ACCSCIENCE PUBLISHING

Journal of Clinical and Translational Research

SHORT COMMUNICATION

Enhancing binary dose–response analysis in clinical and translational research: Leveraging grouped data techniques in R

Jenny-Hoa Q. Nguyen*, Fudan Zheng, Yuan Xiong, and Mahesh N. Samtani

Department of Clinical Pharmacology and Pharmacometrics, Johnson and Johnson, Spring House, Pennsylvania, United States of America

Abstract

Background: In pharmacometric analyses, binary dose–response outcome data are used to understand drug potency through the pharmacologic parameter, Effective dose 50 (ED50). Optimal treatment strategies can be developed by characterizing a drug's dose–response curve, which provides insights into the theoretical maximum effect and the steepness of the curve in response to changes in dose or exposure. Approaches for analyzing group-level response data have not been systematically described in pharmacometric literature, although they are commonly applied in the statistical literature. **Aim:** This study demonstrates the use of R to analyze grouped or ungrouped binary data, with a focus on pharmacometric applications. **Methods:** Simulated data were generated to represent a hypothetical Phase 2, dose-ranging, placebo-controlled, randomized clinical trial of drug X, consisting of 250 participants randomized into five distinct cohorts, including a single placebo arm. Linear and non-linear Emax models were fit to the simulated data. **Results:** Both grouped and ungrouped data approaches produced identical final parameter estimates in logistic regression using the linear and Emax models. The same ED50 value for drug X was obtained from both approaches in the Emax model. **Conclusion:** This study demonstrates the various methods by which summary- or subject-level binary data can be analyzed using R to model binary response data. **Relevance for patients:** This work helps bridge the gap between statistical and pharmacometric analysis techniques in the context of binary data analysis. This type of data may facilitate comparative assessments of drug potency and maximal effect using publicly available information from scientific publications or regulatory approval documents.

Keywords: Generalized linear models; Generalized non-linear models; Logistic regression; R software; Binary data; Dose–response analysis

***Corresponding author:**
Jenny-Hoa Q. Nguyen
(jnguye58@its.jnj.com)

Citation: Nguyen JQ, Zheng F, Xiong Y, Samtani MN. Enhancing binary dose–response analysis in clinical and translational research: Leveraging grouped data techniques in R. *J Clin Transl Res*. 2025;11(6):76-86.
doi: 10.36922/JCTR025300045

Received: July 23, 2025

Revised: August 25, 2025

Accepted: September 11, 2025

Published online: October 22, 2025

1. Introduction

In Phase 2 dose-finding studies, dose–response information is often expressed as a binary outcome versus dose, where the achievement of an endpoint of interest—such as a PASI90 response (representing a 90% improvement from baseline in the Psoriasis Area and Severity Index) in psoriasis clinical trials—is reported at each dose level evaluated

for a given drug. Generalized linear models (GLMs), such as logistic regression, can be used to effectively analyze this type of data, regardless of whether the data are programmed at a grouped or individual (i.e., ungrouped) level.[1,2] In addition, GLMs for binary data can be further extended to generalized non-linear models (GNMs), where non-linear relationships between the logit transformation of the response and the predictors of interest can be introduced into the model, including the standard Emax relationship frequently used in pharmacometric analyses.[3] In the context of logistic regression, the logit link function transforms the predicted probabilities using log-odds, allowing binary outcome data to be modeled while ensuring that predicted probabilities remain within the interval (0, 1). While alternative link functions—such as the probit link function—can also be specified and applied, the logit link function is used here due to its ease of interpretation through odds ratios.

This study demonstrates the use of the "stats::glm()" and "gnm::gnm()" functions in R for analyzing summary-level binary response data with a grouped data approach.[4,5] The simulated data used in this study represent a landmark analysis where only one observation is collected per patient. As a result, complex models accounting for hierarchical or correlated data dependencies were unnecessary, as the simulated data did not require inclusion of random effects in the model structure; therefore, the use of packages such as "glmmTMB" and "nlme" was not required.[6,7] Furthermore, results from the grouped and ungrouped approaches will be provided to illustrate that the results from either approach are identical, thereby giving pharmacometricians the flexibility to work with a variety of data structures to achieve the same goal. The grouped data approach may be a valuable tool for model-based meta-analysis when analyzing summary-level binary data.

