





## PUMP: Estimating Power, Minimum Detectable Effect Size, and Sample Size When Adjusting for Multiple Outcomes in Multi-Level Experiments

Kristen B. Hunter   
University of New  
South Wales

Luke Miratrix   
Harvard Graduate  
School of Education

Kristin Porter  
K.E. Porter  
Consulting LLC

---

### Abstract

For randomized controlled trials (RCTs) with a single intervention's impact being measured on multiple outcomes, researchers often apply a multiple testing procedure (such as Bonferroni or Benjamini-Hochberg) to adjust  $p$  values. Such an adjustment reduces the likelihood of spurious findings, but also changes the statistical power, sometimes substantially. A reduction in power means a reduction in the probability of detecting effects when they do exist. This consideration is frequently ignored in typical power analyses, as existing tools do not easily accommodate the use of multiple testing procedures. We introduce the **PUMP** (Power Under Multiplicity Project) R package as a tool for analysts to estimate statistical power, minimum detectable effect size, and sample size requirements for multi-level RCTs with multiple outcomes. **PUMP** uses a simulation-based approach to flexibly estimate power for a wide variety of experimental designs, number of outcomes, multiple testing procedures, and other user choices. By assuming linear mixed effects models, we can draw directly from the joint distribution of test statistics across outcomes and thus estimate power via simulation. One of **PUMP**'s main innovations is accommodating multiple outcomes, which are accounted for in two ways. First, power estimates from **PUMP** properly account for the adjustment in  $p$  values from applying a multiple testing procedure. Second, when considering multiple outcomes rather than a single outcome, different definitions of statistical power emerge. **PUMP** allows researchers to consider a variety of definitions of power in order to choose the most appropriate types of power for the goals of their study. The package supports a variety of commonly used frequentist multi-level RCT designs and linear mixed effects models. In addition to the main functionality of estimating power, minimum detectable effect size, and sample size requirements, the package allows the user to easily explore sensitivity of these quantities to changes in underlying assumptions.

*Keywords:* power, multiple testing, multi-level models, randomized controlled trials.

---

## 1. Introduction

The **PUMP** (Power Under Multiplicity Project) R package (R Core Team 2023) fills an important gap in open-source software tools for designing multi-level randomized controlled trials (RCTs) with adequate statistical power. With **PUMP**, researchers can estimate statistical power, minimum detectable effect size (MDES), and needed sample size for multi-level experimental designs. In a multi-level design, units are nested within hierarchical structures such as students nested within schools nested within school districts. The statistical power is calculated for estimating the impact of a single intervention on multiple outcomes. The package uses a frequentist framework of linear mixed effects regression models, which is currently the prevailing framework for estimating impacts from experiments in education and other social policy research.

Using education research as a motivating example, we introduce the **PUMP** package to allow for directly answering experimental design questions that take multiple outcomes and multiple testing procedures (MTPs) into account, such as:

- How many schools would I need in my study to detect a given effect on at least three of my five outcomes? (Assuming I have a fixed number of students per school.)
- What size effect can I reliably detect on each outcome, given a planned MTP across all my outcomes?
- How would the power to detect a given effect change if only half my outcomes truly experienced an impact?

We note, however, that the problem of power estimation for multi-level RCTs is not exclusive to the educational setting.

To our knowledge, none of the existing software tools for power calculations allow researchers to account for multiple hypothesis tests and the use of a MTP. MTPs adjust  $p$  values to reduce the likelihood of spurious findings when researchers are testing for effects on multiple outcomes.<sup>1</sup> This adjustment can result in a substantial change in statistical power, greatly reducing the probability of detecting effects when they do exist. Unfortunately, when designing studies, researchers who plan to test for effects on multiple outcomes and employ MTPs frequently ignore the power implications of the MTPs.

Also, as researchers change their focus from one outcome to multiple outcomes, multiple definitions of statistical power emerge (Chen, Luo, Liu, and Mehrotra 2011; Dudoit, Shaffer, and Boldrick 2003; Senn and Bretz 2007; Westfall, Tobias, and Wolfinger 2011). The **PUMP** package allows researchers to consider multiple definitions of power, selecting those most suited to the goals of their study. The definitions of power with multiple outcomes include:

- *Individual power*: The probability of detecting an effect of at least a particular size (specified by the researcher) for a given hypothesis test. Individual power corresponds to how power is defined when the focus is on a single outcome. In the case of multiple outcomes, individual power is the probability of detecting an effect *after* adjustment with a MTP. With multiple outcomes, the researcher may specify different sizes for each outcome.

---

<sup>1</sup>Alternatively, MTPs can decrease the critical values for rejecting hypothesis tests. For ease of presentation, this paper focuses on the approach of adjusting  $p$  values.

- *1-minimal power*: The probability of detecting effects of at least a particular size (which can vary by outcome) on at least one outcome out of all measured outcomes. Similarly, the researcher can consider *d-minimal power* for any  $d$  less than the number of outcomes, or fractional powers, such as 1/2-minimal power (the power to detect effects on at least 50% of the outcomes).
- *Complete power*: The power to detect effects of at least a particular size on *all* outcomes.

As noted in Porter (2018), the prevailing default in many studies – individual power – may or may not be the most appropriate type of power when multiple outcomes are being considered. If the researcher’s goal is to find statistically significant estimates of effects on *most or all* primary outcomes of interest, then their power may be much lower than anticipated. On the other hand, if the researcher’s goal is to find statistically significant estimates of effects on *at least one* or a small proportion of outcomes, their power may be much better than anticipated. Typically, 1-minimal power, even after applying a MTP, is higher than individual power for a hypothesis test on a single, pre-specified outcome. By not accounting for both the challenges and opportunities arising from multiple outcomes, a researcher may find they have wasted resources. They may have designed an underpowered study that cannot detect the desired effect sizes, or they may have designed an overpowered study that had a larger sample size than necessary.

The methods in the **PUMP** package build on those introduced in Porter (2018). This earlier paper focused on a single RCT design and model – a multisite RCT with the blocked randomization of individuals, in which effects are estimated using a model with block-specific intercepts and with the assumption of constant effects across all units. This earlier paper also did not produce software to assist researchers in implementing its methods. With this current paper and with the introduction of the **PUMP** package, we extend the methodology to eleven multi-level RCT designs and models. Also, while Porter (2018) focused on estimates of power, **PUMP** goes further to also estimate MDES and sample size requirements that take multiplicity adjustments into account.

**PUMP** extends functionality of the popular *PowerUp!* tool (which includes a R package called **PowerUpR**, a spreadsheet, and a R **shiny** application), which computes power or MDES for multi-level RCTs with a single outcome (Dong and Maynard 2013; Bulus, Dong, Kelcey, and Spybrook 2022). For a wide variety of RCT designs with a single outcome, researchers can take advantage of closed-form solutions and numerous power estimation tools. For example, in education and social policy research, see Dong and Maynard (2013); Hedges and Rhoads (2010); Spybrook, Bloom, Congdon, Hill, Martinez, and Raudenbush (2011). However, closed-form solutions are difficult or impossible to derive when a MTP is applied to a setting with multiple outcomes. Instead, we use a simulation-based approach to achieve estimates of power for multiple outcomes.

The package uses a frequentist framework of linear mixed effects regression models, which allows straightforward modeling of the multi-level nature of an experiment. The package only applies to estimating the impacts from experiments with a single, binary treatment. **PUMP** supports experiments with up to three levels: Level one is the lowest level, containing individuals or units (students), level two nests level one units into groups (schools), and level three nests level two units into groups (school districts). We consider a variety of models with different combinations of fixed and random effects within the linear mixed effects framework.<sup>2</sup>

<sup>2</sup>Other options include nonparametric or Bayesian methods, but these are less prevalent in applied research

We explain these models in detail in the technical appendix available in the supplementary file `v108i06-appendix.pdf`. Designed-based inference will often map onto these models. See Schochet (2016) for an overview of designed-based inference in RCT contexts. See Miratrix, Weiss, and Henderson (2021) for design-based inference for two-level multisite experiments and Schochet, Pashley, Miratrix, and Kautz (2021) for blocked, cluster randomized experiments.

