



A peer reviewed, open-access electronic journal: ISSN 1531-7714

Power Analysis for Moderated Multiple Regression: An Incremental Model-Building Approach Using R

Ethan C. Brown, *Fulbright University Vietnam* 
 Mohammed A. A. Abulela, *University of Minnesota* 

Abstract: Moderated multiple regression (MMR) has become a fundamental tool for applied researchers, since many effects are expected to vary based on other variables. However, the inherent complexity of MMR creates formidable challenges for adequately performing power analysis on interaction effects to ensure reliable and replicable research results. Prior literature indicates interaction effects are frequently underpowered, and that researchers should attend to the power implications of subtle suppression/enhancement effects, measurement error, and range restriction, not to mention the prevalence of small effect sizes. Despite existing tools and guidance related to MMR power analysis, we have not seen a practical framework for guiding applied researchers and practitioners through this challenging process. In response, we developed an incremental model-building framework that allows for a systematic step-by-step approach to MMR power analysis in R with the *InteractionPowerR* and *simpr* packages using an R^2 change approach. Using the proposed approach, researchers ground their analysis in prior empirical research, and build sequentially more sophisticated power analyses to illuminate the intricacies of their MMR while managing cognitive complexity. We demonstrate the framework through an applied example, with full R code provided, as a resource to support applied researchers and practitioners in their study planning and decision making and to improve the empirical knowledge base. This tutorial is expected to substantially improve practices of conducting power analysis needed to test interaction effects in educational and psychological studies, as well as inspire software development to address current practical challenges in performing power analysis.

Keywords: Moderated Multiple Regression; Interaction Effects; Power Analysis; Sample Size

This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY-4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <https://creativecommons.org/licenses/by/4.0/>

 OPEN ACCESS.

Introduction

According to methodological recommendations, power analysis is recommended due to the prevalence of underpowered tests (e.g., Simmons et al., 2011), and scholars have pointed out how research in the social and behavioral sciences particularly shows “shockingly low levels of power” (Murphy & Myers, 2023, p. 17). Statistical power has been defined as the probability of correctly rejecting a false null hypothesis (e.g., Howell, 2011), or equivalently, the ability of a statistical test to detect an effect which exists in the population (Field, 2013; Lix & Keselman, 2010). Prior researchers (e.g., Liu, 2014) have emphasized the need for power analysis as an integral component of a study design process, which should be considered early in the research process due to its importance for the validity of hypothesis testing. Albers (2017) elaborated on the need for power analysis in the study design stating that collecting more data than required to detect a specific effect was associated with increased time and expenses, whereas fewer subjects could negatively impact power analysis. Additionally, Aberson (2019) argued that nonsignificant results associated with underpowered studies likely increased the possibility of unpublished work, which in turn could reduce the replicability of published findings.

With this in mind, several researchers have accordingly argued against the utility of *post hoc* power analyses based on observed effect sizes, since they provide no information beyond the reported *p* values and constructed confidence intervals (e.g., Hoenig & Heisey, 2001). Instead, *a priori* power analysis, conducted during the study design phase of a research project, can ensure that the planned study has sufficient power in order to address its research question. Accordingly, many resources have been provided for *a priori* power analysis, including textbooks (e.g. Aberson, 2019), tutorials (e.g. Anderson, 2019), software (e.g. G*Power; Faul et al., 2007), and R packages (e.g. *pwr2ppk*; Aberson, 2022). However, these resources can be confusing due to the wide variety of tools and approaches used, leaving applied researchers and practitioners with important open questions for how to actually apply power analysis to practical problems:

1. What tools should be used for a specific power analysis?
2. What are reasonable assumptions for a given power analysis?
3. How should the often contradictory results from different methods/tools for power analysis be integrated?

These questions are arguably less relevant with the textbook forms of power analysis applied to simpler tests, such as *t*-test, ANOVA, or simpler applications of regression. Despite the availability of many resources and articles regarding power analysis for moderated multiple regression (MMR; e.g., Aguinis, 1995; Baranger et al., 2023), there has been no comprehensive applied framework to conduct power analysis for this statistical technique, which is commonly used in applied educational, psychological, and other social science research (O'Boyle et al., 2019; Shieh, 2009). We begin to address this gap with an incremental model-building framework using the R statistical programming environment (R Core Team, 2024), based on an R^2 change approach. In the next sections, we start with a brief introduction to MMR, then we discuss the challenges of power analysis in MMR and the advantages of an incremental model-building framework for addressing these challenges. We conclude the paper with an applied tutorial and worked example for building and comparing power analysis, integrating approaches from analytic power calculators and black box simulations from the MMR-focused package *InteractionPowerR* (Baranger et al., 2023), and general-purpose custom simulations using the *simpr* package (Brown, 2023).

This tutorial has the potential to substantially advance practices of conducting MMR power analysis among applied researchers and practitioners (e.g., evaluators), among others. The proposed incremental model-building approach for conducting power analysis in R using an R^2 change approach has the advantage of starting with simplicity and only adding layers of complexity as needed, which increases the practical utility of the approach and the potential for adoption. Additionally, the proposed framework integrates seemingly

contradictory MMR power analysis techniques to provide initial insights into the relationship between model assumptions and hypothesis tests. Finally, this incremental framework has the potential to be extended to other models, designs, and study planning techniques.

MMR in Educational and Psychological Research

Many effects of interest to educational and psychological researchers are theorized to vary based on other variables, necessitating the use of appropriate techniques like MMR to quantify these interactions (for a comprehensive review of MMR models, readers are referred to Aguinis, 2004). As an educational example, suppose that a researcher is interested in investigating the interaction effect between cognitive engagement and sex on fifth grade students' math achievement scores, controlling for student motivation and socioeconomic status (SES).

Then, the researcher wants to determine the adequate sample size needed to be able to reliably detect the interaction between cognitive engagement and sex. Power analysis is especially important for interaction effects, where effect sizes tend to be small, but it remains challenging to integrate the many tools and approaches for practical applications. Due to the inherent complexity of interactions, one of the potential challenges facing the researcher is the lack of a cohesive power analysis tutorial for computing the required sample size for this study, despite many resources both for regression in general (e.g., Beaujean, 2014), and for MMR specifically (e.g., Baranger et al., 2023). Inadequate consideration of power can lead to underpowered interaction effects, leading to negative consequences for the quality of scientific literature.

Addressing Underpowered Interaction Effects

Power analysis is essential for MMR, where the impact of not detecting interaction effects is well-documented. Aguinis and colleagues (2005) and Shieh (2009) noted that failure to detect interaction effects can be harmful for theory building, since research might incorrectly discard theory-based hypotheses and models. This could have serious implications for practice. If, for instance, an interaction effect is not detected between sex (males/females) and cognitive engagement on math achievement as the outcome variable, this can lead to missed interventions particularly if cognitive engagement negatively affects math achievement for a specific sex group (e.g., males). Conversely, underpowered studies in combination with publication bias can lead to inflated estimates of effect sizes (e.g., Open Science Collaboration, 2015).

What can lead to these undetected effects in MMR? Among other factors, Jaccard and colleagues (1990) implicated insufficient statistical power as a major contributor to undetected interaction effects. Aguinis (1995) and Aguinis and Gottfredson (2010) also discussed the so-called detrimental factors to statistical power in MMR and provided some best practice recommendations for estimating interaction effects. First, variable distributions including range restrictions (i.e., truncated scores in the predictor) and error variance heterogeneity (i.e., subgroups differ in their error variance) were found to affect the ability of MMR to detect moderation effects. Relatedly, statistical power was reduced when the predictor variable variance was negatively biased due to selection or range restriction as well as artificial dichotomization, each of which reduces variability (Aberson, 2019). Second, measurement error due to unreliability of the predictor and criterion variables has been found to reduce power. Relatedly, artificial moderator dichotomization and polychotomization has also been found to reduce statistical power. Third, a larger sample size is positively associated with statistical power. Conversely, in cases where there is an unequal sample size in each moderator-based subgroup, statistical power is reduced for that categorical moderator (e.g., sex). Last, according to Aguinis & Gottfredson (2010), higher predictor intercorrelation can lead to unstable regression coefficients, larger errors, and decreased statistical power, although see comments in Baranger et al. (2023) regarding the variable impact of intercorrelation on power. For more information regarding these four

factors and their impact on statistical power analysis, interested readers are advised to review Aguinis and Stone-Romero (1997).

