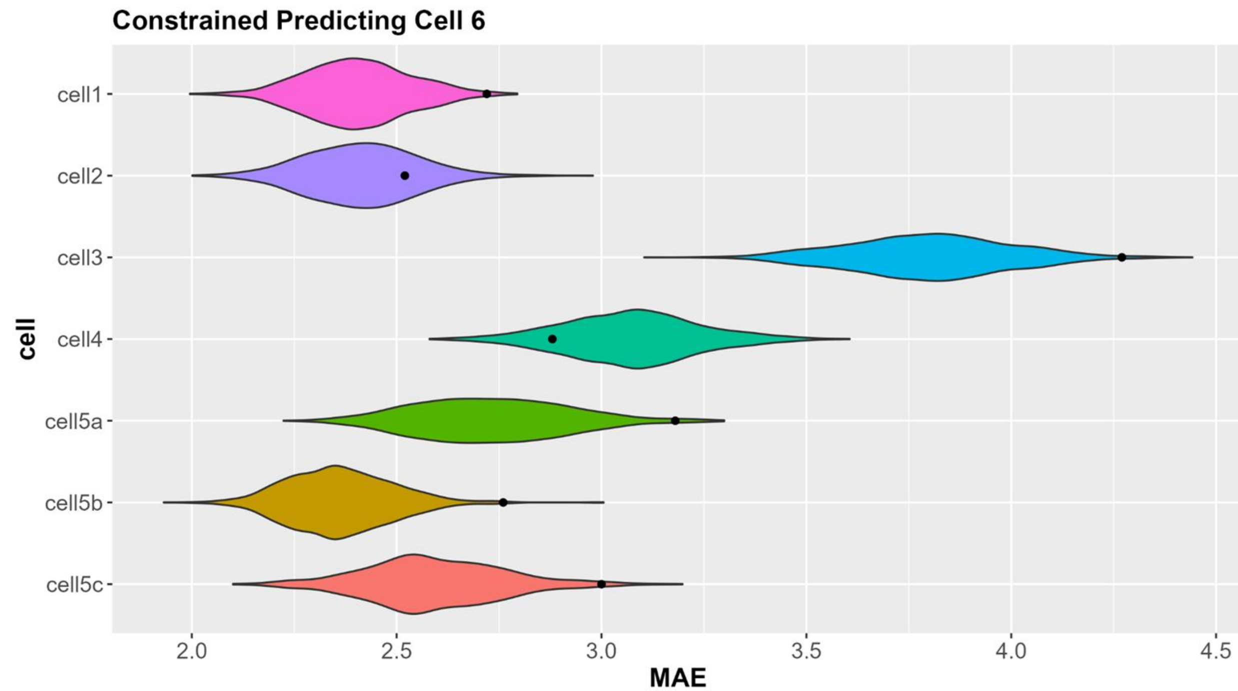


Figure 5.2: Likelihood of a Cell's MAE Outperforming the Other Cells when Predicting Cell 6

cell	Utility prior	Software	Type	cell1	cell2	cell3	cell4	cell5a	cell5b	cell5c
1	zero	Sawtooth Lighthouse	Balanced and Overlap		52%	100%	100%	93%	40%	81%
2	zero	Sawtooth Discover 2.0	Default	48%		100%	100%	92%	39%	80%
3	zero	Julia	Min D-error	0%	0%		0%	0%	0%	0%
4	zero	Julia	Level Balanced Min D-error	0%	0%	100%		9%	0%	3%
5a	Utilities from 1st 50 Cell 4	Julia	Utility & Level Balanced Min D-error	7%	8%	100%	91%		5%	29%
5b	Utilities from 1st 50 Cell 4	Julia	Utility & 50% Level Balanced Min D-error	60%	61%	100%	100%	95%		86%
5c	Utilities from 1st 50 Cell 4	Julia	Utility & Min D-error	19%	20%	100%	97%	71%	14%	

In the constrained models, Cells 1, 2, and 5b do very well with Cell 5c not far behind when predicting Cell 6 data (Figure 5.1). There is more overlap in the performance of Cells 1, 2 and 5b (Figure 5.2) suggesting that a utility balanced design is a viable option when constraints are needed.

Similar to the unconstrained model, when predicting all other cells combined, we see Cell 1 and Cell 5c as the best performers (Figure 5.3, 5.4). In all options, Cells 3 and 4 perform the worst but overall the MAE distributions are still relatively low (<4).