We now define some essential terminology. We define the causal impact (our estimand) on a particular outcome  $m$  as the average treatment effect of the intervention across units at the highest level of our research design. We use the term “impact” and “effect” interchangeably. For example, for a three-level experiment, the causal impact is the average treatment effect across districts. We call the impact a grand mean, because it is a mean of means: first, we calculate the mean treatment effect for students within each district, and then we take the mean across these means. For a two-level experiment, we take the grand mean across schools, and for a one-level experiment, the causal impact is the average treatment effect across students.

The MDES is the smallest true effect size the study can detect with the desired statistical significance level, in standard deviation units. The term “effect size” generally refers to a standardized mean difference effect size (Bloom 2006), which is the difference in mean outcomes for the treatment group and the control group divided by an index standard deviation such as the standard deviation of the measured outcome for the entire control group or, in some cases, some larger reference population. Researchers often use effect sizes in order to compare outcomes with different scales. For studies with two or more levels, there are multiple possibilities for defining variation (Spybrook, Hedges, and Borenstein 2014). In a two-level design, “one might define the effect size in terms of a standard deviation based on the variance between level one units,  $\sigma_1^2$ , the variance between level two units,  $\sigma_2^2$ , or the total variance  $\sigma_1^2 + \sigma_2^2$ ” (Spybrook *et al.* 2014). We follow the final definition, using the total variance over all levels in the denominator of effect size. We do not include any treatment heterogeneity in the standard deviation used to calculate effect sizes, however. Using total variation, and not including treatment variation, means the unit of standardization will be preserved regardless of how units are blocked or organized, and regardless of the degree of impact heterogeneity. Total variation is also considered a more conservative choice, closest to mimicking the variation one would find in larger reference populations (see, e.g., Weiss, Bloom, Verbitsky-Savitz, Gupta, Vigil, and Cullinan 2017). Other effect size definitions can be accommodated via simply rescaling the desired effect sizes entered into or read off of the package by the ratio of the target metric to total variation.

The package includes three core functions:

- `pump_power()` for calculating power given an experimental design and assumed model and MDES.
- `pump_mdes()` for calculating MDES given a target power and sample sizes.
- `pump_sample()` for calculating the required sample size at a given level for achieving a given target power for a given MDES and sample sizes at other levels.

---

(Gelman, Hill, and Yajima 2012, 2007). One might also use generalized linear mixed models such as logistic regression; this would be a future extension.

For any of these core functions, the user begins with two main choices. First, the user chooses the combination of the design and model of the RCT. As previously noted, the **PUMP** package covers a range of multi-level designs and models that researchers typically use in practice, with up to three levels of hierarchy. Second, the user chooses the MTP to be applied. **PUMP** supports five common MTPs: Bonferroni, Holm, single-step and step-down versions of Westfall-Young, and Benjamini-Hochberg.

After these two main choices, the user must also make a variety of decisions about assumed parameters of the data generating distribution. The package allow users to easily explore power over a range of possible values of many of these parameters. This exploration encourages the user to determine the sensitivity of estimates to different assumptions. **PUMP** also visually displays results. These additional functions include:

- `pump_power_grid()`, `pump_mdcs_grid()`, and `pump_sample_grid()` for calculating the given output over a range of possible parameter values.
- `update()` to re-run an existing calculation with a small number of parameters updated.
- `plot()` on **PUMP**-generated objects to generate plots (including grid outputs).

The **PUMP** package is available on CRAN at <https://CRAN.R-project.org/package=PUMP>. The authors of the **PUMP** package have also created a web application built with R **shiny** (Chang *et al.* 2023). This web application calls the **PUMP** package and allows users to conduct calculations with a user-friendly interface, but it is less flexible than the package, with a focus on simpler scenarios (e.g., 10 or fewer outcomes). The app can be found at <https://public.mdrc.org/pump/>.

The remainder of this paper is organized as follows. In Section 2, we introduce Diplomas Now, an educational experiment, to be used as a running example throughout the paper. In Section 3, we provide a summary of the multiple testing problem. Also in Section 3, we present an overview of the statistical challenges introduced by multiple hypothesis testing and how MTPs protect against spurious impact findings. In Section 4, we introduce our methodology for estimating power when taking the use of MTPs into account. This section also briefly discusses our validation process. Section 5 discusses the various choices a user must make when using the package, including the designs and models, MTPs, and key design and model parameters. Section 6 provides a detailed presentation of the **PUMP** package with multiple examples of using the package’s functions to conduct calculations for our education RCT example. Section 7 is a brief conclusion.

## 2. Diplomas Now

We illustrate our package using an example of a published RCT that evaluated a secondary school model called Diplomas Now. The Diplomas Now model is designed to increase high school graduation rates and post-secondary readiness. Evaluators conducted a RCT comparing schools who implemented the model to business-as-usual. We refer to this example throughout this paper to illustrate key concepts and to illustrate the application of the **PUMP** package.

The Diplomas Now model, created by three national organizations, Talent Development, City Year, and Communities In Schools, targets underfunded urban middle and high schools with many students who are not performing well academically. The model aims to transform these schools to better support students who fall off the path to high school graduation. Diplomas Now, with MDRC as a partner, was one of the first validation grants awarded as part of the Investing in Innovation (i3) competition administered by the federal Department of Education.

We follow the general design of the Diplomas Now evaluation, conducted by MDRC. The initial evaluation included two cohorts of schools, with each cohort implementing for two years (2011–2013 for Cohort 1 and 2012–2014 for Cohort 2). The cohorts included 62 secondary schools (both middle and high schools) in 11 school districts that agreed to participate. These schools were grouped in randomization blocks defined by district, cohort, and school type, and then randomized within these blocks. We would therefore say this RCT contains three levels (student, school, and randomization block), with random assignment at level two. We note that for clarity, we generally refer to level three as the “district level” rather than “randomization block” level. Schools assigned to the active treatment group were given the Diplomas Now model, while the schools in the control treatment group continued their existing school programs or implemented other reform strategies of their choosing (Corrin, Sepanik, Rosen, and Shane 2016).

The evaluation focused on three categories of outcomes: Attendance, Behavior, and Course performance, called the “ABC’s”, with multiple measures for each category. In addition, the evaluation measured an overall ABC composite measure of whether a student is above given thresholds on all three categories. This grouping constitutes 12 total outcomes of interest. Evaluating each of the 12 outcomes independently would not be good practice, as the chance of a spurious finding would not be well controlled. The authors of the MDRC report pre-identified three of these outcomes as *primary* outcomes before the start of the study in order to reduce the problem of multiple testing. We, by contrast, use this example to illustrate what could be done if there was uncertainty as to which outcomes should be primary. In particular, we illustrate how to conduct a power analysis to plan a study where one uses multiple testing adjustment, rather than predesignation, to account for the multiple outcome problem.

There are different guidelines for how to adjust for groupings of multiple outcomes in education studies. For example, Schochet (2008) recommends organizing primary outcomes into domains, conducting tests on composite domain outcomes, and applying multiplicity corrections to composites across domains. The What Works Clearinghouse applies multiplicity corrections to findings within the same domain rather than across different domains. We do not provide recommendations for which guidelines to follow when investigating impacts on multiple outcomes. Rather, we address the fact that researchers across many fields are increasingly applying MTPs and therefore need to correctly estimate power, MDES and sample size requirements accounting for this choice. In our example, we elect to do a power analysis separately for each of the three outcome groups of the ABC outcomes to control familywise error rather than overall error. This strategy means we adjust for the number of outcomes within each group independently. For illustration purposes, we focus on one outcome group, attendance, which we will assume contains five separate outcomes.

### 3. Overview of multiple testing

Our motivating example illustrates that researchers are often interested in testing the effectiveness of a single intervention on multiple outcomes.<sup>3</sup> The resulting multiplicity of statistical hypothesis tests can lead to spurious findings of effects. Multiple testing procedures counteract this problem by adjusting  $p$  values for effect estimates; generally,  $p$  values are adjusted upward to require a higher burden of proof. When not using a MTP, the probability of finding false positives increases, sometimes dramatically, with the number of tests. When using a MTP, this probability is controlled. The error inflation in multiple testing means that we cannot draw reliable conclusions about the existence of effects above a specified size unless a MTP is properly applied. For more background on multiple testing, see the section “Overview of Multiple Testing” in [Porter \(2018\)](#).

The MTPs that are the focus of this paper have three key features that affect statistical power: (1) whether the MTP is a familywise procedure or a false discovery rate procedure; (2) whether the MTP is single-step or stepwise; and (3) whether the MTP takes the correlation between test statistics into account. Below we briefly explain each of these features of MTPs and provide discussion of the new parameter specifications induced by some of these features when estimating power.