2. Materials and methods

2.1. Study data

In this study, simulated data representing a hypothetical Phase 2, placebo-controlled, dose-ranging randomized clinical trial for drug X were generated using R version 4.2 (R Foundation for Statistical Computing, Austria). A total of 250 participants were assigned in the simulation to one of five possible dose levels (0 mg [placebo], 1 mg, 5 mg, 10 mg, and 20 mg), with 50 participants per cohort. The probability of response at each dose level was assigned based on real-life clinical data. To generate the individual response data, the probability of response at each dose level was used to assign a 0 or 1 value representing failure or success, respectively, from a binomial distribution.

2.2. Efficacy data

Efficacy results describing participants' responses to drug X as a binary variable (i.e., responder or non-responder) were simulated at the landmark time point, defined here as the primary endpoint evaluated at week 12. To capture elements of a real clinical trial and perform a landmark analysis at the pivotal endpoint, the efficacy data were assigned based on real-life clinical data. The guidelines used to generate the hypothetical clinical trials are as follows:

(i) The placebo response rate was allowed to be >0%
(ii) The maximum response rate was not allowed to exceed 90%
(iii) In general, higher doses of drug X were expected to lead to higher response rates; however, the response rates between placebo and 1 mg may be similar, and the response rates between 10 mg and 20 mg may be similar.

The 90% upper limit for maximum response reflects clinical reality, as responses in many diseases rarely reach 100% due to their complex and multifactorial nature. The similarity observed between the 10 mg and 20 mg doses aligns with the Emax relationship demonstrated in recent studies, which show a plateau in efficacy where responses flatten at higher doses and do not approach complete remission.[8] This pattern is consistent with pharmacodynamic behavior across various diseases, suggesting that increasing the dose beyond a certain point provides minimal additional benefit.

2.3. Fitting GLM with "stats::glm()"

Data were formatted as illustrated in Figure 1A for the grouped data approach using the proportion of successes, or in Figure 1B for the grouped data approach using the number of successes and failures (i.e., non-responders). For the ungrouped data approach, which is the more standard method, data were provided as individual-level responses, as illustrated in Figure 1C. Appendix 1 provides the complete R code used for model implementation, including the embedded datasets defined within the script. It illustrates both the ungrouped "long format" and grouped ("summary-level") data structures, demonstrating how the ungrouped data can be more cumbersome to prepare, whereas the grouped data are more compact and easier to format.

To fit a logistic regression model with dose as the predictor, the following arguments in the "stats::glm()" function were used (Expression I):

```
glm.mod1 <- glm(                                    (I)
formula = <response> ~ <predictor>,
```

A

dose_group	dose_val	time	endpoint	n_subjs	prop_resp
0 mg	0	12	Endpoint A	50	0.08
1 mg	1	12	Endpoint A	50	0.24
5 mg	5	12	Endpoint A	50	0.54
10 mg	10	12	Endpoint A	50	0.84
20 mg	20	12	Endpoint A	50	0.84

B

dose_group	dose_val	time	endpoint	n_success	n_fail
0 mg	0	12	Endpoint A	4	46
1 mg	1	12	Endpoint A	12	38
5 mg	5	12	Endpoint A	27	23
10 mg	10	12	Endpoint A	42	8
20 mg	20	12	Endpoint A	42	8

C

ID	dose_group	dose_val	time	endpoint	response
1	0 mg	0	12	Endpoint A	1
2	0 mg	0	12	Endpoint A	1
3	0 mg	0	12	Endpoint A	1
4	0 mg	0	12	Endpoint A	1
5	0 mg	0	12	Endpoint A	0

Figure 1. Input data formats for fitting models using the logit transformation of the response. (A) Group-level data showing the proportion of successes for each group. (B) Group-level data showing total counts of successes and failures for each dose group. (C) Individual-level data indicating response outcomes as failure (0) or success (1).

```
weights = <total number of subjects per group>,
family = binomial(link = "logit"),
data = <dataframe>
)
```

When fitting the proportion of successes as the response (i.e., data formatted as in Figure 1A), a warning message from R may arise, alerting the user that the specification of "family = binomial(link = "logit")" contains non-integer (i.e., proportion) values. This occurs because the binomial distribution is the sum of *n* Bernoulli trials, where each trial may only take the value 0 or 1 to represent failure or success, and therefore, non-integer values are not expected.[9] Here, the *n* Bernoulli trials refer to the *n* total participants per dosing group; therefore, the "weights" argument provides the complementary information needed to fit the model. This message can be safely ignored, as it does not impact the results returned from R when the "weights" argument is supplied. The model can thus be fit as follows (Expression II):

```
glm.mod1.fig1a <- glm(                                    (II)
formula = prop_resp ~ dose_val,
weights = n_subjs,
family = binomial(link = "logit"),
data = df_input_summary
)
```

