Sawtooth Software

RESEARCH PAPER SERIES

Summed Pricing CBC Experiments and How to Do Them

Bryan Orme
Sawtooth Software, Inc.

3210 N. Canyon Rd., Provo, Utah
+1 801 477 4700
www.sawtoothsoftware.com

Summed Pricing CBC Experiments and How to Do Them

Bryan Orme, 2023[1]

Background

There are multiple ways to treat the price attribute in Choice-Based Conjoint (CBC) studies and some lead to more realistically priced product profiles than others. Four common approaches are:

1. One generic price attribute with typically three to seven levels
2. Alternative-specific (nested) price attributes, e.g., one price attribute for each brand in the study
3. Conditional price attribute, using a generic price attribute (with typically 3-7 low to high price levels) that is replaced with an absolute price from a lookup table depending on triggering attributes such as brand, package size, or premium features
4. Summed pricing attribute, where the total price shown is a function of fixed additive component prices associated with some or all the attributes in the study, perturbed by a random shock (typically in the +/-30% range)

Sawtooth Software's Lighthouse Studio CBC software directly supports approaches 1, 2, and 3. Adaptive CBC (ACBC) directly supports approaches 1, 2, and 4. Some users want to do Summed Pricing (approach 4) in CBC. The software doesn't yet have this feature, but this document describes a straightforward power trick to make it happen.

Why use Summed Pricing in CBC rather than the traditional pricing approach that has one generic price attribute that applies to all product concepts? There are several reasons, including:

- Better product profile realism. Respondents see prices for varying products in the CBC questionnaire that better approximate what they see in the real world. We're trying to mimic the buying experience so that respondents give us as realistic choice data as possible that more closely aligns with what they would do with real purchases.
- We concentrate the design space a bit more tightly around product concepts that would realistically be offered in the real world. This may help improve predictions of products that we're more likely to include in our market simulations of base case and variations from base case.

Other pricing approaches in CBC also address these potential benefits, such as conditional pricing and alternative-specific designs (Orme 2007). Some pros and cons of using the different pricing approaches in CBC are shown in Exhibit 1.

---

[1] This document is a 2024 update of the original 2023 document.

Exhibit 1

Approach	Pros	Cons
1. Generic Price	Simple to do and relatively few utility parameters for price to estimate.	Sometimes not very realistic presentation of product prices, as products with excellent features are frequently shown at the lowest price (and vice-versa) Prohibitions to try to show more realistic prices based on levels of other attributes can have a very negative effect on design efficiency.
2. Alternative-Specific Price Attributes	Greater market realism as you can customize price ranges based typically on one or two other attributes such as brand or package size. Allows you to fit different price sensitivity functions for, say, different brands or package sizes. Tends to reduce the number of "no-brainer" dominating alternatives in choice tasks.	Greater number of utility parameters to estimate due to having more than one price attribute; can become excessive, leading to overfitting. You need to decide ahead of time which attribute levels drive the nested alternative-specific pricing structure. Typically is limited to one or two triggering attributes.
3. Conditional Price	A relatively few utility parameters for price to estimate; but be on the lookout for need to estimate interaction effects involving price. Allows you to customize price ranges shown based on one or multiple attributes. Tends to reduce the number of "no-brainer" dominating alternatives in choice tasks.	Non-price attribute utilities no longer can be interpreted independently of price. For example, the premium SKU utility is confounded with the average shift in price for premium SKU versus discount SKU. If price is made conditional on more than one other attribute, this could cause potential significant 2nd (or higher) order interaction effects (an interaction among 3 or more attributes), which the software doesn't allow you to model. Even if you could model them, the data may be too sparse to do it well. If price lookup entries in the conditional pricing table do not follow formulaic (such as proportional) price changes applied the same way across attributes, you may not be able to fit a proper model using generic price plus interactions to explain the price effect for combinations of triggering attributes.
4. Summed Pricing	Allows you to customize price ranges shown based on one or multiple attributes. Tends to reduce the number of "no-brainer" dominating alternatives in choice tasks. Non-price attribute utilities are estimated independently of price, so interpretation of utilities is as easy as method 1.	Not automatically supported by Sawtooth Software's CBC program, so a power trick needs to be done (described in this document). Causes the price attribute to be correlated with other attributes in the study; if random shock selected by the researcher is not sufficient, too much correlation can cause poor precision of utility estimates for both price and the other attributes.

Thus, Summed Pricing can be thought of as a flexible extension of the Conditional Pricing approach, without so many pitfalls.