*FWER and FDR.* Some MTPs control the familywise error rate (FWER), while others control the false discovery rate (FDR). The FWER is the type I error as a rate across the entire set or “family” of multiple hypothesis tests. The MTPs introduced by Bonferroni ([Dunn 1959, 1961](#)), Holm ([Holm 1979](#)), and Westfall and Young ([Westfall and Young 1993](#)) control the FWER. The FDR, introduced by Benjamini and Hochberg ([Benjamini and Hochberg 1995](#)), is the expected proportion of all rejected hypotheses that are erroneously rejected. The choice between a procedure that controls FWER or FDR will depend on the context. FWER is more stringent and may be preferred when even a single false positive could lead to the wrong conclusion. On the other hand, researchers may choose FDR control if they are willing to accept a few false positives. FDR control is often chosen when the total number of hypotheses is large. A side remark is that MTPs may provide either “weak control” or “strong control” of the error rate they target. For more information on weak and strong control, see [Appendix A](#).

*Single-step and stepwise approaches.* Every MTP can be categorized as either a “single-step” or “stepwise” procedure. Single-step procedures adjust each  $p$  value independently of the other  $p$  values. Stepwise procedures adjust  $p$  values in a sequential manner, adjusting based on the number of null hypotheses that have already been rejected in previous steps. Generally, stepwise procedures are less conservative and preserve more power than single-step approaches. The Bonferroni and Westfall-Young single-step procedures are single-step; the Holm, Benjamini-Hochberg, and Westfall-Young step-down procedures are stepwise procedures. Note that stepwise procedures may be “step-down” or “step-up,” referring to whether a procedure begins with the smallest  $p$  value, and thus the largest effect size (step-down), or the largest  $p$  value (step-up).

Due to the dependencies of adjustments in stepwise MTPs, a new assumption must be considered when estimating power under multiplicity: The proportion of outcomes on which there are truly impacts, or, equivalently, the number of false null hypotheses. Assuming effects

---

<sup>3</sup>Testing the effectiveness of an intervention for multiple subgroups, at multiple points in time, or across multiple treatment groups also results in a multiplicity of statistical hypotheses and can also lead to spurious findings of effects, but this is beyond the scope of this paper.

on all outcomes might seem reasonable, as hypotheses of effects often drive the selection of outcomes in the first place. But, if this assumption is incorrect, the probability of detecting effects can be substantially diminished. If we assume that some effects are truly null, we must change our notion of power for  $d$ -minimal and complete power. See Appendix A for more details.

*Correlation between test statistics.* MTPs vary on whether they directly take into account the correlation between test statistics. The Bonferroni, Holm, and Benjamin-Hochberg procedures do not take the correlation structure into account. When the test statistics are correlated, Bonferroni and Holm still provide strong FWER control, but they adjust  $p$  values more than is necessary in that case. In contrast, both the Westfall-Young MTPs directly incorporate the correlation structure into the adjustment procedure, because the joint distribution of the test statistics under the complete null hypothesis is generated using the observed data.

The correlation between test statistics is a parameter a researcher must specify in order to estimate power, MDES or sample size requirements when using a MTP. One challenge for an analyst is to translate an assumption about the correlation between outcomes, which is a more intuitive quantity, to the correlation between test statistics, which is the quantity which factors into power calculations. When fitting a separate regression model for the impact on each outcome, the  $\binom{M}{2}$  correlations between test statistics are equal to the pairwise correlations between the residuals in the  $M$  impact models (Porter 2018). As a rule of thumb, we use the assumed correlations between outcomes as a proxy for the correlations between test statistics, although there may be some discrepancy between these sets of correlations. We expect the discrepancy between the correlation between outcomes and test statistics to be generally small, and for the discrepancy to not substantially affect power. However, as a sensitivity check we recommend using the correlation checker tool built into the package (discussed in Section 6.5.5) to check the correlation between test statistics for a given design, model, and set of parameter values.

## 4. Estimating power, MDES and sample size

### 4.1. Power estimation approach

We take an innovative simulation-based approach to estimating power, as introduced in Porter (2018). This approach also forms the foundation for estimating MDES and sample size. For a single outcome, we can often use closed-form algebraic expressions, which are derived from the assumed model. However, with multiple outcomes, finding such expressions can be difficult, or even impossible. In cases where it is possible to find a closed-form expression, we would need to find expressions for every design and model, MTP, and definition of power. Importantly, we would *also* need to find new expressions for any possible number of outcomes, which quickly becomes an intractable problem. Furthermore, in some cases, such as permutation-based procedures like Westfall-Young approaches, a closed-form solution does not exist. To avoid these complexities, we rely on simulation. The approach outlined below can estimate power for any scenario.

If we were to rely on a *full* simulation approach, we could use the following method to estimate power. We introduce this full simulation approach to provide intuition, but use a simplified and less computationally intensive approach in the package, as discussed below.



1. *Simulate a data sample according to the joint alternative hypothesis.* First, we formulate what we will refer to as the *joint alternative hypothesis*, which is the set of outcomes we assume to have treatment effects above the desired sizes. We define  $ATE_m$  (average treatment effect) to be the treatment impact for outcome  $m$ , with  $M$  total outcomes. In the Diplomas Now model,  $ATE_m$  is the grand mean treatment effect across randomization blocks, and is defined in Equation 3. If we have  $M = 5$  outcomes, as in the Diplomas Now study, one possible joint alternative hypothesis is that all outcomes have effects above specified sizes:  $H_A : ATE_1 > 0.125, ATE_2 > 0.2, ATE_3 > 0.1, ATE_4 > 0.1, ATE_5 > 0.05$ . Another possible joint alternative hypothesis is one where only the first two outcomes have effects above the desired sizes:  $H_A : ATE_1 > 0.125, ATE_2 > 0.2, ATE_3 = ATE_4 = ATE_5 = 0$ . Once our joint alternative hypothesis is specified, we would generate simulated data under this hypothesis. To simulate data, we also need to specify additional parameters to permit data generation (see Section 5.2 and the technical appendix for further details). For example, for the Diplomas Now experiment, we would assume a specific data generating process to allow us to simulate synthetic students, schools, and randomization blocks, including covariates, outcomes, and treatment assignment. This process would involve specifying parameter values such as  $R^2$  values, the amount of outcome variation explained by covariates at each level, and then translating these parameter choices into data-generating parameters, such as the coefficient values for covariates in a linear model.
2. *Estimate impacts on the simulated data.* Given simulated data, we could fit  $M$  regression models (specified to match the experimental design and model assumptions). For the models supported by **PUMP**, the relevant functions would be `lm()`, `lmer()` from the **lme4** package (Bates, Mächler, Bolker, and Walker 2015), and `interacted_linear_estimators()` from the **blkvar** package.<sup>4</sup> From the model output we extract the test statistics  $t_m$  for the estimated impacts, one statistic for each outcome, along with estimated standard errors.
3. *Calculate unadjusted  $p$  values.* The test statistics and standard errors would in turn give raw (unadjusted)  $p$  values. We can either calculate these by hand, or use the  $p$  values routinely returned by regression functions. For Diplomas Now, for example, we could run a multi-level regression model of each attendance measure on treatment status and student and school covariates, and extract  $p$  values from the regression outputs.
4. *Repeat above steps (1 through 3) for a large number of iterations.* Denote the number of iterations `tnum`. Repeating steps 1-3 `tnum` times results in a matrix of unadjusted  $p$  values which we call **F**, and is of dimension `tnum`  $\times$   $M$ . One row corresponds to one set of simulated raw  $p$  values from regressions for the five attendance outcomes of interest for Diplomas Now.
5. *Adjust  $p$  values.* For each row, corresponding to one simulated data set, the  $M$  raw  $p$  values corresponding to the  $M$  hypothesis tests can be adjusted according to the desired multiple testing procedure. This process generates a new matrix **G** of adjusted  $p$  values (again of dimension `tnum`  $\times$   $M$ ). For Bonferroni, Holm, and Benjamini-Hochberg adjustments, we use the function `p.adjust` in R (found in the **stats** package). We developed our own functions for implementing adjustment using the Westfall-Young

---

<sup>4</sup>This package is currently under development on GitHub; see <https://github.com/lmiratrix/blkvar>

procedures. One row corresponds to one set of simulated *adjusted p* values for the five attendance outcomes of interest for Diplomas Now.