If proportions are not used, the "weights" argument does not need to be supplied, since the total number of successes and failures can be used instead. This results in the following function (Expression III) for data formatted as in Figure 1B:

```
glm.mod1.fig1b <- glm(                                    (III)
formula = cbind(n_success, n_fail) ~ dose_val,
family = binomial(link = "logit"),
data = df_input_summary
)
```

To demonstrate that the grouped and ungrouped responses yield the same model results, the function used

to model individual-level data (Figure 1C) is as follows (Expression IV):

```
glm.mod1.fig1d <- glm                                    (IV)
formula = response ~ dose
family = binomial(link = "logit"),
data = df_input_individual
)
```

2.4. Fitting GNMs with "gnm::gnm()"

Data were presented in one of the forms described in the previous section and illustrated in Figure 1. Arguments such as "formula," "weights," "family," and "data" were provided for "gnm::gnm()" in the same manner as in "stats::glm()," depending on the format of the data. Custom model functions were defined and subsequently supplied to the "formula" argument of the function. Figure 2 illustrates how a custom function was defined to fit the standard Emax function, where the Hill shape parameter was assumed to equal 1. It should be noted that the Emax model may not have been appropriate in cases where the data did not support an Emax-like relationship, such as inverted U-shaped responses or scenarios lacking a clear plateau.

In the function definition for "fx_emax()" provided in Figure 2, "predictors" refers to the list of named parameters (referred to as "predLabels" in the "term" argument) that will be estimated by the model and are provided as a named list with an initial value of 1, while "variables" refers to the independent variable used for the model fit, such as dose (referred to as "varLabels" in the "term" argument). The "term" argument requires the definition of the custom function to be used, and the "sprinf()" function facilitates the definition of custom functions by utilizing the indices provided in the "predictors" and "variables." Additional guidance can be found in the R help documentation for this function (accessible via typing `?sprintf` in the R console, or via https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/sprintf). Finally, the "class" of the function must be defined as "non-lin" before fitting the "fx_emax()" model using "gnm::gnm()."

To fit the data in the formats presented in Figure 1, the guidelines provided in the previous section for logistic regression models were followed, ensuring that the predictors are provided in the formula call with the defined "fx_emax()" function (Figure 2). For example, the function call for the data provided in Figure 1A is as follows (Expression V):

```
gnm.mod2.fig1a <- gnm(                                   (V)
formula = prop_resp ~ fx_emax(dose_val),
weights = n_subjs,
family = binomial(link = "logit"),
data = df_input_summary
)
```

3. Results

3.1. Comparison of grouped and ungrouped approaches

Tables 1 and 2 present the results of the linear (GLM) and non-linear (GNM) model fits, respectively. In addition, Figure 3 illustrates the results of the linear and Emax models. The grouped and ungrouped approaches provided identical model estimates for each model type fitted to the data. Researchers are advised to consult outside texts for information regarding goodness-of-fit assessments for logistic regression, such as deviance residuals and the HosmerLemeshow test.[10]

A Standard Emax function

$$\text{logit}(\text{p} = \text{probability of response}) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \frac{E_{max} \cdot \text{Dose}}{ED_{50} + \text{Dose}}$$

B R code

```
fx_emax <- function(x){
  list(
    predictors = list(Emax = 1, ED50 = 1),
    variables  = list(substitute(x)),
    term       = function(predLabels, varLabels) {
      sprintf(
        "%s * %s / (%s + %s)",
        predLabels[1], varLabels[1],
        predLabels[2], varLabels[1]
      )
    }
  )
}

class(fx_emax) <- "nonlin"
```

Figure 2. Defining custom functions in R for use with the "gnm::gnm()" function based on the standard Emax function. (A) Standard Emax model equation. (B) R code.