Although avoiding "no-brainer" (dominating) product alternatives seems like it should be a good thing in CBC questionnaires, utility estimation routines don't know ahead of time whether low price levels are preferred to high price levels. The algorithm needs to learn that from the data (Orme and Chrzan, 2017). When respondents see what seem like "no-brainer" choice tasks, we're still gaining useful information when they answer the way we'd expect them to. That said, too many dominating choice alternatives should be avoided and summed pricing provides a way to achieve a bit better utility balance and fewer dominating alternatives than CBC designs that use a single generic price.

Example of Summed Price CBC

Consider the following attribute levels for HDTVs, first with just a single generic price attribute for illustration:

Exhibit 2

Brand:	Size:	Resolution:	Price:
Samsung	55-inch	4K	$300
LG	65-inch	8K	$500
Sony	75-inch		$800
			$1200

If we use these four generic prices, sometimes 75-inch 8K HDTVs will be shown at $300. Sometimes 55-inch 4K HDTVs will be shown at $1200. Neither of those offerings are very realistic to the marketplace. It's possible to set up some prohibitions to keep from showing respondents such unrealistic HDTV options, but too many prohibitions will start to damage design efficiency and the precision of the utility estimates can suffer tremendously (both for price and non-price attributes).

The Summed Pricing approach starts with fixed price increments you assign to different attribute levels. (It's not necessary that these be exact to match market pricing; as long as we're reasonably close it will generally do the job by creating more realistic tradeoffs for respondents to consider.)

Exhibit 3
Summed Pricing Table

Attribute	Level	Incremental Price
Brand:	Samsung	+$100
	LG	+$0
	Sony	+$75
Size:	55-inch	+$250
	65-inch	+$350
	75-inch	+$500
Resolution:	4K	+$0
	8K	+$300

After establishing the Summed Pricing Table, we decide on the amount of random shock to give the summed prices for HDTV. If we don't shock the summed prices with enough random variation, the total price attribute will be too highly correlated with combinations of the features from the other attributes. Add no shock and it is impossible to estimate price sensitivity because the total price attribute would be perfectly predictable by a linear combination of the other attributes in the study. This is the multicollinearity problem and leads to difficulties estimating precise utilities for price as well as the other attributes and levels of interest in our study. We've done sensitivity analysis on how much random shock to do and have found that, depending on the study, +/-30% or +/-40% is usually enough to have better product price realism while avoiding too much price correlation in the summed price attribute relative to features shown (Orme 2007).

To calculate what price to display for a product (such as Samsung, 65-inch, 4K) we first sum the prices involved ($100 + $350 + $0) = $450 and then multiply that sum by a randomly drawn price shock (e.g., from the range of possibilities from -30% to +30%), such as +20%. $450 x 1.2 = $540. We can round the calculated price to the nearest $25 or $10 before showing the concept to respondents, if we think that adds more market realism.

Referring to our Summed Pricing table (Exhibit 3), the lowest possible price that could be shown is for an LG, 55-inch, 4K product shocked by -30% (assuming +/-30% is our selected random shock range). That price is ($0 + $250 + $0) x 0.7 = $175.

The highest possible price that could be shown is for a Samsung, 75-inch, 8K product shocked by +30%. That price is ($100 + $500 + $300) x 1.3 = $1170.

As examples of how summed pricing avoids showing unlikely product prices, an LG 55-inch 4K product would never be shown at a price higher than ($0 + $250 + $0) x 1.3 = $325. At the other extreme, a Samsung 75-inch 8K product would never be shown at a price lower than ($100 + $500 + $300) x 0.7 = $630.

We need to assess if the range of possible prices shown ($175 to $1170) is sufficient to represent the current market prices (or prices we want to investigate in market simulations that vary from the base case). We should avoid extrapolating beyond the price range included in the experiment during the analysis phase when conducting market simulations to inform pricing strategy. If the prices shown to respondents aren't wide enough, we can adjust the component prices in the summed pricing table and/or potentially adjust the random shock variation.

Because respondents see potentially dozens or hundreds of unique prices, we typically fit a continuous price function such as linear, log-linear, or piecewise regression. Coding the design matrix to estimate those effects is described in chapter 9 of "Becoming an Expert in Conjoint Analysis" (Orme and Chrzan, 2017). But, there is an easier trick for estimating utility functions for summed pricing we'll show below.

History of Summed Price CBC

I've conducted Summed Price various times over my career, going back to 1994 when I utilized Sawtooth Software's Ci3 system to design/field randomized Summed Pricing experiments for CBC and analyzed the data using a 3rd party (aggregate) multinomial logistic regression program called SYSTAT, followed by building a market simulator in Lotus 1-2-3 (precursor to Excel). My boss in 1994, Jon Pinnell, taught me how to do this.

I don't know of the author and article that first devised and described the summed pricing approach for conjoint analysis. Long-time CBC researchers tell me it goes back as early as the 1980s.