6. *Calculate hypothesis rejection indicators.* For any MTP, the matrix of adjusted  $p$  values  $\mathbf{G}$  can then be compared with a specified value of  $\alpha$  (the default is 0.05, but the value can be changed by the user). For each row, corresponding to one iteration of simulated data, we record whether or not the null hypothesis was rejected for each outcome. This process results in a new matrix  $\mathbf{H}$ , which contains hypothesis rejection indicators (still of dimension  $\mathbf{tnum} \times M$ ). Using  $\mathbf{H}$ , we can compute all definitions of power.
7. *Calculate power.* To compute the different definitions of power:
  - *Individual power* for outcome  $m$  is the proportion of the  $\mathbf{tnum}$  rows in which the null hypothesis  $m$  was rejected (the mean of column  $m$  of  $\mathbf{H}$ ). We would have five different individual power values for Diplomas Now, corresponding to each outcome of interest.
  - *$d$ -minimal power* is the proportion of the  $\mathbf{tnum}$  rows in which at least  $d$  of the  $M$  hypotheses were rejected.<sup>5</sup> For Diplomas Now, we could consider 1-minimal power through 4-minimal power.
  - *Complete power* is the proportion of the  $\mathbf{tnum}$  rows in which all of the null hypotheses were rejected based on the raw  $p$  values rather than adjusted  $p$  values (based on the matrix  $\mathbf{G}$  rather than  $\mathbf{H}$ .) See Appendix A for an explanation of why we calculate complete power using the raw  $p$  values. We would be interested in complete power if we want to evaluate whether Diplomas Now resulted in improvement for every single attendance outcome of interest. With five outcomes, this criteria is a relatively strict indicator of success.

This full simulation approach for estimating power would be computationally intensive because of the need to generate and analyze a full simulated data set at each iteration. We can simplify this process by skipping the simulation of data and modeling steps. Given an assumed model and a specified correlation structure for the test statistics, we can directly sample from  $f(t_1, \dots, t_M)$ , the joint alternative distribution of the test statistics. This shortcut vastly improves both the simplicity and the speed of computation. In summary, our approach is:

1. *Generate draws of test statistics  $t_1, \dots, t_M$  under the joint alternative hypothesis.* This step produces a  $\mathbf{tnum} \times M$  matrix  $\mathbf{E}$ .
2. *Calculate unadjusted  $p$  values.* This produces the matrix  $\mathbf{F}$ , as in the procedure above.
3. *Adjust  $p$  values.* This produces the matrix  $\mathbf{G}$ , as in the procedure above.
4. *Calculate hypothesis rejection indicators.* This produces the matrix  $\mathbf{H}$ , as in the procedure above.
5. *Calculate power.*

---

<sup>5</sup>Note that others refer to 1-minimal power simply as “minimal power” (Maurer and Mellein 1988; Chen *et al.* 2011; Westfall *et al.* 2011), “disjunctive power” (Bretz, Hothorn, and Westfall 2010), or “any pair” power (Ramsey 1978). Chen *et al.* (2011) use the terminology of “ $r$ -power” for what is referred to here as  $d$ -minimal power for  $d > 1$ .

We now describe how to sample from  $f(t_1, \dots, t_M)$  directly. First, we assume a particular research design and model. In our example based on the Diplomas Now study, the research design is a three-level experiment, with randomization at level two. We plan for analyzing our data with a linear regression model with fixed intercepts at the randomization block level, random intercepts at the school level, and a constant treatment impact across schools and randomization blocks. Denote  $\text{ATE}_m$  as the treatment impact for outcome  $m$ .

We express treatment impacts in terms of effect sizes:

$$\text{ES}_m = \frac{\text{ATE}_m}{\sqrt{\text{VAR}_m}},$$

where  $\text{ATE}_m$  is the average treatment effect for outcome  $m$ , and  $\text{VAR}_m$  is the variance of the control outcome (the variance of the outcome across all units who receive no treatment). Remember that the definition of the ATE changes depending on the model: for three-level models it corresponds to the average district treatment effect, for two-level models the average school treatment effect, and for one-level models the average student treatment effect. Equation 4 shows the effect size formula for the Diplomas Now model. In order to calculate power, we also need the standard error of the impact in effect size units, which we denote as

$$Q_m = \text{SE}(\widehat{\text{ES}}_m).$$

The standard error  $Q_m$  is a consequence of the assumed model and a variety of parameters; our technical appendix shows formulae for  $Q_m$  for all the designs and models our package supports. In our Diplomas Now example,  $Q_m$  will be a function of the number of students, schools, and randomization blocks; the proportion of treated units; the number of student and school covariates; the explanatory power of the student and school covariates; the proportion of variation in the outcome explained by schools and randomization blocks; and the amount of impact variation relative to the amount of mean variation. These parameters will be discussed in more detail in Section 5.2. Some parameters, such as the proportion of units treated, will generally be known, while others, such as the  $R^2$  at different levels, would need to be supplied by the user through either estimation on pilot data or assumptions based on prior knowledge.

Given the effect sizes  $\text{ES}_m$  and the standard errors  $Q_m$ , we can determine the distribution of the vector of test statistics. When testing the hypothesis for outcome  $m$ , the test statistic for a  $t$  test is:

$$t_m = \frac{\widehat{\text{ES}}_m}{\widehat{Q}_m}.$$

Under the alternative hypothesis for outcome  $m$ ,  $t_m$  has a  $t$  distribution with degrees of freedom  $df$ , also determined by the model, and mean  $\text{ES}_m/Q_m$ . Finally, in addition to the parameters above, we also need to choose the correlation matrix between test statistics to sample from the joint distribution of  $f(t_1, \dots, t_M)$ . With these distributions specified, we can calculate  $p$  values.

This approach of simulating test statistics builds on work by Bang, Jung, and George (2005), who use simulated test statistics to identify critical values based on the distribution of the maximum test statistics. Their approach produces the same estimates as the approach described here for the single-step Westfall-Young MTP. As an alternative to a simulation-based approach, Chen *et al.* (2011) derived explicit formulae for  $d$ -minimal powers of stepwise procedures and for complete power of single-step procedures, but only for 1, 2, or 3 tests. The

approach presented here is more generally applicable, as it can be used for all MTPs, for any number of tests, and for all definitions of power discussed in the present paper.

*Remark.* The  $p$  value adjustment using Westfall-Young procedures is the most complex correction procedure, so we briefly outline it here. Similar to above, we first explain a full simulation approach, and then discuss our simplification. Under a full simulation approach, we would first generate a single data set under the joint alternative hypothesis and calculate a set of  $M$  observed test statistics. Then, we would permute the single simulated data set, say  $B = 3,000$  times, implicitly assuming the joint null hypothesis, and calculate test statistics on each of these permuted data sets. This process generates an empirical distribution of  $B$  test statistics under the joint null distribution. Next, we compare the distribution of observed test statistics to the generated distribution of test statistics under the joint null distribution to calculate  $p$  values. We would then re-generate a new simulated data set, and repeat the process. If we were to generate  $\mathbf{tnum} = 10,000$  data sets under the joint alternative hypothesis, for each of these data sets we generate  $B = 3,000$  permuted data sets under the joint null, so we would have to analyze  $10,000 \times 3,000$  data sets!

When we skip the step of simulating data, then for each iteration in  $1, \dots, \mathbf{tnum}$  we first generate a set of  $M$  observed test statistics from the joint alternative distribution. Then, we draw  $B$  samples of test statistics under the joint null rather than permuting the data  $B$  times. Under the null hypothesis,  $t_m$  has a  $t$  distribution with degrees of freedom  $df$  and mean zero. As before, we then compare the distribution of observed test statistics to the distribution of test statistics under the joint null distribution to calculate  $p$  values.

In summary, Westfall-Young procedures are computationally intensive, so the approach of skipping the simulated data step is particularly helpful here. Generating test statistics directly instead of generating and permuting data substantially reduces computational time.

## 4.2. Determining MDES and sample size

Frequently, a researcher’s main concern with power is calculating either the MDES for each outcome in a given study, or determining the necessary sample size to achieve a target power given a specified set of MDES values. In *Diplomas Now*, for example, we might want to know what sample sizes we would need to detect at least one significant effect across our outcomes if all the outcomes had a specified effect size (corresponding to 1-minimal power) and we were planning on using the Holm procedure.