Figure 5.3: Distribution of MAEs for Constrained Models when Predicting All Other Cells

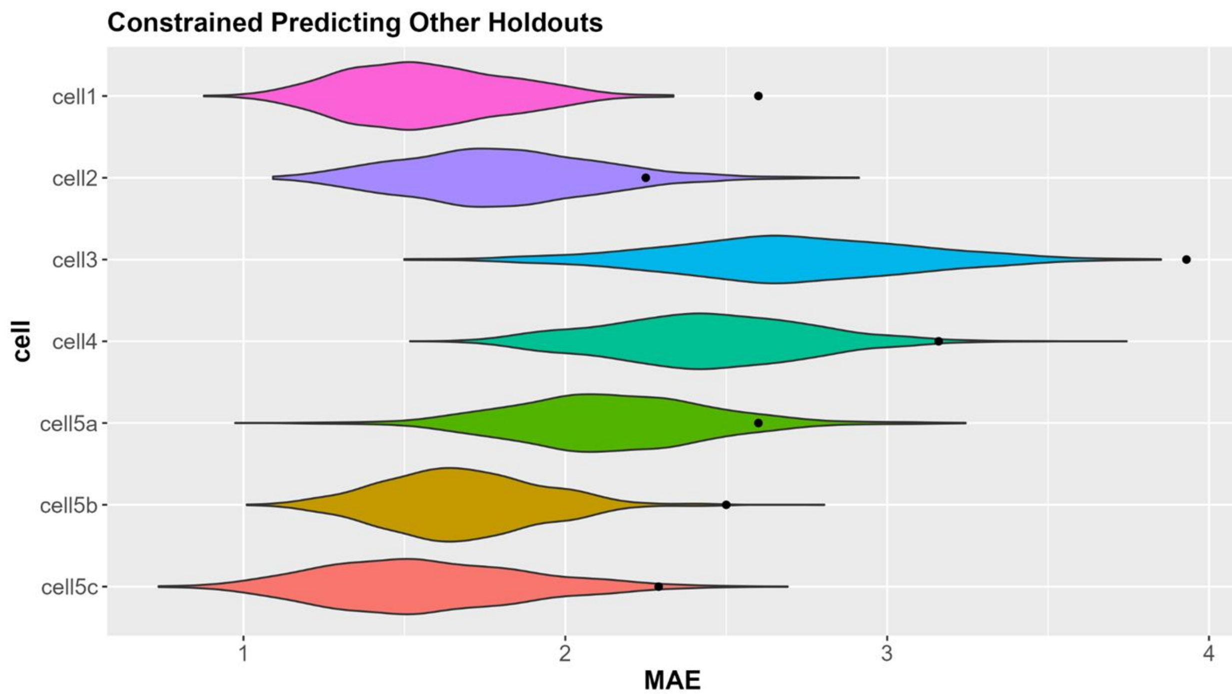


Figure 5.4: Likelihood of a Cell's MAE Outperforming the Other Cells when Predicting All Other Cells

cell	Utility prior	Software	Type	cell1	cell2	cell3	cell4	cell5a	cell5b	cell5c
1	zero	Sawtooth Lighthouse	Balanced and Overlap		72%	99%	99%	93%	63%	49%
2	zero	Sawtooth Discover 2.0	Default	28%		96%	93%	79%	38%	29%
3	zero	Julia	Min D-error	1%	4%		30%	13%	2%	1%
4	zero	Julia	Level Balanced Min D-error	1%	7%	70%		25%	3%	3%
5a	Utilities from 1st 50 Cell 4	Julia	Utility & Level Balanced Min D-error	7%	21%	87%	75%		12%	10%
5b	Utilities from 1st 50 Cell 4	Julia	Utility & 50% Level Balanced Min D-error	37%	62%	98%	97%	88%		38%
5c	Utilities from 1st 50 Cell 4	Julia	Utility & Min D-error	51%	71%	99%	97%	90%	62%	

Respondent Reactions to the Different Cells

From a model standpoint, utility balanced designs, particularly Cells 5b and 5c, seem to be viable options when creating designs that can perform on par with Sawtooth Software designs for this dataset. Next, we want to address how respondents might feel about the different designs.

Using a 5-point semantic differential, we asked respondents to rate the experiment on different dimensions (i.e., long vs. short, difficult vs. easy). Overall, all cells were easy, enjoyable, and appealing (Figure 6.1). Double-clicking into the "easy vs. difficult," we see that over two-thirds of respondents classified the exercise as "easy" regardless of what cell they were in (Figure 6.2). Therefore, one could conclude that for this survey, for these respondents, a utility balanced design is no more difficult than a traditional design.