The Summed Price Discrete Price Levels Bucketing Approach

As mentioned earlier, Sawtooth Software's current CBC software does not natively support the summed pricing approach. The simplest approach (to me) for implementing summed pricing for CBC is an approach where you ask our Lighthouse Studio CBC software to design the attributes and levels to show in the choice tasks, including initially a 7-level placeholder for the summed price "shock" multiplier as (for +/-30% shock) 0.7x, 0.8x, 0.9x, 1.0x, 1.1x, 1.2x, 1.3x; or for +/-40% shock 0.6x, 0.7x, 0.85x, 1.0x, 1.15x, 1.3x, 1.4x. You export the design from our CBC software to a CSV file (30 versions, also known as blocks, should be plenty). Then, in Excel you modify that CSV file to replace the 7-level price multiplier attribute levels with the Summed Price attribute (by adding the assumed average price levels of the features using Excel formulas making up each concept to display and multiplying the sum by the drawn random shock multiplier). You've now got more realistic prices (than generic price CBC) to show for each product concept in each choice task that are modestly correlated with the features going into each product concept—better products tend to show at higher prices.

It makes utility estimation both easier within Lighthouse Studio and more flexible (to account for nonlinearity in the price function) if we discretize the prices rather than treating them as truly continuous. Using if-then Excel formulas, you can bucket (round to the nearest discrete price) the continuous shocked Summed Prices into one of, say, 10 discrete price levels (leading to 10 unique prices of a standard CBC price variable). For example, your new Summed Price attribute might have the following levels: $200, $275, $350, $425, $500, $600, $700, $825, $950, $1150.

Researchers at SKIM (including Michael Smith at our 2022 conference) have recommended 15+ cut-points in piecewise price functions for summed pricing designs, especially if utility constraints are imposed (to avoid overfitting). Yet, I worry about how few observations are supporting the extreme low and high price betas when we use so many cut points.

The next step is to import the new design from CSV file into CBC (after modifying the price variable in your CBC questionnaire now to represent the 10 absolute levels of price, rather than the original 7 relative random shock levels of price). And, ***importantly, you run Test Design and look at the standard errors for the 10 price levels!*** We usually target standard errors from test design data to be about 0.05 or less across the sample, for the planned sample size, using random respondent data and pooled logit analysis (that's what our Test Design does in CBC). But, we have a lot of price levels (10) and price has modest correlation with other attribute levels, so the precision won't be quite as tight as we usually see with standard non-correlated CBC pricing designs that have 4 or 5 total price levels. This is expected. We're trading off some precision for the benefit of showing realistic prices. Because we eventually may be constraining the prices, I'd be willing to live with standard errors as high as about 0.08 or so on the highest and lowest price points. If you're struggling to get reasonably low standard errors when testing the design in this way, you can go back to the first step and either use fewer total price points or shifting your bucket definitions so that the frequencies are better balanced across the discrete price points. (We'll describe this further below.)

Field the study as normally you would with the (say) 10 discrete levels of summed price. The beauty of this power trick is that the analysis feels exactly as if we were using a 10-level generic price attribute. I would typically estimate the model with utility constraints with either HB MNL analysis or Latent Class MNL analysis built into Lighthouse Studio. However, don't take as given that price should be constrained negative across all 10 discrete price points (this is true of any CBC study with price). I recommend first running the model unconstrained in Latent Class (say, a 4-group solution), looking for groups of respondents who may not think price should be constrained monotonically negative from lowest summed price utility to highest. I’ve seen situations with, for example, home prices studied in CBC where the lowest two price levels have worse utility (main effects) than the third price point. In that case, price is a signal for quality and you should take care not to just apply a utility constraint across the full continuum of price levels for all respondents.

Testing the Discretized Summed Price Design

After I’ve imported my discretized (bucketed) summed price design into my Lighthouse Studio CBC project, I check the distribution of occurrences of prices in the design (across all versions and within each version) to make sure it seems pretty reasonable. This is part of the Test Design procedure I emphasized above. Recall that I initially suggested for the HDTV study to round the summed prices (+shock) to these nearest 10 discrete prices:

Exhibit 4

Level#	Price
1	$200
2	$275
3	$350
4	$425
5	$500
6	$600
7	$700
8	$825
9	$950
10	$1150

The distribution of number of times each price level appears across all 30 versions of the design, the average frequency per version (block), and the standard errors (aggregate logit, given n=500 random responders, 15% Dual-Response None choices) are:

Test Design Results
Exhibit 5

Level#	Price	Frequency across 30 versions	Average Frequency per Version	Std Errors (n=500, 15% DR-None)
1	$200	30	1.0	0.099
2	$275	92	3.1	0.064
3	$350	128	4.3	0.052
4	$425	158	5.3	0.043
5	$500	117	3.9	0.047
6	$600	176	5.9	0.040
7	$700	152	5.1	0.043
8	$825	117	3.9	0.052
9	$950	77	2.6	0.065
10	$1150	33	1.1	0.095

We see more observations in the middle part of the price distribution and fewer at the two extremes of the price continuum. Products in the real world are probably more concentrated in the middle of the price range as well. Even so, we may not be terribly pleased with how thin the design seems to be for the lowest and highest prices. The standard errors for prices 1 and 10 are a bit high to my liking (0.099 and 0.095) in an absolute sense and also relative to the other attribute levels. Although not shown in the table above, the biggest standard error for any level of a non-price attribute was 0.026, so we're doing fine for those utility effects.

How can we get a more even distribution of prices such that the lowest and highest prices show more often? We could increase the random shock variation from -40% to +40% (but our products will not be quite as realistic as they were before). I've repeated the steps outlined above, but this time with -40% to +40% range of shock variation. The lowest frequency of the 10 prices is now 57 instead of 30. The worst standard error is now 0.071 instead of 0.099. This trades off some realism for better precision.

There are two simple approaches to increase the precision for the extreme price points in bucketed summed price designs in CBC:

- Use overall fewer discrete prices, such as 7 or 8 total prices instead of 10 (less granularity, but more precision).
- Adjust the widths of your prices making up the buckets, so that there are bigger price gaps between the lowest and highest levels (allowing these extreme price levels to capture a larger number of observations when rounding into their discrete price buckets). For example, we could create the following 10 discrete price levels into which we round and assign each summed product price: $200, $315, $400, $450, $500, $575, $650, $725, $850, $1150.

After adjusting the price levels in this way for rounding and bucketing the summed prices for products in the CBC survey, we obtain the following new results from test design (Exhibit 6):

Test Design Results
Exhibit 6

Level#	Price	Frequency across 30 versions	Average Frequency per Version	Std Errors (n=500, 15% DR-None)
1	$200	91	3.0	0.070
2	$315	108	3.6	0.060
3	$400	108	3.6	0.056
4	$450	113	3.8	0.050
5	$500	125	4.2	0.048
6	$575	113	3.8	0.050
7	$650	112	3.7	0.052
8	$725	120	4.0	0.051
9	$850	108	3.6	0.059
10	$1150	82	2.7	0.071

If you don't like the bigger gaps between price points at the low and high end of the price scale as shown in Exhibit 6, there are more advanced strategies you could devise to increase the frequency of lower-priced and higher-priced products in your study. Your strategy will depend on the specifics of your study and the tradeoffs you're willing to make.

The key point with these summed pricing designs is to ***always check the Test Design report to ensure you can get reasonable estimates of all the attribute levels of interest in your study***. If the standard errors are not turning out well, I've suggested some strategies for improving them.

Using Conditional Pricing Table in CBC

Another way to trick Lighthouse Studio's CBC into doing summed pricing is to use CBC's conditional pricing table to sum the prices and to specify the +/-30% (or +/-40%) price shocks. This is pretty easy setup for designing, displaying the right summed prices to respondents, and fielding the CBC questionnaire. But, then you have to deal with rounding (to the appropriate nearest 1, 5, 10, 25, etc. units of price) in the price look-up table entries for the summed prices to show respondents, exporting the data afterward and treating price as continuous in standalone CBC/HB or standalone Latent class MNL software. Another potential problem is the look-up table could become excessively massive if you have very many triggering attributes and levels that carry component-based prices. You can fit a linear term to continuous price, or log-linear (treating the price variable as "user-specified coding" in the standalone modules). But, you can get fancier to apply your own piecewise coding as "user-specified coding" within the standalone utility estimation modules. (See Chapter 9 in Orme and Chrzan 2017 for details.) Anyway, this approach is not as straightforward, as you don't have the native market simulator that results when you do the rounded and discretized way into a reasonable number of price levels I recommended above that keeps everything native within Lighthouse Studio.

References

Orme, Bryan (2007), “Three Ways to Treat Overall Price in Conjoint Analysis,” Sawtooth Software Research Paper, accessible at: https://sawtoothsoftware.com/resources/technical-papers/three-ways-to-treat-overall-price-in-conjoint-analysis

Orme, Bryan and Keith Chrzan (2017), “Becoming an Expert in Conjoint Analysis”, third edition, Provo, UT.

Smith, Michael (2022), “Are We Overfitting Our Models with Too Many Price Parameters?”, 2022 Sawtooth Software Conference Proceedings, pp 229-238, Provo, UT.