For `pump_mdес()` and `pump_sample()`, the user provides a particular target power, say 80%. The method then solves a stochastic optimization problem to determine a value (of sample size or MDES) that is within a specified tolerance of the target power with high probability. We discuss the algorithm for MDES, although the approach for determining sample size is the same.

The algorithm first determines an initial range of MDES values that likely contain the target MDES. This initial range is calculated using formulae for unadjusted power based on the standard errors and degrees of freedom. In particular, from [Dong and Maynard \(2013\)](#), in general the MDES for a single outcome can be estimated as

$$\text{MDES} = \text{MT}_{df} \times \text{SE}/\sigma_m,$$

where  $\text{MT}_{df}$ , the “multiplier,” is the sum of two  $t$  statistics with degrees of freedom  $df$ . For one-tailed tests,  $\text{MT}_{df} = t_{\alpha}^* + t_{1-\beta}^*$  where  $\alpha$  is the type I error rate and  $\beta$  is the desired power.

For two-tailed tests,  $MT_{df} = t_{\alpha/2}^* + t_{1-\beta}^*$ . We do not explain the details of the derivations of the multiplier here; for more details and understanding, see [Dong and Maynard \(2013\)](#) or [Bloom \(2006\)](#). These expressions can be further manipulated to obtain sample size formulae; see our technical appendix for all formulae used in the package.

We can calculate our initial range by manipulating the  $\alpha$  and  $\beta$  values in the above. First, to calculate the preliminary lower bound, we apply the formula above as given, because individual unadjusted power is generally the least conservative adjustment and thus will give the smallest possible MDES. To calculate the preliminary upper bound, we apply the formula using  $\alpha/M$  to correspond to a Bonferroni correction, which is generally the most conservative adjustment and thus will give the largest possible MDES.

We also adjust  $\beta$  to account for different power types. For example, if we are interested in complete power, we need a larger upper bound than for individual power: In order to have a complete power of 80%, we would need each outcome to have an individual power of  $0.8^{(1/M)}$ , assuming independence. If we are interested in minimal power, we must have a smaller lower bound: In order to have 1-minimal power of 80%, each outcome would only need to have individual power of  $1 - (1 - 0.8)^{(1/M)}$ . We ignore correlation in the setting of the initial bounds because the bounds do not need to be strict, given the adaptive nature of the subsequent search.

Once the initial range is established, we use `pump_power()` with the complete array of design parameters to obtain rough (using a small `tnum`, or number of simulation trials) estimates of power for five initial values across this range. We then fit a scaled logistic curve to these five points, and identify where the curve crosses the desired power level. After fitting an initial curve, we iterate, repeatedly calculating power for the targeted point and using the result to update the logistic curve model. At any point, if the current fitted curve's range does not contain the target power, the algorithm extrapolates for the next step. With each iteration we increase `tnum` to increase precision as we narrow in on the final answer, and with each update to our estimated power curve, we weigh the collection of observations by their precisions (determined by corresponding `tnum` value). Once a test point achieves an estimated target power to within tolerance, we conduct an additional simulation check using a high number of replicates to verify the proposed answer is within a specified tolerance of the target power; if it is not, we continue the iterative search. The default tolerance is 1%, so given a target power of 80%, we stop when we find a MDES that gives an estimated power between 79% and 81%.

In practice, due to the monotonic nature of the logistic functional form, our algorithm generally converges fairly rapidly. However, in certain edge cases the algorithm may fail to converge on a value within tolerance. For more information on applying the search algorithm, see the sample size vignette in the package documentation.

### 4.3. Package validation

We completed extensive validation checks to ensure our power calculation procedures are correct. First, we compared our power estimates in scenarios with only one outcome,  $M = 1$ , to those from the **PowerUpR** package. Without a multiple testing procedure adjustment, our estimates match. Second, in order to validate our estimates under multiplicity, we followed the full simulation approach outlined above, in Section 4.1. The simulation approach involves generating many iterations of full data sets according to the assumed design and model, calcu-

lating  $p$  values, and calculating an empirical estimate of power. Using a binomial distribution we constructed Monte Carlo confidence intervals for the true power from the full simulation approach. Then, we validated that the **PUMP** estimates fall within these confidence intervals. A more detailed explanation of the validation procedure can be found in Appendix B, and full validation code and results are in a supplementary GitHub repository [https://github.com/MDRCNY/pump\\_validate](https://github.com/MDRCNY/pump_validate). For some scenarios, we have some apparent discrepancies from **PowerUpR**, but these result from different modeling choices. For example, for certain models *PowerUp!* assumes the intraclass correlation is zero, while we allow for nonzero values. In the technical appendix, we note any different choices between our approach and *PowerUp!*'s approach for each model.

## 5. User choices

In this section, we outline the choices a user must make when calculating power, MDES, or sample size.

### 5.1. Designs and models

When planning a study, the researcher first has to identify the design of the experiment, including the number of levels, and the level at which randomization occurs. These decisions can be a mix of the realities of the context (e.g., the treatment must be applied at the school level, and students are naturally nested in schools, making for a cluster randomization), or deliberate (e.g., the researcher groups similar schools to block their experiment in an attempt to improve power). Second, based on the design and the inferential goals of the study, the researchers choose an assumed model, including whether intercepts and treatment impacts should be treated as constant, fixed, or random. For the same experimental design, the analyst can sometimes choose from a variety of possible models, and these two decisions should be kept conceptually separated from each other.

*The design.* The **PUMP** package supports designs with one, two, or three levels, with randomization occurring at any level. For example, a design with two levels and randomization at level one is a blocked design (or equivalently a multisite experiment), where level two forms the blocks (blocks being groups of units, some of which are treated and some not). Ideally, the blocks in a trial will be groups of relatively homogeneous units, but frequently they are a consequence of the units being studied (e.g., evaluations of college supports, with students, the units, nested in colleges, the blocks). A design with two levels and randomization at level two is commonly called a cluster design (e.g., a collection of schools, with treatment applied to a subset of the schools, with outcomes at the student level); here the schools are the clusters, with a cluster being a collection of units which is entirely treated or entirely not. We can also have both blocking and clustering: Randomizing schools within districts, creating a series of cluster-randomized experiments, would be a blocked (by district), cluster-randomized experiment, with randomization at level two.

*The model.* Given a design, the researcher can select a model via a few modeling choices. In particular the researcher has to decide, for each level beyond the first, about the intercepts and the estimated treatment impacts:

- Whether level two and level three intercepts are:

- Fixed (**f**): We have a separate intercept for each unit.
- Random (**r**): We have a separate intercept for each unit as above, but model the collection of intercepts as Normally distributed, allowing for partial pooling.
- Whether level two and level three treatment impacts are:
  - Constant (**c**): We model all units within a group as having the same average impact.
  - Fixed (**f**): We allow each block or cluster within a level to have its own individual estimated impact (we can only do this if we have treated and control units within said block or cluster).
  - Random (**r**): We allow variation as with fixed, but model the collection of treatment impacts as Normally distributed around a grand mean impact. Modeling treatment impact as a random effect implicitly targets a superpopulation context where the sites are themselves a sample from a larger population. Modeling impacts as fixed or constant, on the other hand, targets a finite population context. Thus, depending on the inferential goal of the analyst (whether they want their estimate to only apply to the units at hand, or whether they want to generalize to a larger population), they might prefer one model over another. See [Miratrix et al. \(2021\)](#) for further discussion.

We denote the research design by **d**, followed by the number of levels and randomization level, so **d3.1** is a three-level design with randomization at level one. The model is denoted by **m**, followed by the level and the assumption for the intercepts (**f** or **r**), and then the assumption for the treatment impacts (**c**, **f**, or **r**). For example, **m3ff2rc** means at level three, we assume fixed intercepts and fixed treatment impacts, and at level two we assume random intercepts and constant treatment impacts. The full design and model are specified by concatenating these together, e.g., **d3.1\_m3ff2rc**. The Diplomas Now model, for example, is **d3.2\_m3fc2rc**, as we explain below.