Table 1. Final model parameter estimates from the linear model (logistic regression) fit using a generalized linear model

Parameter	Grouped data with the proportion of success		Grouped data with failure and success counts		Ungrouped data	
	Estimate	Standard error	Estimate	Standard error	Estimate	Standard error
Intercept	−1.270	0.2130	−1.270	0.2130	−1.270	0.2130
Slope	0.204	0.0284	0.204	0.0284	0.204	0.0284
AIC[a]	45.857		45.857		266.358	

Notes: [a]AICs are not directly comparable between the grouped and ungrouped approaches, as the definitions of the likelihood functions optimized in the different model fits differ. As a metric, AIC combines the number of parameters with a likelihood measure, such as negative two times the log-likelihood. Figure 3A for the model-predicted results using this generalized linear model.
Abbreviation: AIC: Akaike information criterion.

Table 2. Final model parameter estimates from the non-linear model (standard Emax) fit using a generalized non-linear model

Parameter	Grouped data with the proportion of success		Grouped data with failure and success counts		Ungrouped data	
	Estimate	Standard error	Estimate	Standard error	Estimate	Standard error
Intercept	−2.28	0.423	−2.28	0.423	−2.28	0.423
Emax	5.01	0.742	5.01	0.742	5.01	0.742
ED50	4.29	2.310	4.29	2.310	4.29	2.310
AIC[a]	27.944		27.944		248.715	

Notes: [a]AICs are not directly comparable between the grouped and ungrouped approaches, as the definitions of the likelihood functions optimized in the different model fits differ. As a metric, AIC combines the number of parameters with a likelihood measure, such as negative two times the log-likelihood. Figure 3B for the model-predicted results using this generalized non-linear model.
Abbreviations: AIC: Akaike information criterion; ED50: Effective Dose 50.

3.2. Comparison of Effective Dose 50 (ED50) estimates between the generalized linear and GNMs

In the GLM, the ED50 value can be estimated by algebraically solving for the dose level at which 50% of the response is achieved. Using the output from Table 1, the derived ED50 value for GLM was calculated to be 6.23 mg, whereas the estimated ED50 value for the GNM was 4.23 mg (Table 2). However, Figure 3 shows that the GNM provides a better fit to the simulated data, which was further confirmed by the lower Akaike information criterion value for the GNM relative to the GLM when both models were fit using the same data structure.

The difference in the ED50 values obtained from the two model approaches may be attributed to several factors that demonstrate the inadequacy of the linear model for the simulated data. First, the maximum response rate was not allowed to exceed 90% in the simulated data and therefore cannot approach 100%, as shown in Figure 3A. It should be noted that the underlying assumption in a logistic regression model is that there is a monotonic increase toward 100% (or a monotonic decrease toward 0%) and, therefore, it cannot plateau at an intermediate value. Second, a placebo response was observed in the hypothetical trial, which could not be adequately captured by the GLM approach; however, as shown in Figure 3B, this placebo response was well represented in the GNM approach.

When applied to clinical trial data, the GNM offers the capability to analyze situations in which the observed maximum response rate does not approach 100% and a placebo response is observed. While the intercept in the GLM can represent the placebo response, its ability to model such data are limited as the shape parameter is not estimable, which likely contributes to the overestimation of the placebo effect in the GLM fit (Figure 3A). In addition, it should be noted that the GLM offers the flexibility to estimate a polynomial fit (e.g., $y \sim x + x^2 + x^3 \ldots$), but such a model lacks pharmacological interpretability with respect to its parameters.

3.3. Applications within pharmacometrics and drug development

In this study, using simulated data for drug X, the ED50 value estimated from the GNM—whether using either the grouped or ungrouped approach—was 4.29 mg (Table 2). In the context of drug development, the ED50 value can be used to compare different drugs within the same class to determine which drug may be more potent and thus requires a lower dose to achieve the same therapeutic effect. If several candidates within the same drug class are