In Figure 6.3, we can see some additional metrics such as length of interview (LOI) and drop-off rate. LOI does appear to be higher for the utility balanced cells as respondents no longer have tasks that are “no brainers”—but that does not seem to impact respondent opinion, error, or drop-off rates.

Figure 6.1: Respondent Ratings (Means) of Cell Experience

cell	Utility prior	Software	Type	Long vs. Short	Difficult vs. Easy	Unappealing vs. Appealing	Dull vs. Fun	Unenjoyable vs. Enjoyable
1	zero	Sawtooth Lighthouse	Balanced and Overlap	3.4	4.4	4.2	3.9	4.2
2	zero	Sawtooth Discover 2.0	Default	3.5	4.4	4.3	4.1	4.3
3	zero	Julia	Min D-error	3.5	4.4	4.2	4.1	4.3
4	zero	Julia	Level Balanced Min D-error	3.5	4.4	4.2	4.0	4.3
5a	Utilities from 1st 50 Cell 4	Julia	Utility & Level Balanced Min D-error	3.5	4.4	4.2	4.0	4.2
5b	Utilities from 1st 50 Cell 4	Julia	Utility & 50% Level Balanced Min D-error	3.5	4.4	4.2	4.1	4.3
5c	Utilities from 1st 50 Cell 4	Julia	Utility & Min D-error	3.4	4.4	4.2	4.0	4.1
6	zero	Sawtooth Lighthouse	Balanced and Overlap (1 version)	3.5	4.4	4.1	4.0	4.2

Figure 6.2: Frequencies of Easy versus Difficult by Cell

cell	Utility prior	Software	Type	Easy 1	2	3	4	Difficult 5
1	zero	Sawtooth Lighthouse	Balanced and Overlap	68%	12%	14%	2%	5%
2	zero	Sawtooth Discover 2.0	Default	69%	12%	13%	2%	4%
3	zero	Julia	Min D-error	66%	18%	12%	1%	3%
4	zero	Julia	Level Balanced Min D-error	68%	12%	13%	3%	5%
5a	Utilities from 1st 50 Cell 4	Julia	Utility & Level Balanced Min D-error	67%	15%	12%	2%	5%
5b	Utilities from 1st 50 Cell 4	Julia	Utility & 50% Level Balanced Min D-error	68%	13%	13%	2%	4%
5c	Utilities from 1st 50 Cell 4	Julia	Utility & Min D-error	67%	12%	13%	3%	4%
6	zero	Sawtooth Lighthouse	Balanced and Overlap (1 version)	68%	12%	12%	2%	5%

Figure 6.3: Additional Metrics Captured by Cell

cell	Utility prior	Software	Type	Median LOI mins	% LT 10 min	% GT 10 min	Drop-off Rate
1	zero	Sawtooth Lighthouse	Balanced and Overlap	9.18	57%	43%	4.4%
2	zero	Sawtooth Discover 2.0	Default	9.95	51%	49%	3.7%
3	zero	Julia	Min D-error	9.55	54%	46%	5.2%
4	zero	Julia	Level Balanced Min D-error	9.07	58%	42%	3.6%
5a	Utilities from 1st 50 Cell 4	Julia	Utility & Level Balanced Min D-error	10.28	48%	52%	3.0%
5b	Utilities from 1st 50 Cell 4	Julia	Utility & 50% Level Balanced Min D-error	9.77	52%	48%	4.0%
5c	Utilities from 1st 50 Cell 4	Julia	Utility & Min D-error	10.08	50%	50%	4.1%
6	zero	Sawtooth Lighthouse	Balanced and Overlap (1 version)	9.72	53%	47%	6.2%

Comparing Predictions of Utility vs. Non-Utility Balanced Designs

One additional finding to be discussed is the ability of the model from a utility balanced design to predict responses from non-utility balanced design cell. In Figure 7.1 we can see that Cells 5a, 5b and 5c have low MAEs when predicting other utility balanced cells and non-utility balanced cells (Cell 1, 2, 3, and 4). However, the non-utility balanced cells (Cell 1, 2, 3, and 4) struggle to predict utility balanced cells (5b and 5b).