The full list of supported design and model combinations is shown in Table 1. The user can see this information in the package by calling `pump_info()`, which provides the designs and models, MTPs, power definitions, and model parameters. Calling `pump_info()` with `comment = TRUE` provides additional detail; this additional output is shown in Table 2. We also include the corresponding names from *PowerUp!* where appropriate. For more details about each combination of design and model, see the technical appendix.

### *Diplomas Now*

We walk through the design and model for the Diplomas Now example.

*The design.* As noted above, the RCT contains three levels with random assignment at level two. In our notation, this is a **d3.2** design. The MDRC researchers conducted randomization of the schools within blocks defined by district, school type, and year of roll-out. After some schools were dropped from the study due to structural reasons, the researchers were left with 29 high schools and 29 middle schools grouped in 21 random assignment blocks. Within each block, schools were randomized to the active treatment or business-as-usual, resulting in 32 schools in the treatment group, and 30 schools in the control group.

Code	Design	Model	<i>PowerUp!</i>	Parameters
d1.1_m1c	d1.1	m1c	-	R2.1
d2.1_m2fc	d2.1	m2fc	bira2_1c	R2.1, ICC.2
d2.1_m2ff	d2.1	m2ff	bira2_1f	R2.1, ICC.2
d2.1_m2fr	d2.1	m2fr	bira2_1r	R2.1, ICC.2, omega.2
d2.1_m2rr	d2.1	m2rr	-	R2.1, ICC.2, omega.2
d2.2_m2rc	d2.2	m2rc	cra2_2r	R2.1, R2.2, ICC.2
d3.1_m3rr2rr	d3.1	m3rr2rr	bira3_1r	R2.1, ICC.2, omega.2, ICC.3, omega.3
d3.2_m3ff2rc	d3.2	m3ff2rc	bcra3_2f	R2.1, R2.2, ICC.2, ICC.3
d3.2_m3fc2rc	d3.2	m3fc2rc	-	R2.1, R2.2, ICC.2, ICC.3
d3.2_m3rr2rc	d3.2	m3rr2rc	bcra3_2r	R2.1, R2.2, ICC.2, ICC.3, omega.3
d3.3_m3rc2rc	d3.3	m3rc2rc	cra3_3r	R2.1, R2.2, ICC.2, R2.3, ICC.3

Table 1: Supported designs and models: Summary.

Code	Description
d1.1_m1c	1 level, level 1 rand / constant impacts model
d2.1_m2fc	2 levels, level 1 rand / fixed intercepts, constant impacts
d2.1_m2ff	2 levels, level 1 rand / fixed intercepts, fixed impacts
d2.1_m2fr	2 levels, level 1 rand / fixed intercepts, random impacts (FIRC)
d2.1_m2rr	2 levels, level 1 rand / random intercepts & impacts (RIRC)
d2.2_m2rc	2 levels, level 2 rand / random intercepts, constant impacts
d3.1_m3rr2rr	3 levels, level 1 rand / level 3 random intercepts, random impacts, level 2 random intercepts, random impacts
d3.2_m3ff2rc	3 levels, level 2 rand / level 3 fixed intercepts, fixed impacts, level 2 random intercepts, constant impacts
d3.2_m3fc2rc	3 levels, level 2 rand / level 3 fixed intercepts, constant impact, level 2 random intercepts, constant impact
d3.2_m3rr2rc	3 levels, level 2 rand / level 3 random intercepts, random impacts, level 2 random intercepts, constant impacts
d3.3_m3rc2rc	3 levels, level 3 rand / level 3 random intercepts, constant impacts, level 2 random intercepts, constant impacts

Table 2: Supported designs and models: Model details.

*The model.* Given the design, we next need to specify how we will analyze our data; this choice also determines which design parameters we will need to specify. Following the original Diplomas Now report, we plan on using a multi-level model with fixed effects at level three, a random intercept at level two, and a single treatment coefficient. We represent this model as `m3fc2rc`. The `3fc` means we are including block fixed effects, and not modeling any treatment impact variation at level three. The `2rc` means random intercept and no modeled variation of treatment within each block (the `c` is for constant). The Diplomas Now report authors call their model a “two-level” model, but this is not quite aligned with the language of this package. In particular, fixed effects included at level two are actually accounting for variation at level three; we therefore identify their model as a three-level model with fixed effects at level three.



*Combined design and model.* Our final combination of design and model is `d3.2_m3fc2rc`. For full detailed information about this design and model, see Section 2.5.1 of the technical appendix.

*Equations.* The observed outcome for student  $i$  in school  $j$  in randomization block  $k$  for outcome  $m$  is  $Y_{ijkm}$ . Our assumed model is:

$$\begin{aligned} Y_{ijkm} &= \theta_{0,jkm} + \gamma_m C_{ijkm} + r_{ijkm} \\ \theta_{0,jkm} &= \psi_{0,km} + \Xi_{1,m} T_{jk} + \delta_m X_{jkm} + u_{0,jkm} \\ \psi_{0,km} &= \Xi_{0,m} + w_{0,km}, \end{aligned}$$

with distributions:

$$u_{0,jkm} \sim N\left(0, \tau_{0,m}^2\right).$$

To explain the model, we consider one line at a time. The first line describes how the observed outcomes are a function of school membership and individual-level factors. The second line describes how school-level average outcomes are a function of district membership and school-level factors (including treatment). The third line describes how districts vary.

- An individual's outcome  $Y_{ijkm}$  is a linear combination of a level one mean  $\theta_{0,jkm}$  under control treatment, a level one covariate  $C_{ijkm}$  and its coefficient  $\gamma_m$ , and a residual  $r_{ijkm}$ . We assume  $r_{ijkm} \sim N(0, \sigma_m^2)$ , although we do not model the individual residuals as random effects.
- The level one grand mean  $\theta_{0,jkm}$  can be decomposed into a level two grand mean  $\psi_{0,km}$  under control treatment, the level two treatment indicator  $T_{jk}$  and treatment impact  $\Xi_{1,m}$ , a level two covariate  $X_{jkm}$  and its coefficient  $\delta_m$ , and a random school-level intercept  $u_{0,jkm} \sim N(0, \tau_{0,m}^2)$ .
- The level two grand mean  $\psi_{0,km}$  can be decomposed into a level three grand mean  $\Xi_{0,m}$  plus a fixed third-level intercept  $w_{0,km}$ ; we have no level three covariate for this model due to the third-level fixed effects. We assume the fixed effects  $w_{0,km}$  have variance  $\eta_{0,m}^2$ .

Covariates can be at level one, two, or three. Level two covariates, for example, would not vary for individuals within a given level two group, but could be different for different level two groups. We assume only one covariate is observed per level without loss of generality; the case of multiple covariates can be reduced to the single covariate corresponding to the linear projection of the outcome on the full set of covariates. The only difference would be the degrees of freedom loss in estimation. We also assume all covariates are group mean centered within their next higher level groups, so only covariates at a particular level can explain variance at that level (see [Raudenbush and Bryk 2002](#), pp. 31–35, for further discussion). Centering can be done without loss of generality: If the mean of a level one covariate varies across level two units, for example, it can be split into the group-mean centered version at level one and an additional level two covariate of the group means. We finally assume covariates are standardized to unit variance; as we measure covariate influence by different  $R^2$  measures, this can be done without loss of generality as well.

Each unit has two potential outcomes:  $Y_{ijkm}(T_{ijk} = 1)$  is the potential outcome given the active treatment, and  $Y_{ijkm}(T_{ijk} = 0)$  is the potential outcome given the control treatment. We can then define average treatment effects for outcome  $m$  at different levels:

$$\text{Level 1: } \theta_{1,jkm} = \frac{1}{\bar{n}} \sum_{i=1}^{\bar{n}} [Y_{ijkm}(1) - Y_{ijkm}(0)] \quad \text{for a given school } j \text{ in district } k. \quad (1)$$

$$\text{Level 2: } \psi_{1,km} = \frac{1}{J} \sum_{j=1}^J \theta_{1,jkm} \quad \text{for district } k. \quad (2)$$

$$\text{Level 3: } \Xi_{1,m} = \frac{1}{K} \sum_{k=1}^K \psi_{1,km}. \quad (3)$$

We call the means at levels two and three “grand means” because they are means of means. We are interested in estimating  $\Xi_{1,m}$ , the grand mean treatment effect across randomization blocks. Here, we assume a constant treatment impact across schools and randomization blocks, so  $\theta_{1,jkm} = \psi_{1,km} = \Xi_{1,m}$ .