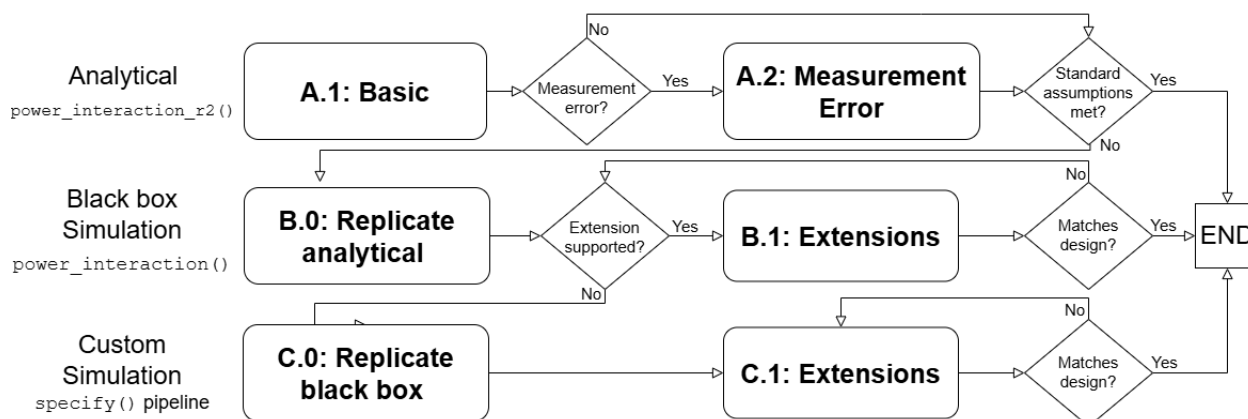
Other factors paramount to regression and power analysis, but less investigated, are suppression and enhancement, which are typically related (Beckstead, 2012; Lewis & Escobar, 1986; Shieh, 2006). Suppression refers to a situation where the amount of variance explained for two predictor variables together is less than the sum of each predictor variable, whereas enhancement refers to the converse situation where the amount of variance explained for two predictor variables together is greater than the sum of each predictor variable (Friedman & Wall, 2005). Taken together, suppression and enhancement can impact MMR power analyses in two different directions. Suppression can reduce the power to detect interaction effects, whereas enhancement can increase power (Baranger et al., 2023). Therefore, power analysis techniques which account for suppression and enhancement, such as more sophisticated analytical approaches as well as simulation-based approaches, result in more accurate estimates of statistical power.

How can applied researchers and practitioners navigate this maze of complex influences on the power to detect interaction effects? The proposed framework and associated tutorial for MMR power analysis, described in greater detail below, gradually incorporates complex factors into the power analysis in a way that clarifies how much impact each of these factors is expected to have on the resulting power.

An Incremental Model-Building Approach to MMR Power Analysis

This article does not introduce any new analysis techniques; instead, we provide a practical and extensible framework in which to organize and integrate existing power analysis techniques for MMR using the R statistical programming environment (R Core Team, 2024), using an R^2 change approach. When there are many different techniques for power analysis, it can be unclear to applied researchers and practitioners when a given approach should be used, and what the differences in results between different techniques are. We here propose a process for power analysis of MMR that integrates multiple approaches incrementally to gain an overall picture of statistical power for a given study that can guide more informed and responsible decision making. The proposed approach, shown in Figure 1, progresses through Analytical, Black box Simulation, and Custom Simulation (ABC) phases with seven incremental stages, each of which represents increasingly complex power analysis models. Although the scope of the presented approach uses an R^2 change approach and normally distributed predictors, and is specific to currently available R functions—in particular those from the *InteractionPowerR*, and *simpr* packages—we deliberately describe the overall process in general terms so that that readers can tailor the approach to their context and preferred packages.

Figure 1. An incremental approach to moderated multiple regression (MMR) power analysis, progressing from simple analytic analysis, through black box simulations, to custom simulations.



We recommend beginning the model-building process by incrementally using increasingly complex analytical power analyses (stages A.1–A.2). Analytical power calculators estimate power based on the specification of input parameters, usually with strong distributional assumptions, and use mathematical calculations to estimate power or the required sample size. The simplest model in Stage A.1 accounts for the basic structure of interaction models, using the *InteractionPoweR* package (Baranger et al., 2023). Stage A.2, again using *InteractionPoweR*, additionally accounts for measurement error within an analytical framework. Comparing the results of each of these analytical models already provides a lot of useful information for many applied research projects, both in terms of the required sample size for a given degree of power but also for understanding the relative impact of various factors on power. For instance, power can be improved not only by increasing the sample size, but also by decreasing measurement error.

After relaxing as many assumptions as possible using analytic tools, the applied researcher can build on these analyses and use simulation to more closely match their research design. However, building simulations from scratch can be complex and we have found that it is easy for researchers to create simulations that do not align with their research design, or that are so opaque that they are difficult to update or tweak. Therefore, we recommend that initial simulations are what we call “black box simulations” — simulation functions that from the user perspective are quite similar to analytic methods, where the user simply specifies their parameters or ranges of parameters and receives power analysis output without explicitly specifying the models and assumptions. However, these functions, such as those provided by the *InteractionPoweR* package, may allow the relaxing of more assumptions due to the flexibility of simulation analysis. Because of their black box nature, these simulations may also make *different* assumptions than the analytical model, and the differences may not always be clear from the documentation. Therefore, in stage B.0 we recommend first *replicating* the analytic results with any black box simulation tool. Once this step is complete, the analyst can then explore any available extensions provided by the black box simulation tool (stage B.1). For instance, the black box simulations in *InteractionPoweR* currently support binary and ordinal variables.

Yet even more flexibility can be found in the creation of custom simulations, which allow the user to explicitly specify the assumptions of the distributions. Both analytic and black box simulations have implicit restrictions and assumptions that are built into their functions. In contrast, a custom-built simulation using the *simpr* package (Brown, 2023) can accommodate any arbitrary type of model or data-generating process. However, this explicit specification also creates challenges and opportunities for error, so we again in stage C.0 recommend replicating the black box simulation first to ensure consistency and comparability. Stage C.0 has the added advantage of making assumptions explicit by fully specifying the data generating process in code. The R example below shows how the results from a custom-built simulation in *simpr* match the results from *InteractionPoweR*. Again, now we can proceed to extending the model by relaxing assumptions in stage C.1, where we give an example of how a range restriction, and then non-normal error distributions, can affect the power analysis.

The incremental nature of the proposed approach allows it to address the limitations of the currently scattered literature of different types of power analysis for interaction effects. Moreover, this approach explicitly includes comparisons between differing approaches so that a more standardized and unified operating procedure can be followed by applied researchers.

R Tutorial: Implementing the Incremental Model-Building Approach for MMR Power Analysis

Below, we demonstrate how to carry out the incremental model-building approach using R, and how it can directly impact practice. The context for our example is an observational study of how general anxiety and Facebook-specific anxiety predict Facebook usage (McCord et al., 2014). McCord et al. (2014) found

strong evidence for an interaction between general anxiety and Facebook-specific anxiety in a linear regression predicting Facebook usage, whereas a replication study found little evidence of the interaction (Sillence et al., 2021). Sillence et al. (2021) performed an *a priori* simulation-based power analysis of this scenario concluding that a sample of size greater than 350 would be necessary to achieve a power of .80. However, this power analysis had several major limitations. First, the authors assumed an interaction effect size of $\Delta R^2 = 0.0225$ and called it a “moderate” effect size, whereas even $\Delta R^2 = 0.01$ is considered a very large effect size for an observational study (Aberson, 2019, p. 141). Second, the authors relied on an approximate association between standardized coefficients and correlation coefficients to set up their simulation (Peterson & Brown, 2005), and their simulations actually produce effect sizes much smaller than intended. Third, the simulation assumed that the two predictors had no correlation with each other; the two predictors are both measures of anxiety and would be theoretically predicted to have strong correlations. Moreover, empirical results from the data suggest correlations higher than 0.5 (Sillence et al., 2021), and the correlation between predictors is known to have a strong influence on power (Aberson, 2019; Baranger et al., 2023). Finally, score reliability issues were not addressed, even though these scales were based on measured variables from surveys.