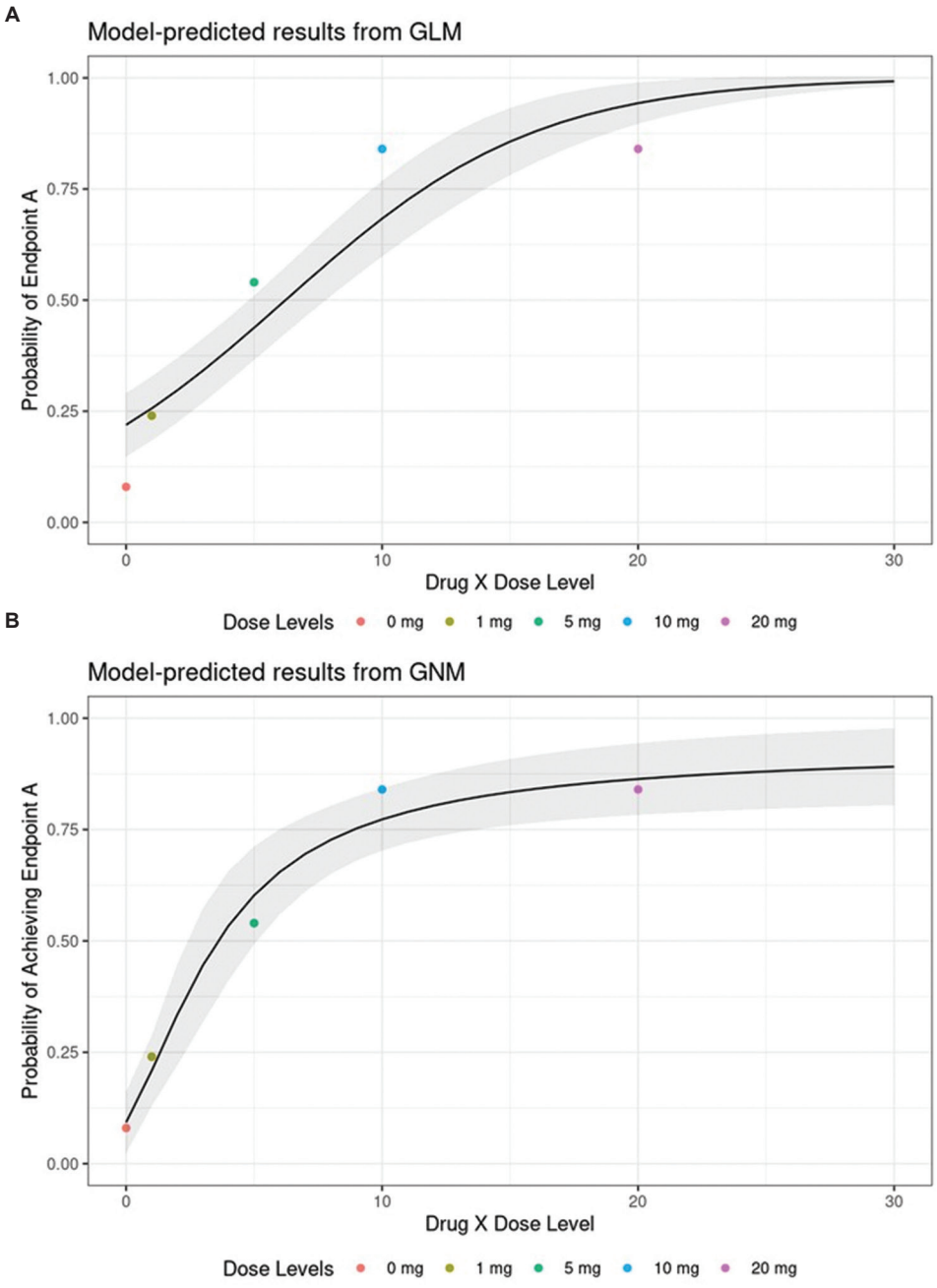

Figure 3. Model-predicted probability of achieving endpoint A at week 12 using linear and non-linear models. (A) Model-predicted results obtained using a standard logistic regression model implemented in "stats::glm()." (B) Model-predicted results obtained using a logit-transformed Emax model implemented in "gnm::gnm()."
Abbreviations: GNM: Generalized non-linear model; GLM: Generalized linear model.

being considered for further development, and assuming all candidates demonstrate similar safety profiles, the candidate with a lower ED50 value may be a more suitable option from a cost-of-goods or dosing perspective. This would allow a lower injection volume for a subcutaneously administered drug, assuming similar pharmacokinetic half-lives across drugs.

However, if several candidates from different classes (e.g., biologics versus small-molecule drugs) are being considered for development, and the Emax value can help determine which drug class is more efficacious based on the maximal effect of the drug, where a higher Emax value indicates greater efficacy. For example, biologics are generally more efficacious than small-molecule drugs

for the treatment of psoriasis based on efficacy data from registrational clinical trials.[11-18] The tutorial described in this study could be applied to estimate Emax values for these drugs, provided that appropriate data are available.