Figure 7.1: Average MAEs per Cell when Predicting Other Cells

cell	Utility prior	Software	Type	cell1	cell2	cell3	cell4	cell5a	cell5b	cell5c	cell6	Average	insample
1	zero	Sawtooth Lighthouse	Balanced and Overlap		2.06	1.49	2.14	4.10	2.68	3.00	2.44	2.56	2.06
2	zero	Sawtooth Discover 2.0	Default	2.41		2.20	1.97	3.16	3.05	2.83	2.62	2.60	1.69
3	zero	Julia	Min D-error	2.53	2.46		2.52	4.58	3.31	3.33	3.10	3.12	1.55
4	zero	Julia	Level Balanced Min D-error	2.73	2.07	2.02		3.48	2.67	3.12	2.79	2.70	1.70
5a	Utilities from 1st 50 Cell 4	Julia	Utility & Level Balanced Min D-error	2.64	2.54	2.25	2.35		2.21	1.81	2.99	2.40	2.71
5b	Utilities from 1st 50 Cell 4	Julia	Utility & 50% Level Balanced Min D-error	2.29	2.68	2.14	2.43	3.55		2.74	2.21	2.58	2.27
5c	Utilities from 1st 50 Cell 4	Julia	Utility & Min D-error	2.12	2.38	1.73	1.95	3.17	1.87		2.01	2.18	2.06

This makes us wonder if people are responding to the choice exercises differently, relative to which cell they are in. One hypothesis is that by using a utility balanced design, we might be priming people to answer the fixed tasks differently than they would if it were a standard design. Initial exploration seems to suggest that the utility balanced cells are potentially using the none alternative differently.

If we look at a different study and compare a Lighthouse Studio design (Cell 1) to a Julia, utility balanced design, we can see that when the none is excluded (Figure 7.2), the Julia cells (JL) perform much better than the Lighthouse Studio cells (LH). But, when we include the none in the model (Figure 7.3), the Lighthouse Studio design cells do much better than the Julia cells.

Figure 7.2: Comparing MAEs of a Lighthouse Design versus a Julia Utility Balanced Design, Excluding the None Option

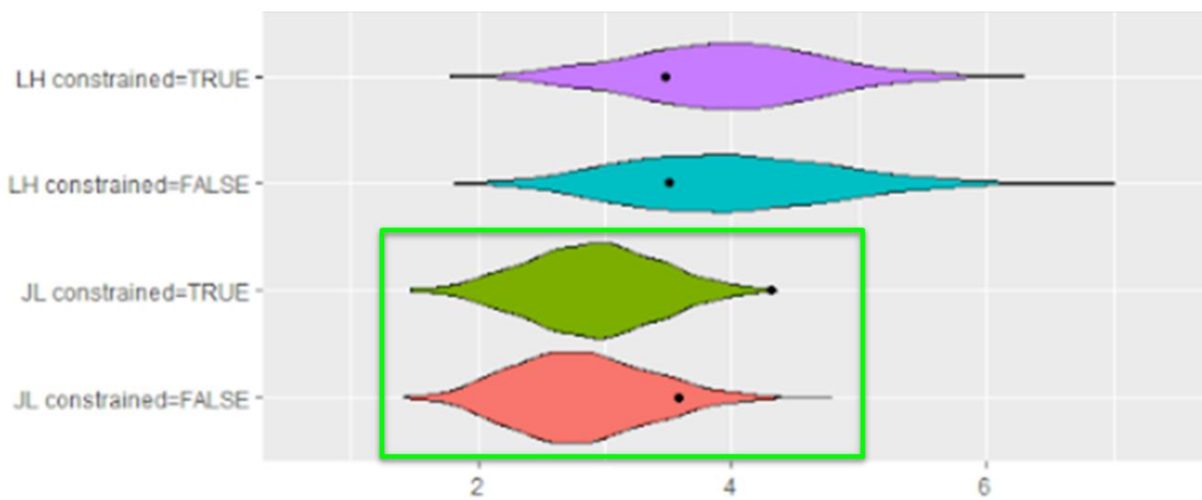
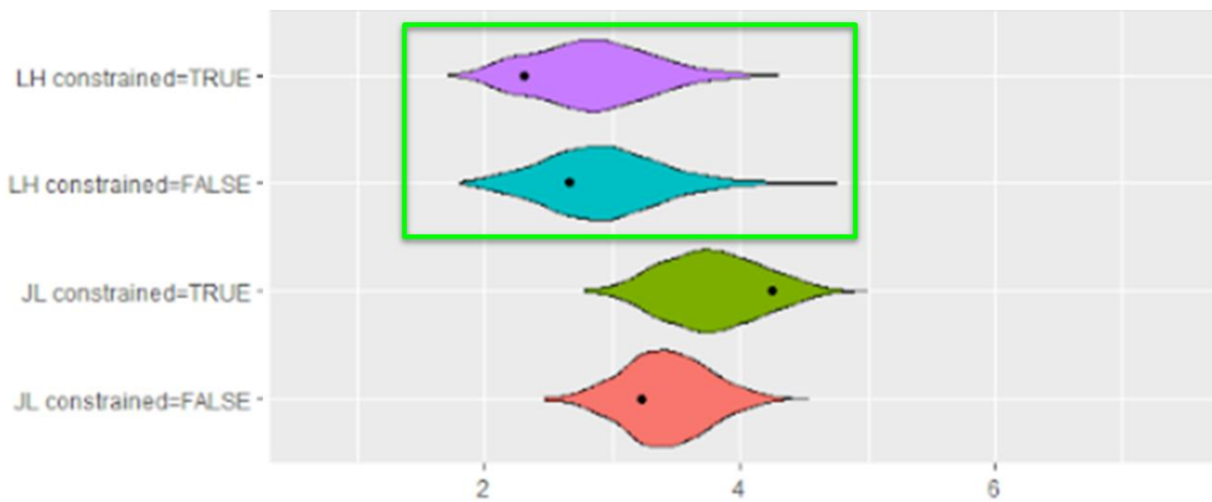


