




validateHOT - an R package for the analysis of holdout/validation tasks and other choice modeling tools

Joshua Benjamin Schramm¹ and Marcel Lichters¹

DOI:

¹ Otto-von-Guericke-University Magdeburg

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Submitted:

Published:

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

validateHOT is an R package that provides functions for preference measurement techniques such as (adaptive) choice-based conjoint analyses (hereafter CBC and ACBC, respectively) and maximum difference scaling (hereafter MaxDiff). Specifically, the package allows users to analyze validation tasks, perform market simulations, and rescale raw utility scores. It is compatible with data obtained using, for example, the ChoiceModelR package (Sermas, 2022) or Sawtooth Software's Lighthouse Studio (Sawtooth Software Inc., 2024).¹

Statement of need

Researchers and practitioners use preference measurement techniques for various research purposes, such as simulating markets or determining the importance of attributes when making choices (Steiner & Meißner, 2018). The ultimate goal is to predict future behavior (Green & Srinivasan, 1990). To make accurate predictions and informed decisions, it is crucial to ensure that the collected data is valid. A prominent way to test a model's validity is the analysis of validation tasks (e.g., Orme, 2015), which are typically fixed tasks (i.e., same across participants) and excluded for utility estimation.

The validateHOT package provides the necessary tools for the aforementioned applications: (1) assessing the model's validity, (2) performing market simulations, (3) converting raw utility scores. Finally, as an open-source tool, it helps academics to report analysis scripts in scientific journal articles, fostering reproducibility.

State of the field

Other R packages offer functions to calculate validation metrics. However, these are not always tailored to the individual raw utilities extracted from preference measurement techniques. The Metrics package (Hamner & Frasco, 2018), for example, provides functions to run validation metrics such as mean absolute error (MAE) or metrics of the confusion matrix. However, converting outputs such as those from estimations using Sawtooth Software's Lighthouse Studio (Sawtooth Software Inc., 2024) or the ChoiceModelR package (Sermas, 2022) into the proper format requires some data wrangling. The conjoint package (Bak & Bartlomowicz, 2012) offers functions that are similar to those in validateHOT but it lacks validation functions and focuses primarily on classical conjoint analysis. Thus, it is limited when applied to more common conjoint methods. The logitr package (Helveston, 2023) offers market simulation tools but does not include validation metrics such as mean hit probability or hit rate. [Figure 1](#) compares validateHOT's functions with those of other R packages. To the best of our knowledge, no package converts raw utility

¹We refer to both validation and holdout tasks interchangeably.

Key functionalities	validateHOT (v 1.0.5)	Metrics (v 0.1.4)	caret (v 7.0-1)	conjoint (v 1.41)	phileentropy (v 0.9.0)	logitr (v 1.1.2)	mlogit (v 1.1-1)	support.bws (v 0.4-6)
Confusion matrix	✓	✓	✓					
Creating design matrix				✓		✓	✓	✓
Creating holdout / market scenario	✓			✓		✓		
Estimate utilities				✓		✓	✓	✓
Estimate WTP						✓		
Hit rate	✓							
Kullback-Leibler-Divergence	✓				✓			
MAE, MedAe, RMSE	✓	✓	✓					
Market Shares	✓		✓	✓		✓		✓
Mean hit probability	✓							
TURF, TURF ladder	✓							

Figure 1: Comparison of the functions of validateHOT with those of existing R packages

scores into validation metrics or running a variety of marketing simulations (especially Total Unduplicated Reach and Frequency (TURF) and TURF ladder).

We introduce validateHOT drawing on data estimated using Sawtooth Software Lighthouse Studio (Sawtooth Software Inc., 2024). However, it can also be used on results stemming from packages such as ChoiceModelR (Sermas, 2022), bayesm (Rossi, 2023), or STAN (2023), if used with similar settings.

Key functions

validateHOT's functions can be categorized into four main components (see Table 1). To convert the data in the correct format for most functions, we created the `create_hot()` function, which calculates each alternative's total utility in conjoint studies by applying the additive utility model.