4. Discussion

The choice between the grouped and ungrouped approaches for analyzing binary dose–response data depends on the nature of the available dataset and the objectives of the analysis. Both methods yield identical parameter estimates when applied correctly, as demonstrated in this study, although they differ in practical considerations.

Grouped data methods are particularly advantageous when only summary-level information is available, such as in meta-analyses or when extracting data from published literature or regulatory documents. These approaches reduce data-handling complexity by condensing individual-level observations into aggregated counts or proportions, thereby simplifying model implementation.

On the other hand, ungrouped data methods should be prioritized when individual-level data are available, as they allow greater flexibility in modeling, including the incorporation of covariates, random effects, and hierarchical structures. These features are essential for addressing variability across subjects and for conducting more detailed analyses, such as mixed-effects modeling or time-to-event evaluations.

GLMs are straightforward to implement and interpret, making them suitable for scenarios where the dose–response relationship is approximately linear on the logit scale. GNMs, which incorporate Emax or similar functions, offer greater flexibility for modeling pharmacologically plausible relationships. While GNMs require more complex model specification and interpretation, they provide parameters (e.g., Emax, ED50) that are directly relevant to pharmacology and drug development decision-making. In practice, GLMs can be used for rapid assessments or when the primary interest lies in odds ratios and the data exhibit near-linear behavior on the logit scale. When modeling dose–response relationships that exhibit saturation effects, or when pharmacologic interpretability of parameters is critical, GNMs should be preferentially employed.

Furthermore, the ED50 and Emax parameters derived from the GNMs have direct clinical relevance. ED50 indicates the dose required to achieve half of the maximal therapeutic effect, guiding dose selection and optimization strategies. A lower ED50 may indicate a more potent drug, which can influence candidate prioritization and cost considerations. Similarly, Emax reflects the ceiling of efficacy achievable with a drug, which is critical for comparing therapeutic classes and establishing realistic expectations for treatment outcomes. Understanding these parameters within the context of grouped versus ungrouped approaches ensures that pharmacometric analyses remain clinically meaningful and aligned with patient-centric goals.

Ultimately, bridging statistical and pharmacometric methodologies enhances the robustness of dose–response evaluations and facilitates informed decision-making in drug development.

5. Conclusion

This simple and practical approach is presented not only to promote the use of grouped data for analyzing publicly available information on categorical endpoints but also to highlight the value of utilizing statistical methods within pharmacometrics and to further advocate for bridging these two disciplines.[19] This study illustrates different approaches by which grouped data can be analyzed using R—for example, as group-level outcomes represented either as proportions of successes or as a two-column matrix of grouped data showing the numbers of successes and failures. In addition, dose–response data can be replaced with quantile-based exposure-response data obtained from published studies, where the proportions of responders are reported per quartile or tertile of exposure data, or with information extracted from digitized clinical datasets. Such data may be valuable for deriving EC50 estimates for assets with publicly available information reported in scientific publications or on summary basis of approval documents from regulatory agencies.

Acknowledgments

None.

Funding

None.

Conflict of interest

Jenny-Hoa Q. Nguyen, Fudan Zheng, Yuan Xiong, and Mahesh N. Samtani are employees and stockholders of Johnson & Johnson (JNJ). This work was conducted as part of fulfilling responsibilities within the company.

Author contributions

Conceptualization: Mahesh N. Samtani
Data curation: Jenny-Hoa Q. Nguyen
Formal analysis: Jenny-Hoa Q. Nguyen
Methodology: Jenny-Hoa Q. Nguyen, Fudan Zheng
Visualization: Jenny-Hoa Q. Nguyen
Writing–original draft: Jenny-Hoa Q. Nguyen

Writing–review & editing: Fudan Zheng, Yuan Xiong, Mahesh N. Samtani

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data

Data and relevant R code used in this publication are provided in Appendix 1.