We show below how this situation could instead be analyzed using the incremental model-building approach to MMR power analysis. Throughout, we use a more realistic, though still large, effect size for the interaction: $\Delta R^2 = 0.01$ (Aberson, 2019, p. 141). The incremental model-building approach allows a clearer comparison of different methods and assumptions, as well as allowing for more flexibility in relaxing assumptions. We conclude the section with a comparison of the results of all simulations. As we shall see, the estimates of power provided in Sillence et al. (2021) are extremely optimistic by any standard.

Although many simpler forms of statistical analysis and designs have published rules of thumb for the required sample size, such as for *t*-tests, correlations, and ANOVA (Brysbaert, 2019), the added complexities of MMR preclude simple rules that can apply to all studies. The closest to rules of thumb we were able to locate for the power to detect regression interactions is given by Sheih (2009), which used a random regression framework to derive estimates for required sample size to achieve various levels of power for bivariate normally distributed and gamma distributed predictor and moderator variables. However, the parameter values specified in Sheih (2009) do not straightforwardly match the context of our study, which among other differences involved predictor-outcome relationships with opposite signs. Therefore, we proceed to the analytical power calculations in step A.1.

A.1 Analytical: Basic

We start with a basic analytical power analysis for Sillence et al. (2021) Study 2. The *InteractionPowerR* package provides both analytic and simulation-based tools for estimating power for interaction terms. The analytic power function `power_interaction_r2()` estimates the power for a change in R^2 (assuming normally distributed predictors) using an analytic approach similar to Shieh (2009), allows systematic variation of any parameter, and allows for measurement error. As an initial step, this function produces similar results to `regintR2()` from the *pwr2ppl* package (Aberson, 2022), but allows for further extensions such as allowing systematic variation of any parameter, and correction for measurement error (see step A.2).

Because suppression and enhancement affect the R^2 , the input effect size is specified in terms of a correlation between the interaction term and the response variable rather than as a change in R^2 .

```
library(InteractionPowerR)
a.1_basic =
  power_interaction_r2(
    N = seq(500,          # Minimum sample size examined
            1500,         # Maximum sample size examined
```

```
      by = 25),      # Interval of sample sizes attempted
  r.x1.y = -0.18,    # Correlation of x1 with y (response)
  r.x2.y = -0.03,    # Correlation of x2 with y
  r.x1.x2 = 0.64,    # Correlation of x1 with x2
  r.x1x2.y = 0.1,    # Correlation of interaction with y
  alpha = 0.05)      # Alpha-level for rejecting null
```

The sample size, specified via the `N` argument, incorporates the `seq()` command, which creates a sequence of all possible sample sizes to be produced by the model — here, going from 500 to 1500 by 25. Here, `y` represents Facebook usage, `x1` represents Facebook-specific anxiety, and `x2` represents social anxiety. The above code specifies the correlations between each predictor and the outcome (`r.x1.y` for Facebook-specific anxiety with Facebook usage, and `r.x2.y` for social anxiety with Facebook usage), the correlation between the Facebook-specific anxiety and social anxiety (`r.x1.x2`). The effect size is here expressed as the correlation between the interaction and `y` (`r.x1x2.y`) as 0.10, the square root of the R^2 change specified above — but since `power_interaction_r2()` incorporates some subtle analysis of enhancement and suppression effects, the actual R^2 change implied by the model will not always align with the squared correlation, even though the interaction effect is assumed to be uncorrelated with the main effects. This analysis reveals that the minimum sample size required to achieve a power of .80 is approximately 750 (see Figure 2, below).

If, say, the sample size were already fixed at 500, one could alternatively estimate the minimum detectable effect size at power = .80 by varying the possible values of `r.x1x2.y`, again using the `seq()` command, here varying from a correlation with the outcome of 0.1 to 0.2:

```
a.1_vary_cors =
  power_interaction_r2(
    N = 500,          # Sample size examined
    r.x1.y = -0.18,   # Correlation of x1 with y (response)
    r.x2.y = -0.03,   # Correlation of x2 with y
    r.x1.x2 = 0.64,   # Correlation of x1 with x2
    r.x1x2.y = seq(0.1,      # Min x1:x2 correlation
                  0.2,      # Max x1:x2 correlation
                  by = 0.01), # Interval for correlation
    alpha = 0.05)      # Alpha-level for rejecting null
```

The ability to dynamically vary parameters easily allows the user to more fully explore the parameter space and perform sensitivity analysis by varying the assumed correlations between predictors to explore the impact on power. This initial analytic stage is an ideal place for these explorations because the model is so far relatively simple, results are quickly estimated, and the impact of individual changes in parameters are more easily tracked.

A.2. Analytical: Accounting for Measurement Error

Importantly, score reliability of each scale can also be incorporated into the calculation of power via `power_interaction_r2()`. Based on the reported score reliabilities from Sillence et al. (2021), these can be entered into the function by specifying the score reliability of the outcome (`rel.y`), and each predictor (`rel.x1` and `rel.x2`):

```
a.2_measurement =
  power_interaction_r2(
    N = seq(500,          # Minimum sample size examined
```

```
      1500,      # Maximum sample size examined
      by = 25), # Interval of sample sizes attempted
r.x1.y = -0.18, # Correlation of x1 with y (response)
r.x2.y = -0.03, # Correlation of x2 with y
r.x1.x2 = 0.64, # Correlation of x1 with x2
r.x1x2.y = 0.1, # Correlation of interaction with y
rel.y = 0.79,   # Reliability of y
rel.x1 = 0.89,  # Reliability of x1
rel.x2 = 0.95,  # Reliability of x2
alpha = 0.05)   # Alpha-level for rejecting null
```

The required sample size, now incorporating both enhancement/suppression and measurement error, is now approximately 1075, quite a bit higher than the 750 originally computed in Stage A.1 using more simplistic assumptions. Note, however, that the presence of measured variables may also indicate that a latent variable approach such as structural equation modeling should be used instead of moderated multiple regression. Power analysis for latent variable approaches is certainly possible in R (e.g. Moshagen & Bader, 2024), but are beyond the scope of this paper.

B.0 Black Box Simulations: Replicating Analytical Approaches

Analytical power analysis, while computationally inexpensive and relatively straightforward, has restrictive assumptions that are not always met in practice. The `power_interaction()` function from the *InteractionPoweR* package (Baranger et al., 2023) additionally allows ordinal predictors and outcomes, and utilizes a simulation-based approach. Rather than calculating the power mathematically, the function instead actually simulates many datasets under the same parameters and assumptions, fits a linear regression model, and estimates the power to detect an effect by the proportion of the time the null hypothesis is correctly rejected at a given alpha level. We can create a simulation using very similar syntax to `power_interaction_r2()` with additions **bolded** below:

```
set.seed(131051)      # Set random seed for reproducibility
b.0_replicate =
  power_interaction(
    c1 = 6,              # Number of computing clusters
    n.iter = 10000,     # Number of simulation replications
    N = seq(500,        # Minimum sample size examined
            1500,       # Maximum sample size examined
            by = 25),   # Interval of sample sizes attempted
    r.x1.y = -0.18,     # Correlation of x1 with y (response)
    r.x2.y = -0.03,     # Correlation of x2 with y
    r.x1.x2 = 0.64,     # Correlation of x1 with x2
    r.x1x2.y = 0.1,     # Correlation of interaction with y
    rel.y = 0.79,       # Reliability of y
    rel.x1 = 0.89,      # Reliability of x1
    rel.x2 = 0.95,      # Reliability of x2
    alpha = 0.05)       # Alpha-level for rejecting null
```