For calculating power, we are interested in the effect size, which is the difference in means divided by the total control variation:

$$\text{ES}_m = \frac{\text{ATE}_m}{\sqrt{\text{VAR}(Y_{ijkm}(0))}} = \frac{\Xi_{1,m}}{\sqrt{(\xi_m^2 + \gamma_m^2) + (\delta_m^2 + \eta_{0,m}^2) + (\tau_{0,m}^2 + \sigma_m^2)}}. \quad (4)$$

The reduced form model is:

$$Y_{ijkm} = \Xi_{0,m} + \Xi_{1,m}T_{ijk} + \delta_m X_{ijkm} + \gamma_m C_{ijkm} \\ + w_{0,km} + u_{0,jkm} + r_{ijkm}.$$

This reduced form leads us to the formula we would use if we were fitting this model in R using the **lme4** package.

```
R> lmer(Yobs ~ 0 + T.jk + X.jk + C.ijk + D.id + (1 | S.id), data = data)
```

In this formula, **S.id** is a categorical variable denoting the school membership of each unit, and **D.id** is the categorical variable denoting district membership. Each level of **D.id** will be converted to a fixed effect. Each level of **S.id** will be modeled as a random effect.

## 5.2. Model parameters

Table 3 shows the parameters that can affect  $Q_m$ , the standard error, for different designs and models. The first column shows the mathematical notation as introduced in this text and the technical appendix, the second column shows the name of the parameter in the **PUMP** package, and the third column contains a description of the parameter. Note that some of the parameters are scalars ( $J$ ), while others are allowed to be vectors ( $R_2^2$  can be a  $M$ -vector of the  $R^2$  values for level two for each outcome  $m$ ).

A few parameters warrant more explanation.

Notation	Parameter	Description
$\bar{n}$	nbar	Scalar; harmonic mean of number of level 1 units per level 2 unit (students per school)
$J$	J	Scalar; harmonic mean of number of level 2 units per level 3 unit (schools per district)
$K$	K	Scalar; number of level 3 units (districts)
$\bar{T}$	Tbar	Scalar; proportion of units assigned to treatment
$g_1$	numCovar.1	Scalar; number of level 1 (individual) covariates
$g_2$	numCovar.2	Scalar; number of level 2 (school) covariates
$g_3$	numCovar.3	Scalar; number of level 3 (district) covariates
$R_1^2$	R2.1	Scalar/vector; percent of variation explained by level 1 covariates
$R_2^2$	R2.2	Scalar/vector; percent of variation explained by level 2 covariates
$R_3^2$	R2.3	Scalar/vector; percent of variation explained by level 3 covariates
ICC <sub>2</sub>	ICC.2	Scalar/vector; level 2 intraclass correlation
ICC <sub>3</sub>	ICC.3	Scalar/vector; level 3 intraclass correlation
$\omega_2$	omega.2	Scalar/vector; ratio of variance of level 2 average impacts to level 2 random intercepts
$\omega_3$	omega.3	Scalar/vector; ratio of variance of level 3 average impacts to level 3 random intercepts
$\rho$	rho	Scalar; correlation between all pairs of test statistics

Table 3: PUMP model parameters.

- ICC is the unconditional intraclass correlation, and gives a measure of variation at a particular level of the model. The ICC for each level is defined as the ratio of the variance at that level divided by the overall variance of the individual outcomes. The ICC includes the variation due to covariates. It does not include variation due to treatment impact. ICC is defined on levels two and three. In the Diplomas Now model, these quantities are:

$$\text{ICC}_{3,m} = \frac{\text{VAR}(\psi_{0,km})}{\text{VAR}(Y_{ijkm}(0))} = \frac{\eta_{0,m}^2}{(\eta_{0,m}^2) + (\delta_m^2 + \tau_{0,m}^2) + (\gamma_m^2 + \sigma_m^2)} \quad (5)$$

$$\text{ICC}_{2,m} = \frac{\text{VAR}(\theta_{0,jkm} | \psi_{0,km})}{\text{VAR}(Y_{ijkm}(0))} = \frac{\delta_m^2 + \tau_{0,m}^2}{(\eta_{0,m}^2) + (\delta_m^2 + \tau_{0,m}^2) + (\gamma_m^2 + \sigma_m^2)}, \quad (6)$$

where the  $\eta_{0,m}^2$  term is the variance of the district-level fixed effects, or, in other words, the variance of the mean control-side outcome of the districts. Based on these definitions,  $\text{ICC}_{2,m} + \text{ICC}_{3,m} \leq 1$ .

- $\omega$  is the ratio between impact variation at a level and the variation in intercepts (including covariates) at that level. It is a measure of treatment impact heterogeneity.  $\omega$  is defined on levels two and three. In the Diplomas Now model, we assume  $\omega_{3,m} = 0$

and  $\omega_{2,m} = 0$ , i.e., there is no treatment impact variation. In general, these quantities are defined as:

$$\omega_{3,m} = \frac{\text{VAR}(\psi_{1,km})}{\text{VAR}(\psi_{0,km})}$$

$$\omega_{2,m} = \frac{\text{VAR}(\theta_{1,jkm} \mid \psi_{1,km})}{\text{VAR}(\theta_{0,jkm} \mid \psi_{0,km})}.$$

- $R^2$ , defined for each level, is the percent of variation at a level that can be predicted by covariates. Pilot data can be used to achieve reasonable estimates of  $R^2$ s in practice. For two-level data, an analyst can first split any lower level covariate into a lower level group-mean centered covariate and higher level group mean covariate before estimating the relevant level-specific  $R^2$ s; this will capture the total explanatory power of covariates for explaining variation at level one and level two. For three-level data, the level two group means (and any other level two covariate) can again be split by de-meaning within level three groups. In the Diplomas Now model, given our assumption of single, group-mean centered covariates at each level, the  $R^2$  quantities are as follows:

$$R_{2,m}^2 = 1 - \frac{\text{VAR}(u_{0,jkm})}{\text{VAR}(\theta_{0,jkm} \mid D_{id})} = 1 - \frac{\tau_{0,m}^2}{\delta_m^2 + \tau_{0,m}^2} \quad (7)$$

$$R_{1,m}^2 = 1 - \frac{\text{VAR}(r_{ijkm})}{\text{VAR}(Y_{ijkm}(0) \mid S_{id}, D_{id})} = 1 - \frac{\sigma_m^2}{\gamma_m^2 + \sigma_m^2}. \quad (8)$$

The denominator in  $R_{2,m}^2$  is the within-randomization block variation of the school mean outcomes under control treatment, so that we are only quantifying variation at level two. The denominator of  $R_{1,m}^2$  is the variation of student outcomes within a school, so that we are only quantifying variation at level one.

All of these expressions are defined for each outcome, so each outcome can have a different value. These expressions above have been adapted to the Diplomas Now setting, but for more general formulae for these expressions (that apply to all designs and models) see the technical appendix, which outlines the assumed data-generating process and the resulting expressions for ICC,  $\omega$ , and  $R^2$ .

In addition to design parameters, there are additional parameters that control the precision and speed of the power estimates themselves:

- `tnum` is the number of test statistics generated in order to estimate power. A larger number of test statistics results in greater computation time, but also a more precise estimate of power. The `pump_mdes()` and `pump_sample()` have additional `tnum` parameters to further control the precision of the search, if desired.
- `B` is the number of Westfall-Young permutations. Again, there is a trade-off between precision and computation time.
- `parallel.WY.cores` specifies the number of cores to use for parallel computation of the Westfall-Young step-down procedure, which is the most computationally intensive MTP. The default of 1 does not result in parallel computation. Parallelization is done using `parApply` from the `parallel` package.

Procedure	Control	Single-step or stepwise	Accounts for correlation
Bonferroni (BF)	FWER	Single-step	No
Holm (HO)	FWER	Stepwise	No
Westfall-Young single-step (WY-SS)	FWER	Single-step	Yes
Westfall-Young step-down (WY-SD)	FWER	Stepwise	Yes
Benjamini-Hochberg (BH)	FDR	Stepwise	No

Table 4: Summary of MTP procedures.