References

1. Venables WN, Ripley BD. Generalized linear models. In: *Modern Applied Statistics with S*. 4th ed. New York: Springer; 2002. p. 330-370.
2. Faraway JJ. Generalized linear models. In: *Extending the Linear Model with R*. 2nd ed. New York: Chapman and Hall; 2016. p. 151-174.
3. Lane PW. Generalized nonlinear models. In: Prat A. editors. *COMPSTAT*. Germany: Physica-Verlag HD; 1996.

   doi: 10.1007/978-3-642-46992-3_42
4. R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria; 2022. Available from: https://www.com/r-project.org [Last accessed on 2025 Aug 20].
5. Turner H, Firth D. *gnm: Generalized Nonlinear Models*. R Package Version 1.1-5. 2023. Available from: https://www.com/cran.r-project.org/package=gnm [Last accessed on 2025 Aug 20].
6. Brooks ME, Bolker B, Kristensen K, *et al*. *glmmTMB: Generalized Linear Mixed Models using Template Model Builder*. R Package Version 1.1.12; 2025. Available from: https://www.com/cran.r-project.org/package=glmmtmb [Last accessed on 2025 Aug 20].
7. Pinherio J, Bates D, DebRoy S, *et al*. *Nlme: Linear and Nonlinear Mixed Effects Models*. R Package Version 3.1-168; 2025. Available from: https://www.com/cran.r-project.org/package=nlme [Last accessed on 2025 Aug 20].
8. Checchio T, Ahadieh S, Gupta P, *et al*. Quantitative evaluations of time-course and treatment effects of systemic agents for psoriasis: A model-based meta-analysis. *Clin Pharmacol Ther*. 2017;102(6):1006-1016.

   doi: 10.1002/cpt.732
9. Casella G, Berger RL. Binomial distribution. In: *Statistical Inference*. 2nd ed. Pacific Grove: Duxbury Press; 2002. p. 89-91.
10. Hosmer DW Jr., Lemeshow S, Sturdivant RX. *Applied Logistic Regression*. 3rd ed. New York: Wiley; 2013.
11. Armstrong AW, Puig L, Joshi A, *et al*. Comparison of biologics and oral treatments for plaque psoriasis: A meta-analysis. *JAMA Dermatol*. 2020;156(3):258-269.

    doi: 10.1001/jamadermatol.2019.4029
12. Thaçi D, Blauvelt A, Reich K, *et al*. Secukinumab is superior to ustekinumab in clearing skin of subjects with moderate to severe plaque psoriasis: CLEAR, a randomized controlled trial. *J Am Acad Dermatol*. 2015;73(3):400-409.

    doi: 10.1016/j.jaad.2015.05.013
13. Reich K, Armstrong AW, Langley RG, *et al*. Guselkumab versus secukinumab for the treatment of moderate-to-severe psoriasis (ECLIPSE): Results from a phase 3, randomised controlled trial. *Lancet*. 2019;394(10201):831-839.

    doi: 10.1016/S0140-6736(19)31773-8
14. Gordon KB, Strober B, Lebwohl M, *et al*. Efficacy and safety of risankizumab in moderate-to-severe plaque psoriasis (UltIMMa-1 and UltIMMa-2): Results from two double-blind, randomised, placebo-controlled and ustekinumab-controlledphase3trials.*Lancet*.2018;392(10148):650-661.

    doi: 10.1016/S0140-6736(18)31713-6
15. Reich K, Pinter A, Lacour JP, *et al*. Comparison of ixekizumab with ustekinumab in moderate-to-severe psoriasis: 24-week results from IXORA-S, a phase III study. *Br J Dermatol*. 2017;177(4):1014-1023.

    doi: 10.1111/bjd.15666
16. Blauvelt A, Papp KA, Griffiths CEM, *et al*. Efficacy and safety of guselkumab, an anti-interleukin-23 monoclonal antibody, compared with adalimumab for the continuous treatment of patients with moderate to severe psoriasis: Results from the phase III, double-blinded, place- and active comparator-controlled VOYAGE 1 trial. *J Am Acad Dermatol*. 2017;76(3):405-417.

    doi: 10.1016/j.jaad.2016.11.041
17. Lebwohl M, Strober B, Meneter A, *et al*. Phase 3 studies comparing brodalumab with ustekinumab in psoriasis. *N Engl J Med*. 2015;373(13):1318-1328.

    doi: 10.1056/NEJMoa1503824
18. Reich K, Gooderham M, Green L, *et al*. The efficacy and safety of apremilast, etanercept and placebo in patients with moderate-to-severe plaque psoriasis: 52-week results from a phase IIIb, randomized, placebo-controlled trial (LIBERATE). *J Eur Acad Dermatol Venereol*. 2017;31(3):507-517.