As is typical in computer simulations, we set the random “seed” to an arbitrary number. This allows the code to be reproducible — to produce the same result every time given a certain seed. The only new arguments compared to Stage A.2 are `c1`, which specifies how many computing clusters the simulation should use on the user’s computer; and `n.iter`, which specifies how many simulations of each condition

should be generated. Generally, the more clusters you assign to a computing task, the faster the simulation will run and the more of your computer's resources will be occupied while the machine is running. The time savings will generally not be quite proportional to the number of clusters assigned, due to features of individual computers and the tasks assigned. You cannot assign more than the number of clusters available on your machine, and we generally recommend assigning one less than that number to allow you to perform other tasks on your computer while the simulation is running. For more discussion about parallel computing in R, the interested reader is referred to Bengtsson (2021). The above command generates 410,000 simulations, since there are 41 values of sample size multiplied by 10,000 iterations. The command then calculates the percentage of the time that the analysis results in a significant interaction as a way of calculating power.

The `power_interaction()` function again assumes that all variables and errors are normally distributed and centered, just like the analytical function `power_interaction_r2()`. Since it is based on the same assumptions, this simulation-based approach results in very similar estimates to the analytical approach, with a sample size of approximately 1100 required to reach a power of .80. Rerunning the simulation provides an important triangulation of the validity of both the analytic and simulation estimates. A prior version of *InteractionPoweR* resulted in different power estimates between these two functions given high predictor intercorrelations, which has been recently corrected (Baranger, 2025, personal communication). The advantage of an easy-to-use black box simulation tool, however, is that it additionally can support additional extensions to the model that analytic methods cannot easily handle, such as ordinal and binary predictors.

B.1 Black Box Simulations: Extending Analytical Approaches

Black box simulations can extend beyond the analytical approach by relaxing assumptions. Currently, the only additional available functionality in `power_interaction()` is the ability to specify some of the variables as binary or ordinal using `k.y`, `k.x1`, and `k.x2` to specify the number of levels of the associated variable. For instance, if the outcome variable of Facebook usage had four levels (`k.y = 4`), and Social Anxiety was a binary variable (`k.x2 = 2`), appropriate values of the arguments could be included as shown below in **bold**:

```
set.seed(131051)                # Set random seed for reproducibility
b.l_extensions =
  power_interaction(
    cl = 6,                        # Number of computing clusters
    n.iter = 1000,                 # Number of simulation replications
    N = seq(500,                  # Minimum sample size examined
           1500,                  # Maximum sample size examined
           by = 25),              # Interval of sample sizes attempted
    r.x1.y = -0.18,               # Correlation of x1 with y (response)
    r.x2.y = -0.03,               # Correlation of x2 with y
    r.x1.x2 = 0.64,               # Correlation of x1 with x2
    r.x1x2.y = 0.1,               # Correlation of interaction with y
    rel.y = 0.79,                 # Reliability of y
    rel.x1 = 0.89,                 # Reliability of x1
    rel.x2 = 0.95,                 # Reliability of x2
    alpha = 0.05,                 # Alpha-level for rejecting null
    k.y = 4,                      # Number possible values for y
    k.x2 = 2)                   # Number possible values for x2
```

Although the outcome is ordinal, the above simulation would still generate, fit, and report the power for MMR using ordinary least squares regression; but in the case of `k.y = 2` a logistic regression would be

used instead. We do not report the results of the above simulation, since it is just an example and is not relevant to the analysis. However, we now turn to custom simulation approaches to validate the black box simulation as well as to examine potential effects of range restriction and error distribution.

C.0 Custom Simulations: Replicating Model-Specific Approaches

Although black box simulation tools such as *InteractionPoweR* (Baranger et al., 2023) enable more flexibility than analytic power calculators, there are still scenarios outside of what those tools are able to model, since the convenience of having a black box simulation tool comes at the cost of only making available the particular simulation scenarios that the function author expressly supports. In contrast, a custom simulation tool such as *simpr* (Brown, 2023) opens the black box and supports the analyst in creating simulations from scratch that are only limited by the analyst's technical skills and imagination. Here, we demonstrate how the *simpr* package provides the flexibility to extend the model by exploring the impact of range restriction and non-normal errors on statistical power.

Creating a custom simulation is conceptually challenging. Our incremental approach ensures that custom simulations work as intended by initially replicating a simpler method (here, a black box simulation). This sets up the basic structure of a simulation and ensures that the simulation matches prior stages before varying distributions or parameters beyond what analytic and black box simulation tools can support. This replication stage involves two steps: a) calculating appropriate parameters for the simulation, and b) simulating and calculating power with *simpr*.

Calculating Appropriate Parameters for the Simulation. Our simulation approach involves simulating the data from the population regression equation. How do we calculate the appropriate parameters, such as population slopes, from the correlations and score reliabilities specified so far? Because we have already done the analytic power calculation, we do not have to calculate our simulation parameters from scratch. The `interaction_params()` function from the *simpr.interaction* package (Brown, 2025) leverages the `power_interaction_r2()` in the *InteractionPoweR* package (also used in from stages A.1 and A.2, above; Baranger et al., 2023) to calculate the appropriate slopes and adjusted predictor correlation matrix to use in the simulation:

```
library(simpr.interaction)
p = interaction_params(
  r.x1.y = -0.18,      # Correlation of x1 with y (response)
  r.x2.y = -0.03,      # Correlation of x2 with y
  r.x1.x2 = 0.64,      # Correlation of x1 with x2
  r.x1x2.y = 0.1,      # Correlation of interaction with y
  rel.y = 0.79,        # Reliability of y
  rel.x1 = 0.89,        # Reliability of x1
  rel.x2 = 0.95,        # Reliability of x2
)
```

The same parameters used in stages A.2 and B.0 with `power_interaction_r2()` and `power_interaction()` (correlations and reliabilities) are again repeated here. The `interaction_params()` function returns an object which contains the key information necessary for a simulation, including the following components:

```
$b1          # Implied slope of x1
[1] -0.207

$b2          # Implied slope of x2
[1] 0.0961
```

```
$b3                                # Implied slope of interaction
[1] 0.0839

$r2                                # R-squared of full model
[1] 0.0359

$sigma                             # Regression sigma
[1] 0.982

$cov                                # Predictor correlation matrix
      [,1]      [,2]
[1,] 1.0000000 0.5884869
[2,] 0.5884869 1.0000000
```

The three population regression coefficients, b_1 , b_2 , and b_3 that match the given input parameters are calculated, adjusting for the provided correlations and measurement error, as well as the R^2 (r^2), σ (the population error standard deviation), and cov , the predictor correlation matrix.

We are now ready to create simulated data and fit models based on these parameters.