Figure 7.3: Comparing MAEs of a Lighthouse Design versus a Julia Utility Balanced Design, Including the None Option



To explore the potential influence of the design alternatives on respondent behavior, we examined the willingness-to-pay (WTP) values for each design. We hypothesized that if the design was influencing respondent behavior, we might see differences in WTP values between the designs. Our results showed that for one product, all four designs produced similar WTP estimates. However, for another product, there were significant differences between the Lighthouse and Julia designs. This finding suggests that further research is necessary to uncover the factors that may be driving these differences.

Conclusion

In this study, we investigated the use of utility balance designs in CBC experiments. Our results suggest that leveraging estimated prior utilities to inform the CBC design can be valuable, particularly if the researcher plans to constrain the model. The authors encourage other practitioners to test out this approach by exploring the literature around utility balanced and/or two-stage designs (additional references can be found in References) as well as reaching out should you choose to leverage Numerious' Julia designer package.

However, further research is necessary to explore the potential influence of the design on respondent behavior and to uncover the factors that may be driving differences in WTP estimates between designs. In addition, caution should be exercised when using utility balance designs, as they may result in sparse data at the interaction level. In these cases, a standard balance and overlap design or an alternative specific design may be more appropriate.

Next Steps

If you are curious to try out a utility balanced design, reach out to the authors of this paper. We've created a private GitHub repo and welcome anyone that wants access upon request. We've only just scratched the surface on features and would love to build a more robust, open source tool together.

Megan Peitz

Trevor Olsen

REFERENCES

Bliemer, Michiel C.J. & John M. Rose (2011) "Experimental design influences on stated choice outputs: An empirical study in air travel choice," Transportation Research Part A: Policy and Practice, Volume 45, Issue 1, Pages 63–79. ISSN 0965-8564, https://doi.org/10.1016/j.tra.2010.09.003.

Huber, Joel and Klaus B. Zwerina (1996) "The Importance of Utility Balance in Efficient Choice Designs," Journal of Marketing Research 33 (August), 307–17.

Kuhfeld, Warren, (1997) "Efficient Experimental Designs Using Computerized Searches," Sawtooth Software Conference Proceedings, 71–86.

McFadden, Daniel (1974), "Conditional Logit Analysis of Qualitative Choice Behavior," in Frontiers of Econometrics, P. Zarembka, ed. New York: Academic Press, 105–42.

Rose, John M. and Michiel C. J. Bliemer (2009) "Constructing Efficient Stated Choice Experimental Designs," Transport Reviews, 29:5, 587–617, DOI: 10.1080/01441640902827623.

Rossi, P. E., Allenby, G. M., and McCulloch, R. (2005). Bayesian Statistics and Marketing. Wiley-Blackwell.

Sawtooth Software (2018) "Discover-CBC: How and Why It Differs from Lighthouse Studio's CBC Software," https://content.sawtoothsoftware.com/assets/0684c377-88bf-4478-8535-e3841079afba

Traets, F., Sanchez, D.G. and Vandebroek, M. (2020) "Generating Optimal Designs for Discrete Choice Experiments in R: The idefix Package," Journal of Statistical Software, 96, 3 (Nov. 2020), 1–41. DOI: https://doi.org/10.18637/jss.v096.i03.

Walker, J.L., Y. Wang, M. Thorhauge and M. Ben-Akiva (2018) "D-efficient or Deficient? A Robustness Analysis of Stated Choice Experimental Designs," Theory and Decision, 84: 215–238.