    doi: 10.1111/jdv.14015
19. Ryeznik Y, Sverdlov O, Svensson EM, Montepiedra G, Hooker AC, Wong WK. Pharmacometrics meets statistics-A synergy for modern drug development. *CPT Pharmacometrics Syst Pharmacol*. 2021;10:1134-1149.

    doi: 10.1002/psp4.12696

Appendix

R code

```
# --- [0] R libraries
library(tidyverse)
library(gnm)
# --- [1] Data Setup
# summary-level dataframe
df_input_summary <- structure(
list(
dose_group = structure(1:5, levels = c("0 mg", "1 mg", "5 mg", "10 mg", "20 mg"), class = "factor"),
dose_val = c(0L, 1L, 5L, 10L, 20L),
time = c(12L, 12L, 12L, 12L, 12L),
endpoint = c("Endpoint A", "Endpoint A", "Endpoint A", "Endpoint A", "Endpoint A"),
n_success = c(4L, 12L, 27L, 42L, 42L),
n_fail = c(46L, 38L, 23L, 8L, 8L),
n_subjs = c(50L, 50L, 50L, 50L, 50L),
prop_resp = c(0.08, 0.24, 0.54, 0.84, 0.84)
),
row.names = c(NA, -5L),
class = c("tbl_df", "tbl", "data.frame"))
# individual-level dataframe
df_input_individual <- structure(
list(
ID = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71,
72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92,93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104,
105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130,
131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156,
157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182,
183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208,
209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234,
235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250),

dose_group = structure(c(1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L,
1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L,
2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L,
2L, 2L, 2L, 2L, 2L, 2L, 2L, 2L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L,
3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 3L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L,
4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L,
4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 4L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L,
5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L,5L, 5L), levels = c("0 mg", "1 mg",
"5 mg", "10 mg", "20 mg"), class = "factor"),

dose_val = c(0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L,0L, 0L, 0L, 0L, 0L,
0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L, 0L,
1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L,
1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L,
1L, 1L, 1L, 1L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L,
5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 5L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L,
10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L,
10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 10L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L,
20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L,
20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L, 20L),
```

```
time = c(12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L,
12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L,
12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L,
12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L,
12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L,
12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L,
12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L,
12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L,
12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L,
12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L, 12L,
12L, 12L),

endpoint = c(“Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint
A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”,
“Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint
A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”,
“Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint
A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”,
“Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint
A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”,
“Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint
A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”,
“Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint
A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”,
“Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint
A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”,
“Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint
A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”,
“Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint
A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”,
“Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint
A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”,
“Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint
A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”,
“Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint
A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”,
“Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint
A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”,
“Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint
A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”,
“Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint
A”, “Endpoint A”, “Endpoint A”, “Endpoint A”, “Endpoint A”),

response = c(1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0)

),
row.names = c(NA, -250L),
class = c(“tbl_df”, “tbl”, “data.frame”)
)
# --- [2] GLM example fit
```

```
glm.mod1.fig1A <- glm(
formula = prop_resp ~ dose_val,
weights = n_subjs,
family = binomial(link = “logit”),
data = df_input_summary
)
glm.mod1.fig1B <- glm(
formula = cbind(n_success, n_fail) ~ dose_val,
family = binomial(link = “logit”),
data = df_input_summary
)
glm.mod1.fig1C <- glm(
formula = response ~ dose_val,
family = binomial(link = “logit”),
data = df_input_summary
)
# --- [3] GNM example fit
# define fx_emax
fx_emax <- function(x){
list(
predictors = list(Emax = 1, ED50 = 1),
variables = list(substitute(x)),
term = function(predLabels, varLabels) {
sprintf(
“%s*%s/(%s + %s) ”,
predLabels[1], varLabels[1],
predLabels[2], varLabels[1]
)
}
)
}
class(fx_emax) <- “nonlin”
# fit gnm
gnm.mod2.fig1A <- gnm(
formula = prop_resp ~ fx_emax(dose_val),
weights = n_subjs,
family = binomial(link = “logit”),
data = df_input_summary
)
glm.mod2.fig1B <- gnm(
formula = cbind(n_success, n_fail) ~ fx_emax(dose_val),
family = binomial(link = “logit”),
data = df_input_summary
)
glm.mod2.fig1C <- gnm(
formula = response ~ fx_emax(dose_val),
family = binomial(link = “logit”),
data = df_input_summary
    )
```