Simulating and Calculating Power with *simpr*. The *simpr* package (Brown, 2023) makes it easy to vary parameters, generate many simulated datasets, and compile results for further plotting and analysis. The *simpr* workflow distills a simulation study into five primary steps:

1. `specify()` your data-generating process
2. `define()` parameters that you want to systematically vary across your simulation design (e.g. n , effect size)
3. `fit()` models to your data (e.g., `lm()`)
4. Compile model results using `tidy_fits()`
5. `generate()` the simulation data, fits, and model results

These steps can be piped together using the piping operator from the *magrittr* package (Bache & Wickham, 2022), `%>%`, for readability and flow. To demonstrate the basic syntax, we first show a simple example where we run a simple MMR simulation with $n = 100$, and then run this simulation 50 times. This has no varying parameters so we skip step #2, `define()`.

```
set.seed(131051)                    # Set random seed for reproducibility
c.0_example =
  specify(                            # Specify data-generating process
    X =                                # Generate predictor matrix
      c(x1, x2) ~                      # Name predictors x1 and x2
      MASS::mvrnorm(                  # Generate as multivariate normal
        n = 100,                      # - Sample size of 100
        mu = c(0, 0),                # - Mean 0 for both predictors
        Sigma = p$cov),              # Covariance matrix (specified above)
    y =                                # Simulate outcome: sum of
      ~ p$b1 * x1 +                    # Slope times x1
      p$b2 * x2 +                      # Slope times x2
      p$b3 * x1 * x2 +                # Slope times interaction
      rnorm(n = 100,                 # Random normal error
            sd = p$sigma)             # sd matches sigma
```

```

    )) %>%
fit(lm =~ lm(y ~ x1*x2)) %>% # Fit linear interaction model
tidy_fits() %>% # Run broom::tidy on each model
generate(50) # Generate 50 replications

```

First, we set the seed to an arbitrary number to ensure reproducibility. Each argument of `specify` is an R formula (involving the `~` symbol) that *simpr* uses for generating the data. We create the X variable matrix by specifying the names of the variables using `c(x1, x2)` on the left-hand side of the formula and use the function `MASS::mvrnorm()` (from the *MASS* package, included with R by default; Venables & Ripley, 2002) to draw multivariate normal data with the specified covariance structure that we extracted above (`Sigma = p$cov`) and a mean of 0 for simplicity, since we are not evaluating the intercept (`mu = c(0, 0)`). Similarly, we create `y` using a formula: Now that we have the predictors, we can compute `y` as the sum of the predictors and the interaction term, weighted by the population slopes `b1`, `b2`, and `b3`; plus normally distributed error with standard deviation `p$sigma`.

Once the model is specified, we `fit()` the model using the standard R linear regression function, `lm()`. Again, `fit()` takes a formula containing whatever model-fitting functions are to be applied to every simulated dataset. Afterwards, we run `tidy_fits()` to bring all the coefficient estimates and *p*-values together into a handy table, and actually generate described simulated data, model fits, and results using the `generate()` command, specifying that we want 50 simulated datasets:

```

# A tibble: 200 × 8
  .sim_id rep Source term      estimate std.error statistic p.value
  <int> <int> <chr> <chr>      <dbl>    <dbl>    <dbl>    <dbl>
1     1     1 1m (Intercept) -0.0445  0.0955   -0.467  0.642
2     1     1 1m x1         -0.382   0.104   -3.69  0.000372
3     1     1 1m x2          0.212   0.113    1.88  0.0635
4     1     1 1m x1:x2        0.113   0.107    1.06  0.293
5     2     2 1m (Intercept)  0.115   0.119    0.963  0.338
6     2     2 1m x1        -0.175   0.141   -1.24  0.218
7     2     2 1m x2          0.212   0.126    1.68  0.0959
8     2     2 1m x1:x2        0.103   0.0764   1.35  0.180
9     3     3 1m (Intercept)  0.0407  0.102    0.399  0.691
10    3     3 1m x1        -0.167   0.123   -1.36  0.176
# [i] 190 more rows
# [i] Use `print(n = ...)` to see more rows

```

The table has one row for each of the 4 regression coefficients—(Intercept), `x1`, `x2`, and `x1:x2` (the interaction)—for each of the 50 simulated dataset, leading to $4 \times 50 = 200$ rows. Since we are only interested in the power for the interaction, we can use the *dplyr* (Wickham et al., 2023) functions `filter()` and `summarize()` for an estimate of power by filtering down to only the interaction terms and calculating what proportion of their *p*-values (`p.value`) are less than an alpha level, here 0.05:

```

library(dplyr)
c.0_example_power = c.0_example %>%
  filter(term == "x1:x2") %>% # Filter to interactions
  summarize(pwr = mean(p.value < 0.05)) # Calculate power

```

This calculates the interaction power as .14. However, we only examined one rather small sample size of 100. For fitting a more computationally intensive simulation, we take advantage of *simpr*'s support for parallel processing in order to speed up computation, using the *future* package (Bengtsson, 2021):

```
library(future)
plan(multisession,          # Allow parallel processing
     workers = availableCores() - 1) # Use all but 1 compute core
```

The `plan()` function call specifies that the “plan” of execution will be “multisession”, a type of parallel processing, and that the “workers” (i.e. clusters) will be one less than all of the computing cores available on the current machine. Again, this allows the user to still do other business on their computer while the simulation is running, if desired.

To fit a wider range of sample sizes, we add a `define()` statement to vary sample size over a range of values from 500 to 1500 by 25, similar to the *InteractionPowerR* examples above. We additionally fit many more simulations (10,000 per condition) in order to get an accurate estimate of power. Changes from the previous simulation are **bolded** below:

```
set.seed(131051)          # Set random seed for reproducibility
c.0_replicate =
  specify(                 # Specify data-generating process
    X =                    # Generate predictor matrix
      c(x1, x2) ~          # Name predictors x1 and x2
      MASS::mvrnorm(       # Generate as multivariate normal
        n = N,             # - Sample size is `N`
        mu = c(0, 0),      # - Mean 0 for both predictors
        Sigma = p$cov),    # Covariance matrix (specified above)
    y =                    # Simulate outcome: sum of
      ~ p$b1 * x1 +         # Slope times x1
      p$b2 * x2 +          # Slope times x2
      p$b3 * x1 * x2 +     # Slope times interaction
      rnorm(n = N,         # Random normal error
            sd = p$sigma)   # sd matches sigma
      ) %>%
    define(                # Define varying parameters, here `N`
      N = seq(500,         # Minimum sample size examined
            1500,         # Maximum sample size examined
            by = 25)      # Interval of sample sizes attempted
    ) %>%
    fit(lm =~ lm(y ~ x1*x2)) %>% # Fit linear interaction model
    tidy_fits() %>%           # Run broom::tidy on each model
    generate(10000)         # Generate 10,000 replications
```

Now, note that `N` appears in `specify()`, but its possible values are defined in `define()`. Otherwise the above code is identical to what appeared previously. Now, the results include varied sample sizes (small *p*-values truncated):

```
# A tibble: 1,640,000 × 9
  .sim_id      N    rep Source term      estimate std.error statistic    p.value
  <int> <dbl> <int> <chr> <chr>      <dbl>      <dbl>      <dbl>      <dbl>
1     1     500     1 lm    (Intercept)  0.0389    0.0516     0.754  0.451
2     1     500     1 lm    x1          -0.164    0.0596    -2.75  0.00617
3     1     500     1 lm    x2           0.0626    0.0560     1.12  0.265
```

```

4      1    500    1 lm    x1:x2      0.0323    0.0410    0.788 0.431
5      2    525    1 lm    (Intercept) -0.0189    0.0502   -0.377 0.706
6      2    525    1 lm    x1         -0.201    0.0558   -3.61 0.00034
7      2    525    1 lm    x2         0.0528    0.0544    0.971 0.332
8      2    525    1 lm    x1:x2      0.0676    0.0369    1.83 0.0676
9      3    550    1 lm    (Intercept) 0.0143    0.0484    0.296 0.767
10     3    550    1 lm    x1        -0.259    0.0556   -4.65 0.00000
# [i] 1,639,990 more rows
# [i] Use `print(n = ...)` to see more rows

```

There are many rows, because *simpr* generates 10,000 datasets for each possible value in `define()`. We can now compute power for each possible value of *N* with the help of `group_by()` in addition to the `filter()` and `summarize()` commands used previously:

```

c.0_replicate_power = c.0_replicate %>%
  group_by(N) %>%                                # Group by sample size
  filter(term == "x1:x2") %>%                     # Filter to interactions
  summarize(pwr = mean(p.value < 0.05))           # Calculate power

```

This produces separate power estimates for each *N*:

```

# A tibble: 41 × 2
      N    pwr
  <dbl> <dbl>
1    500 0.474
2    525 0.492
3    550 0.508
4    575 0.529
5    600 0.546
6    625 0.566
7    650 0.572
8    675 0.597
9    700 0.610
10   725 0.629
# [i] 31 more rows
# [i] Use `print(n = ...)` to see more rows

```

As shown in Figure 2 (below), the results from this replication closely match the black box simulation from step B.0, with a nearly identical power curve. However, since we now have a custom simulation set up, and have validated it compared to the black box simulation, we can now make any desired changes to the data-generating process, including imposing range restrictions and non-normal error distributions to assess the effect on power.