Table 1: Overview of validateHOT's functions

Validation metrics	Confusion matrix	Market simulations	Rescaling scores
<code>hitrate()</code>	<code>accuracy()</code>	<code>freqassort()</code>	<code>att_imp()</code>
<code>kl()</code>	<code>f1()</code>	<code>marksim()</code>	<code>prob_scores()</code>
<code>mae()</code>	<code>precision()</code>	<code>reach()</code>	<code>zc_diffs()</code>
<code>medae()</code>	<code>recall()</code>	<code>turf()</code>	<code>zero_anchored()</code>
<code>mhp()</code>	<code>specificity()</code>	<code>turf_ladder()</code>	
<code>rmse()</code>			

Typical workflow

We present the workflow for a MaxDiff (Schramm & Lichters, 2024) and a CBC study with a linear-coded price attribute (the vignette provides further examples; Sablotny-

Wackershauser, Lichters, Guhl, Bengart, & Vogt (2024)).

MaxDiff

Creating validation task/market scenario

Following hierarchical Bayes estimation, the raw utilities must be imported into an R `data.frame` object. The first example assumes a validation task with seven alternatives plus a no-buy alternative. We define the data set (`data`), and the column names of the unique identifier (`id`) and the no-buy alternative (`none`). Next, we specify the attribute levels of each alternative (`prod.levels`), the `method`, the variables that should be kept in the new data frame (`varskeep`), and finally, the actual `choice` in the validation task.

```
hot_mxd <- create_hot(  
  data = maxdiff,  
  id = "id",  
  none = "none",  
  prod.levels = list(2, 9, 10, 14, 15, 16, 17),  
  method = "maxdiff",  
  varskeep = "group",  
  choice = "hot"  
)
```

Validation metrics

To calculate the hit rate using the `hitrate()` function, we define the arguments: `data`, `opts` (the alternatives in the validation task), and `choice`.

```
hitrate(  
  data = hot_mxd,  
  opts = c(option_1:none),  
  choice = choice  
)
```

Market simulation

`turf()` is a tool helping to find the optimal assortment generating the highest reach. This method is particularly useful for MaxDiff studies (Chrzan & Orme, 2019, p. 108). Users can specify the arguments `fixed` (i.e., alternatives that are mandatory in an assortment) and `prohib` (i.e., forbid specific combinations).

Below we assume the user conducted an anchored MaxDiff analysis with 10 items (`opts`) and wants to find the best assortment with a size of 3 items (`size`). The anchor (no-buy alternative) is the threshold that must be exceeded (`none`). Finally, `option_01` (`fixed`) is fixed meaning that it must be part of each combination.

```
turf(  
  data = maxdiff,  
  opts = c(option_01:option_10),  
  none = none,  
  size = 3L,  
  fixed = "option_01"  
  approach = "thres"  
) %>%  
  head(n = 5)
```

CBC

Creating validation task/market scenario

The CBC setup of `create_hot()` is nearly the same as for the MaxDiff example. Besides the arguments defined above, we must define the linear-coded variable (`lin.p`), the coding of the attributes (`coding`), and the values that were used when estimating the utilities for the linear-coded variable (`interpolate.levels`).

```
hot_cbc_linear <- create_hot(  
  data = cbc_linear,  
  id = "id",  
  none = "none",  
  prod.levels = list(  
    c(3, 6, 10, 13, 16, 20, 24, 32, 248.55),  
    c(3, 5, 10, 14, 16, 18, 22, 27, 237.39),  
    c(4, 6, 9, 14, 15, 20, 25, 30, 273.15),  
    c(4, 5, 10, 11, 16, 19, 26, 32, 213.55),  
    c(2, 6, 8, 14, 16, 17, 26, 31, 266.10),  
    c(2, 5, 7, 12, 16, 20, 26, 29, 184.50)  
  ),  
  coding = c(rep(0, times = 8), 1),  
  lin.p = "price",  
  interpolate.levels = list(c(seq(from = 175.99, to = 350.99, by = 35))),  
  method = "cbc",  
  choice = "hot"  
)
```