C.1 Custom Simulations: Extensions (range restrictions, non-normal errors, etc.)

Unlike analytic calculations and black box simulations, custom simulations are easily extensible. Since the data-generating mechanism is explicitly specified, the researcher simply makes whatever changes to the mechanism to investigate the effect of those changes on statistical power. We here demonstrate the possibilities by exploring two extensions: a range restriction and non-normal errors. These extensions are presented for pedagogical purposes to demonstrate the breadth of possibilities available to the researcher. The power calculations are not intended to be a specific criticism of Sillence et al. (2021), since there was not a particular concern with either range restrictions or non-normality raised in that study.

Range restriction. Range restriction is a common situation with applied social science data, and as reviewed above can impact power (Aguinis & Stone-Romero, 1997). For demonstration purposes, we here implement a relatively severe range restriction where all variables are not allowed to exceed approximately 1 standard deviation away from the mean (since here $p\$sigma$ is approximately 1), and see the impact on power. The data generating process is largely the same, with some custom code to restrict the range before fitting models:

```
set.seed(131051)                                # Set random seed for reproducibility
c.l_range_normal =
  specify(                                       # Specify data-generating process
    X =                                         # Generate predictor matrix
      c(x1, x2) ~                             # Name predictors x1 and x2
      MASS::mvrnorm(                           # Generate as multivariate normal
        n = N,                                # - Sample size is `N`
        mu = c(0, 0),                         # - Mean 0 for both predictors
        Sigma = p$cov),                       # Covariance matrix (specified above)
    y =                                         # Simulate outcome: sum of
      ~ p$b1 * x1 +                            # Slope times x1
      p$b2 * x2 +                              # Slope times x2
      p$b3 * x1 * x2 +                         # Slope times interaction
      rnorm(n = N,                             # Random normal error
            sd = p$sigma                       # sd matches sigma
      )) %>%
  define(                                       # Define varying parameters, here `N`
    N = seq(500,                               # Minimum sample size examined
            1500,                               # Maximum sample size examined
            by = 25)                             # Interval of sample sizes attempted
  ) %>%
  per_sim() %>%                                # Access each simulation for mutate
  mutate(across(everything(),                 # Mutate all columns
    ~ case_when(                               # Conditionally replace values:
      .x > 1 ~ 1,                               # Replace x > 1 with 1
      .x < -1 ~ -1,                             # Replace x < 1 with -1
      TRUE ~ .x))                             # Otherwise keep x the same
  ) %>%
  fit(lm =~ lm(y ~ x1*x2)) %>%                # Fit linear interaction model
  tidy_fits() %>%                             # Run broom::tidy on each model
  generate(10000)                             # Generate 10,000 replications
```

The code is identical up through the `define()` step. We then run the special command `per_sim()` to indicate that we will be doing data manipulations that will be applied to each individual simulation. Then we run `mutate()`, a *dplyr* function (Wickham et al., 2023) for adding or modifying columns to the dataset, and use `across()` (also from *dplyr*) to indicate that we truncate all variables (`everything()`) using `case_when()` so that all values greater than 1 become (\sim) 1, and all values less than -1 become -1, and otherwise (TRUE) the original value is preserved. Thus, any value exceeding 1 standard deviation from the mean is truncated to 1 for positive values, and to -1 for negative values. The remainder of the code is the same, again demonstrating the flexibility and extensibility of a custom simulation.

As shown in Figure 2, this range restriction has a severe effect on power, such that even a sample size of 1500, the max of the range estimated, only has a power of .70.

Non-normal errors. All previous steps have assumed the normality of the main effects and errors. The distributional forms of the variables and errors, however, can have consequences for statistical power (Shieh, 2009). We here show a brief demonstration of relaxing a distributional assumption by using a non-normal

error distribution, while still assuming normality of the predictor variables and including the range restriction. It is certainly possible to also have non-normal predictor variables within this custom simulation framework, but the R coding involved to ensure the same correlation structure is currently unwieldy for this pedagogical example, an issue we hope to address in future versions of the *simpr.interaction* package (Brown, 2025).

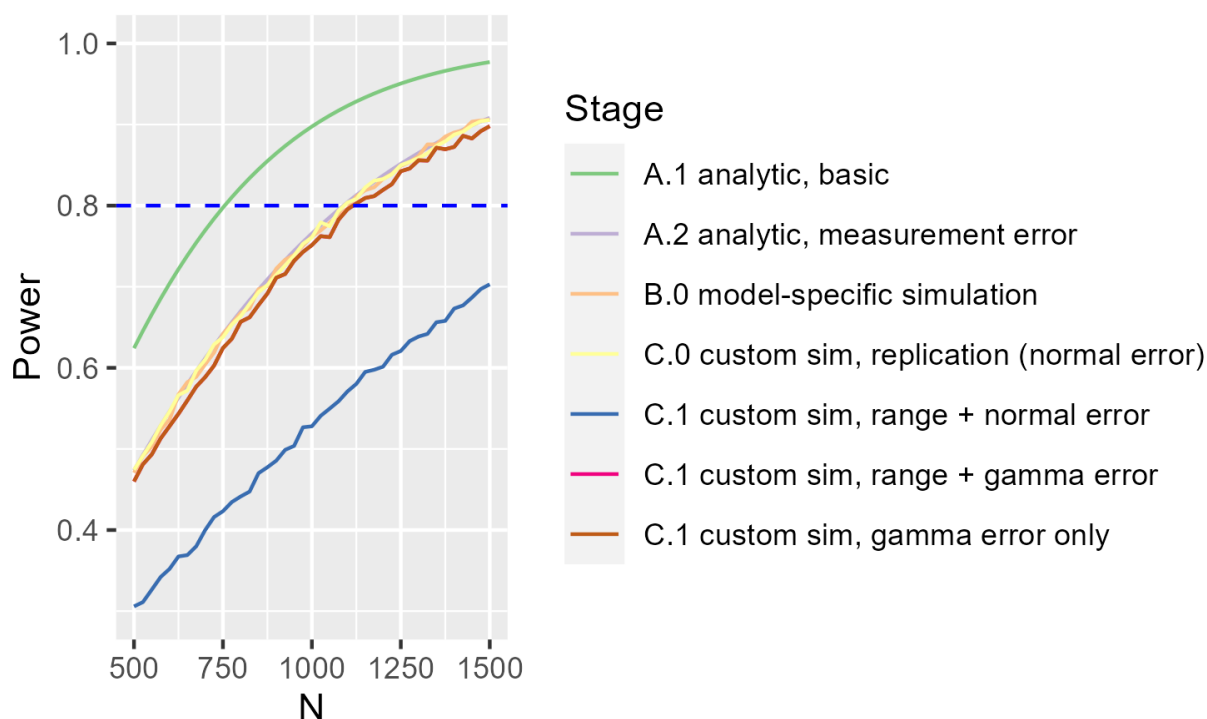
Changing the distribution of the error is essentially as simple as changing the error expression within `specify()` from a random normal variable, `rnorm()`, to a non-normal random variable. Here we use `rgamma()` to invoke a skewed-right gamma distribution with `shape = 1` and `scale = sigma` to make the standard deviation equal to `sigma`, just as in the normal simulation above in stage C.0. However, since this would produce a gamma distribution with mean `sigma`, we also subtract `sigma` from the resulting value to ensure that the error has mean 0 (as appropriate for a regression error). The other portions of the simulation are exactly the same, with additions again **bolded** below:

```
set.seed(131051)                # Set random seed for reproducibility
c.l_gamma =
  specify(                        # Specify data-generating process
    X =                           # Generate predictor matrix
      c(x1, x2) ~                 # Name predictors x1 and x2
      MASS::mvrnorm(             # Generate as multivariate normal
        n = N,                  # - Sample size is 'N'
        mu = c(0, 0),           # - Mean 0 for both predictors
        Sigma = p$cov),         # Covariance matrix (specified above)
    y =                           # Simulate outcome: sum of
      ~ p$b1 * x1 +              # Slope times x1
      p$b2 * x2 +               # Slope times x2
      p$b3 * x1 * x2 +          # Slope times interaction
      rgamma(n = N,              # Random gamma error
        shape = 1,            # shape parameter
        scale = p$sigma       # scale is sigma
        ) - p$sigma) %>%      # adjust mean error to 0
  define(                        # Define varying parameters, here 'N'
    N = seq(500,                # Minimum sample size examined
            1500,               # Maximum sample size examined
            by = 25)            # Interval of sample sizes attempted
  ) %>%
  fit(lm =~ lm(y ~ x1*x2)) %>%  # Fit linear interaction model
  tidy_fits() %>%              # Run broom::tidy on each model
  generate(10000)              # Generate 10,000 replications
```

The flexibility to change the data-generation process in just a few lines demonstrates the advantage of the custom simulation approach for relaxing assumptions beyond what was intentionally designed by authors of black box simulation functions such as `interaction_power()`.

Interestingly, the gamma error distribution implemented here seems to entirely mitigate the detrimental effect of the range restriction, with again a minimum sample size of approximately 1100 required to reach a power of .80. However, in a supplementary analysis not presented here in detail, we found that using this gamma error distribution *without* the range restriction resulted in similar power to the normal error distribution (see Figure 2). The exact mechanism is unclear, but we believe that the gamma error distribution, even with the same mean and standard deviation, resulted in less frequently generating values beyond the edges of the range. This interesting phenomenon shows the value of using custom simulation to explore scenarios that are beyond what analytic and black box simulations are programmed to handle.

Figure 2. Comparison of power curves for study 2 in Sillence et al. (2021), from simplest to more complex. See text for details.



Comparison of Power Analysis Curves

Now that the full approach has been demonstrated, we next compare the power curves to show how the incremental approach allows researchers to see the relative contributions to the final power analysis. A comparison of the power analysis curves is shown in Figure 2.

Our approach addresses several limitations in the power analysis in Sillence et al. (2021), noted above, including inflated effect sizes and uncorrelated predictors. All of the power analyses suggest that the actual sample sizes of 216 in the original Facebook usage study (McCord et al., 2014) and of 206 and 542 for the replication studies (Sillence et al., 2021) are likely well below the power to detect an effect of size $\Delta R^2 = 0.01$. Moreover, it is clear that increasingly sophisticated power analyses that relax additional assumptions greatly decrease power, with the effect of measurement error substantially decreasing power across models. These differences between power estimates demonstrate the sensitivity of power to varying parameters, including the correlation between predictors that was not accounted for in Sillence et al. (2021).

Finally, although the non-normal errors and range restrictions were primarily added to demonstrate the flexibility of the custom simulation approach, these two extensions appeared to interact in a complex way. The non-normal error, in this case highly skewed right, had a negligible effect on statistical power on its own. In contrast, adding a range restriction severely restricted power, consistent with the theoretical findings described in the literature review. Yet combining the non-normal error with the range restriction in this case produced power estimates very similar to having no range restriction at all, likely due to the distributional change leading to fewer truncated values.

Discussion and Conclusions

Interaction effects in educational and psychological studies are common but rarely detected (Morris et al., 1986), due likely to the lack of conducting an appropriate a priori power analysis. This could be partially attributed to the paucity of well-developed step-by-step tutorials presented in nontechnical language easily understood and implemented by applied researchers and practitioners. In the current article, we proposed an incremental model-building framework for conducting power analysis for MMR to ensure detecting the interaction effects and hence draw valid conclusions from hypothesis tests.

The proposed approach starts with examining whether there are applicable rules of thumb available from the literature. Then, stage A.1 uses *InteractionPoweR* to generate analytic estimates of the power of the regression coefficient. If variables are measured with error, stage A.2 again uses *InteractionPoweR* to account for these issues. Stage B.0 moves to using black box simulations, which use prepackaged simulation code such as that provided by *InteractionPoweR*, but we advocate initially simply replicating the analytic analysis before relaxing additional assumptions. This clears the way for stage B.1, if applicable, where any available extensions to the model that are supported by the black box simulation can be included. If more extensions or clarity are required, applied researchers and practitioners can then reimplement the simulation using the provided code in *simpr* in stage C.0, which allows for nearly any assumption to be relaxed or changed in stage C.1. Throughout the process, researchers can carefully track the changes in power estimates and the plausibility of models, much like the process of building a complex linear regression model with many covariates.

To increase the practical accessibility and utility of the proposed incremental approach, we demonstrated these six stages with an applied example and an easy-to-implement tutorial using R. Our demonstration shows the feasibility of the approach, as well as the advantages of comparing multiple power analysis models to clarify the nature of the context and reveal unexpected complexities in differing methods. Thus, this proposed framework has practical implications for applied researchers and practitioners, among others. Unlike tutorials where authors have advocated for building a single simulation model when conducting power analysis for regression models (e.g., Beaujean, 2014), the utility of the incremental approach lies in its model-building framework, where one would start with the simplest and increase complexity as needed. Second, it does not require advanced statistical knowledge or coding expertise to begin the model-building process—interested users can proceed up to the level of their expertise.

A remaining challenge for power analysis of MMR is the lack of software support to more easily implement common extensions to applied power analysis for MMR. These extensions include examination of centering issues (e.g. Afshartous & Preston, 2017) and non-normal predictors. Future analytic and black box simulation functions should support a wider range of extensions, especially non-normal predictor variables (e.g., Shieh, 2009). Although custom simulations are indeed flexible and can theoretically accommodate these or any other change to the data-generating mechanism, additional helper functions to simplify the process would make this functionality more widely accessible to R users. This is one of the goals of our ongoing work with the *simpr.interaction* package (Brown, 2025).

Moreover, we focused this tutorial on the R^2 change approach to evaluating power because this measure of effect size is most well-implemented in the R ecosystem, but other effect size measures such as the one by Liu and Yuan (2021) has potential advantages for practitioners. In the presence of measurement error, MMR is likely not the best approach and a latent variable approach may be more powerful (e.g., Su et al., 2019). We also focused on a traditional power analysis, rather than other possible outcomes such as the precision of confidence intervals for interaction effects (see Shieh, 2010). Our model-building approach would require some alterations to accommodate these other endpoints, and future research and software development should examine how to generalize our approach to these other situations.

To conclude, we believe that implementing and reporting on the proposed approach, along with continued software development on extensions, will increase confidence in the inferences where interaction effects are estimated as well as improving prospects for replication.

Received: 6/26/2024. **Accepted:** 5/23/2025. **Published:** 6/3/2025.

Citation: Brown, E. C., & Abulela, M. A. A. (2025). Power analysis for moderated multiple regression: An incremental model-building approach using R. *Practical Assessment, Research, & Evaluation*, 30(5). Available online: <https://doi.org/10.7275/pare.2233>

Corresponding Author: Ethan C. Brown, Fulbright University Vietnam. Email: ethan.brown@fulbright.edu.vn.