Rescaling scores

The function `att_imp()` calculates the attributes' relative importance. `attrib` defines the attribute levels. The other arguments are the same as above.

```
att_imp(  
  data = cbc_linear,  
  attrib = list(  
    paste0("att1_lev", c(1:3)),  
    paste0("att2_lev", c(1:2)),  
    paste0("att3_lev", c(1:4)),  
    paste0("att4_lev", c(1:4)),  
    paste0("att5_lev", c(1:2)),  
    paste0("att6_lev", c(1:4)),  
    paste0("att7_lev", c(1:6)),  
    paste0("att8_lev", c(1:6)),  
    "price"  
  ),  
  coding = c(rep(0, times = 8), 1),  
  interpolate.levels = list(c(seq(from = 175.99, to = 350.99, by = 35))),  
  res = "agg"  
)
```

Availability

The package `validateHOT` is available on [GitHub](https://github.com/validateHOT/validateHOT).

Acknowledgments

We would like to thank Sawtooth Software (Sawtooth Software Inc., 2024) for their transparent documentation. The first author receives an [academic grant](#) from Sawtooth Software, Inc. (USA) during his Ph.D. time.

References

- Bak, A., & Bartlomowicz, T. (2012). Conjoint analysis method and its implementation in conjoint r package, 239–248. Retrieved from http://keii.ue.wroc.pl/pracownicy/tb/Bak_A_and_Bartlomowicz_T_Conjoint_analysis_method_and_its_implementation_in_conjoint_R_package.pdf
- Chrzan, K., & Orme, B. K. (2019). *Applied MaxDiff: A practitioner's guide to best-worst scaling*. Provo, UT: Sawtooth Software.
- Green, P. E., & Srinivasan, V. (1990). Conjoint Analysis in Marketing: New Developments with Implications for Research and Practice. *Journal of Marketing*, 54(4), 3–19. doi:[10.1177/002224299005400402](https://doi.org/10.1177/002224299005400402)
- Hamner, B., & Frasco, M. (2018). Metrics: Evaluation metrics for machine learning. Retrieved from <https://CRAN.R-project.org/package=Metrics>
- Helveston, J. P. (2023). {Logitr}: Fast estimation of multinomial and mixed logit models with preference space and willingness-to-pay space utility parameterizations, 105. doi:[10.18637/jss.v105.i10](https://doi.org/10.18637/jss.v105.i10)
- Orme, B. K. (2015). Including holdout choice tasks in conjoint studies.
- Rossi, P. (2023). Bayesm: Bayesian inference for marketing/micro-econometrics. Retrieved from <https://CRAN.R-project.org/package=bayesm>
- Sablotny-Wackershauser, V., Lichters, M., Guhl, D., Bengart, P., & Vogt, B. (2024). Crossing incentive alignment and adaptive designs in choice-based conjoint: A fruitful endeavor. *Journal of the Academy of Marketing Science*. doi:[10.1007/s11747-023-00997-5](https://doi.org/10.1007/s11747-023-00997-5)
- Sawtooth Software Inc. (2024). Lighthouse studio 9. Sequim, WA.: Sawtooth Software Inc. Retrieved from <https://www.sawtoothsoftware.com/products/online-surveys>
- Schramm, J. B., & Lichters, M. (2024). Incentive alignment in anchored MaxDiff yields superior predictive validity. *Marketing Letters*. doi:[10.1007/s11002-023-09714-2](https://doi.org/10.1007/s11002-023-09714-2)
- Sermas, R. (2022). ChoiceModelR: Choice modeling in r. Retrieved from <https://CRAN.R-project.org/package=ChoiceModelR>
- Stan Development Team. (2023). {RStan}: The {r} interface to {stan}. Retrieved from <https://mc-stan.org/>
- Steiner, M., & Meißner, M. (2018). A user's guide to the galaxy of conjoint analysis and compositional preference measurement. *Marketing ZFP*, 40(2), 3–25. doi:[10.15358/0344-1369-2018-2-3](https://doi.org/10.15358/0344-1369-2018-2-3)