References

- Aberson, C. L. (2019). *Applied power analysis for the behavioral sciences* (2nd ed.). Routledge.
- Aberson, C. (2022). *pwr2ppl*: Power analysis for common designs (R package version 0.5.0). <https://cran.r-project.org/web/packages/pwr2ppl/index.html>
- Afshartous, D., & Preston, R. A. (2011). Key results of interaction models with centering. *Journal of Statistics Education*, 19(3). <https://doi.org/10.1080/10691898.2011.11889620>
- Aguinis, H. (1995). Statistical power problems with moderated multiple regression in management research. *Journal of Management*, 21(6), 1141–1158. [https://doi.org/10.1016/0149-2063\(95\)90026-8](https://doi.org/10.1016/0149-2063(95)90026-8)
- Aguinis, H. (2004). *Regression analysis for categorical moderators*. The Guilford Press.
- Aguinis, H., Beaty, J. C., Boik, R. J., & Pierce, C. A. (2005). Effect size and power in assessing moderating effects of categorical variables using multiple regression: A 30-year review. *Journal of Applied Psychology*, 90(1), 94–107. <https://doi.org/10.1037/0021-9010.90.1.94>
- Aguinis, H., & Stone-Romero, E. F. (1997). Methodological artifacts in moderated multiple regression and their effects on statistical power. *Journal of Applied Psychology*, 82(1), 192–206. <https://doi.org/10.1037/0021-9010.82.1.192>
- Aguinis, H., & Gottfredson, R. K. (2010). Best-practice recommendations for estimating interaction effects using moderated multiple regression. *Journal of Organizational Behavior*, 31(6), 776–786. <https://doi.org/10.1002/job.686>
- Albers, M. J. (2017). *Introduction to quantitative data analysis in the behavioral and social sciences*. John Wiley & Sons, Inc.
- Anderson, S. F. (2019). Best (but oft forgotten) practices: Sample size planning for powerful studies. *The American Journal of Clinical Nutrition*, 110(2), 280–295. <https://doi.org/10.1093/ajcn/nqz058>
- Bache, S. M., & Wickham, H. (2022). *magrittr: A Forward-Pipe Operator for R*. <https://CRAN.R-project.org/package=magrittr>
- Baranger, D. A., Finsaas, M., Goldstein, B., Vize, C., Lynam, D., & Olino, T. (2023). Tutorial: Power analyses for interaction effects in cross-sectional regressions. *Advances in Methods and Practices in Psychological Science*, 6(3), 1–13. <https://doi.org/10.31234/osf.io/5ptd7>
- Beaujean, A. A. (2014). Sample size determination for regression models using Monte Carlo methods in R. *Practical Assessment, Research, and Evaluation*, 19(12). <https://openpublishing.library.umass.edu/pare/article/1361/galley/1312/view/>
- Bengtsson, H. (2021). A unifying framework for parallel and distributed processing in R using futures. *The R Journal*, 13(2), 208–227. <https://doi.org/10.32614/RJ-2021-048>

- Beckstead, J. W. (2012). Isolating and examining sources of suppression and multicollinearity in multiple linear regression. *Multivariate Behavioral Research*, 47(2), 224-246.
<https://doi.org/10.1080/00273171.2012.658331>
- Brown, E. C. (2023). *simpr*: Flexible 'tidyverse'-friendly simulations (R package version 0.2.6).
<https://cran.r-project.org/package=simpr>
- Brown, E. C. (2025). *simpr.interaction*: Calculating parameters for simulation-based regression interaction power analysis (R package version 0.1.0). <https://cran.r-project.org/package=simpr.interaction>
- Brysbaert, M. (2019). How many participants do we have to include in properly powered experiments? A tutorial of power analysis with reference tables. *Journal of Cognition*, 2(1), 16. <https://doi.org/doi:10.5334/joc.72>
- Faul, F., Erdfelder, E., Lang, A.-G., & Buchner, A. (2007). G*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior Research Methods*, 39(2), 175–191.
<https://doi.org/10.3758/BF03193146>
- Field, A. (2013). *Discovering statistics using SPSS* (3rd ed.). Sage.
- Friedman, L., & Wall, M. (2005). Graphical views of suppression and multicollinearity in multiple linear regression. *The American Statistician*, 59(2), 127–136. <https://doi.org/10.1198/000313005X41337>
- Hoening, J. M., & Heisey, D. M. (2001). The abuse of power: The pervasive fallacy of power calculations for data analysis. *The American Statistician*, 55(1), 19–24. <https://doi.org/10.1198/000313001300339897>
- Howell, D. C. (2011). *Fundamental statistics for the behavioral sciences* (7th ed.). Wadsworth.
- Jaccard, J., Wan, C. K., & Turrissi, R. (1990). The detection and interpretation of interaction effects between continuous variables in multiple regression. *Multivariate Behavioral Research*, 25(4), 467-478.
https://doi.org/10.1207/s15327906mbr2504_4
- Lewis, J. W., & Escobar, L. A. (1986). Suppression and enhancement in bivariate regression. *The Statistician*, 35(1), 17-26. <https://doi.org/10.2307/2988294>
- Lix, L. M., & Keselman, H. J. (2010). Analysis of variance: Repeated measures designs. In G. R. Hancock & R. O. Mueller (Eds.), *The reviewer's guide to quantitative methods in the social sciences* (pp. 15-27). Routledge.
- Liu, X. S. (2014). *Statistical power analysis for the social and behavioral sciences: Basic and advanced techniques*. Taylor & Francis.
- Liu, H., & Yuan, K. (2021). New measures of effect size in moderation analysis. *Psychological Methods*, 26(6), 680-700. <https://doi.org/10.1037/met0000371>
- McCord, B., Rodebaugh, T. L., & Levinson, C. A. (2014). Facebook: Social uses and anxiety. *Computers in Human Behavior*, 34, 23–27. <https://doi.org/10.1016/j.chb.2014.01.020>
- Morris, J. H., Sherman, J. D., & Mansfield, E. R. (1986). Failures to detect moderating effects with ordinary least squares-moderated multiple regression: Some reasons and a remedy. *Psychological Bulletin*, 99(2), 282–288. <https://doi.org/10.1037/0033-2909.99.2.282>
- Moshagen, M., & Bader, M. (2024). *semPower*: General power analysis for structural equation models. *Behavior Research Methods*, 56(4), 2901–2922. <https://doi.org/10.3758/s13428-023-02254-7>
- Murphy, K., & Myers, B. (2023). *Statistical power analysis: A simple and general model for traditional and modern hypothesis tests* (5th ed.). Routledge.
- O'Boyle, E., Banks, G. C., Carter, K., Walter, S., & Yuan, Z. (2019). A 20-year review of outcome reporting bias in moderated multiple regression. *Journal of Business and Psychology*, 34(1), 19–37.
<https://doi.org/10.1007/s10869-018-9539-8>
- Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science*, 349(6251), aac4716. <https://doi.org/10.1126/science.aac4716>
- Peterson, R. A., & Brown, S. P. (2005). On the use of beta coefficients in meta-analysis. *Journal of Applied Psychology*, 90(1), 175–181. <https://doi.org/10.1037/0021-9010.90.1.175>

- R Core Team. (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. <https://www.R-project.org/>
- Shieh, G. (2006). Suppression situations in multiple linear regression. *Educational and Psychological Measurement*, 66(3), 435–447. <https://doi.org/10.1177/0013164405278584>
- Shieh, G. (2009). Detecting interaction effects in moderated multiple regression with continuous variables: Power and sample size considerations. *Organizational Research Methods*, 12(3), 510–528. <https://doi.org/10.1177/1094428108320370>
- Shieh, G. (2010). Sample size determination for confidence intervals effects in moderated multiple regression with continuous predictor and moderator variables. *Behavior Research Methods*, 42(3), 824–833. <https://doi.org/10.3758/BRM.42.3.824>
- Sillence, E., Saxton, T. K., & Pollet, T. (2021). Facebook social use and anxiety: A replication attempt. *Human Communication & Technology*, 2(1), Article 1. <https://doi.org/10.17161/hct.v2i1.14847>
- Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, 22(11), 1359–1366. <https://doi.org/10.1177/0956797611417632>
- Su, R., Zhang, Q., Liu, Y., & Tay, L. (2019). Modeling congruence in organizational research with latent moderated structural equations. *Journal of Applied Psychology*, 104(11), 1404–1433. <https://doi.org/10.1037/apl0000411>
- Venables, W. N., & Ripley, B. D. (2002). *Modern Applied Statistics with S* (Fourth). Springer. <https://www.stats.ox.ac.uk/pub/MASS4/>
- Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *dplyr: A Grammar of Data Manipulation*. <https://CRAN.R-project.org/package=dplyr>