### 5.3. Multiple testing procedures

Here we provide a review of the multiple testing procedures supported by the **PUMP** package:

- *Bonferroni* (BF): Adjusts  $p$  values by multiplying them by  $M$  to ensure strong control of the FWER. Bonferroni is a simple procedure, but the most conservative.
- *Holm* (HO): A step-down version of Bonferroni. Starting from smallest to largest,  $p$  values are sequentially adjusted by different multipliers. Holm is less conservative than Bonferroni for larger  $p$  values.
- *Benjamini-Hochberg* (BH): A sequential, step-up procedure that controls the FDR. Using the BH method, only null hypotheses with  $p$  values below a certain threshold are rejected, where the threshold is determined by the number of tests and the level  $\alpha$ .
- *Single-step Westfall-Young* (WY-SS): A permutation-based procedure for controlling the FWER, which directly takes into account the joint correlation structure of the outcomes. In the single-step approach, all outcomes are adjusted by using the permuted distribution of the minimum  $p$  value. Although Westfall-Young procedures are less conservative while still protecting against false discoveries, they are computationally very intensive.
- *Step-down Westfall-Young* (WY-SD): A similar approach to the single-step procedure, except that outcomes are adjusted sequentially from smallest to largest according to the permuted distributions of the corresponding sequential  $p$  values.

Table 4 from [Porter \(2018\)](#) summarizes the important features for each of the MTPs supported by **PUMP**. For a more detailed explanation of each MTP, see Appendix A of [Porter \(2018\)](#).

## 6. Using the PUMP package

In this section, we illustrate how to use the **PUMP** package, using our example motivated by the Diplomas Now study. Given the study’s design, we ask a natural initial question: What size of impact could we reasonably detect after using a MTP to adjust  $p$  values to account for our multiple outcomes?

We mimic the planning process one might use for planning a study similar to Diplomas Now (e.g., if we were planning a replication trial in a slightly different context). To answer this

question we therefore first have to decide on our experimental design and modeling approach. We also have to determine values for the associated design parameters that accompany these choices. In the following sections, we walk through selecting these parameters (sample size, control variables, intraclass correlation coefficients, impact variation, and correlation of outcomes). We calculate MDES for the resulting context and determine how necessary sample sizes change depending on what kind of power we desire. We finally illustrate some sensitivity checks, looking at how MDES changes as a function of ICC,  $\rho$ , and the number of assumed null outcomes.

### 6.1. Establishing needed design parameters

To conduct power, MDES, and sample size calculations, we first specify the design, sample sizes, analytic model, and level of statistical significance. We also must specify parameters of the data generating distribution that match the selected design and model. All of these numbers have to be determined given resource limitations, or estimated using prior knowledge, pilot studies, or other sources of information. For further discussion of selecting these parameters see, for example Bloom (2006) and Dong and Maynard (2013). For discussion in the multiple testing context, especially with regards to the overall power measures such as 1-minimal or complete power, see Porter (2018); the findings there are general, as they are a function of the final distribution of test statistics. The key insight is that power is a function of only three summarizing elements: The individual-level standard errors, the degrees of freedom, and the correlation structure of the test statistics. Once we calculate these three elements using the overall design and various design parameters, we can directly simulate the  $t$  statistics and calculate power.

*Sample sizes.* We assume equal size randomization blocks and schools, as is typical of most power analysis packages. For our context, this gives about three schools per randomization block; we can later do a sensitivity check where we increase and decrease this number to see how power changes. The Diplomas Now report states there were 14,950 students, yielding around 258 students per school. Normally we would use the geometric means of schools per randomization block and students per school as our design parameters, but that information is not available in the report. We assume 50% of the schools are treated; our calculations will be approximate here in that we could not actually treat exactly 50% in small and odd-sized blocks.

*Control variables.* We next need values for the  $R^2$  of the possible covariates. Remember our definition of  $R^2$  in Equations 7–8: The percent of variation at a level predicted by covariates specific to that level. The report does not provide these quantities, but it does mention covariate adjustment in the presentation of the model. Given the types of outcomes we are working with, it is unlikely that there are highly predictive individual-level covariates, but our prior year school-average attendance measures are likely to be highly predictive of corresponding school-average outcomes. We thus set  $R_1^2 = 0.1$  and  $R_2^2 = 0.5$ . We assume five covariates at level one and three at level two; this decision, especially for level one, usually does not matter much in practice, unless sample sizes are very small (the number of covariates along with sample size determine the degrees of freedom for our planned tests).

To provide some additional intuition,  $R_1^2 = 0.1$  means that:

$$0.1 = 1 - \frac{\text{VAR}(r_{ijkm})}{\text{VAR}(Y_{ijkm}(0) | S_{id}, D_{id})},$$

so

$$\frac{\sigma_m^2}{\gamma_m^2 + \sigma_m^2} = 0.9.$$

This means the variation in residuals  $\sigma_m^2$  describes 90% of the level one variation (excluding variation due to treatment impact).

*ICCs.* We also need a measure of where variation occurs: The individual, the school, or the randomization block level. As explained earlier, we capture this measure with ICCs, one for level two and one for level three. ICC measures specify overall variation in outcome across levels, e.g., do we see relatively homogeneous students within schools that are quite different, or are the schools generally the same with substantial variation within them? We typically would obtain ICCs from pilot data or external reports on similar data. We here specify a level two ICC of 0.05, and a level three ICC of 0.4. We set a relatively high level three ICC as we expect our school type by district randomization blocks to isolate variation; in particular we might believe middle and high school attendance rates would be markedly different.

Returning to Equation 5, for level three ICC we have

$$0.4 = \frac{\text{VAR}(\psi_{0,km})}{\text{VAR}(Y_{ijkm}(0))}.$$

This expression means that the variance of level two means (modeled using school-level fixed intercepts) explains 40% of the total variation (excluding variation due to treatment impact).

*Correlation of outcomes.* We finally need to specify the number of and relationship among our outcomes and associated test statistics. For illustration, we select attendance as our outcome group. We assume we have five different attendance measures. The main decision regarding outcomes is the correlation of our test statistics. As a rough proxy, we use the correlation of the outcomes at the level of randomization; in our case this would be the correlation of school-average attendance within each block. We believe the attendance measures would be fairly related, so we select  $\rho = 0.4$  for all pairs of outcomes. This value is an estimate, and we strongly encourage exploration of different values of this correlation choice as a sensitivity check for any conducted analysis. As a final step, we use the **PUMP** correlation checker to determine if an outcome correlation of 0.4 results in a test statistic correlation of 0.4, and check if any difference impacts our conclusions about power. Selecting a candidate  $\rho$  is difficult, and will be new for those only familiar with power analyses of single outcomes; we need to more research in the field, both empirical and theoretical, to further guide this choice.

If the information were available, we could specify different values for the design parameters such as the  $R^2$ s and ICCs for each outcome, if we thought they had different characteristics; for simplicity we do not do this here. The **PUMP** package also allows specifying different pairwise correlations between the test statistics of the different outcomes via a matrix of  $\rho$ s rather than a single  $\rho$ ; also for simplicity, we do not do that here.

Once we have established initial values for all needed parameters, we first conduct a baseline calculation, and then explore how MDES or other quantities change as these parameters change.

## 6.2. Calculating MDES

We now have an initial planned design, with a set number of schools and students. But is this a large enough experiment to reliably detect reasonably sized impacts? To answer this

MTP	Adjusted.MDES	D1indiv.power
HO	0.106	0.797

Table 5: MDES estimate.

question we calculate the minimum detectable effect size (MDES), given our planned analytic strategy, for our outcomes.

To identify the MDES of a given setting we use the `pump_mdes` method, which conducts a search for a MDES that achieves a target level of power. The MDES depends on all the design and model parameters discussed above, but also depends on the type of power and target level of power we are interested in. For example, we could determine what size effect we can reliably detect on our first outcome, after multiplicity adjustment. Or, we could determine what size effects we would need across our five outcomes to reliably detect an impact on at least one of them. We set our goal by specifying the type (`power.definition`) and desired power (`target.power`).

Here, for example, we find the MDES if we want an 80% chance of detecting an impact on our first outcome when using the Holm procedure:

```
R> library("PUMP")
R> m <- pump_mdes(d_m = "d3.2_m3fc2rc", MTP = "HO",
+ target.power = 0.80, power.definition = "D1indiv", M = 5, J = 3,
+ K = 21, nbar = 258, Tbar = 0.50, alpha = 0.05, numCovar.1 = 5,
+ numCovar.2 = 3, R2.1 = 0.1, R2.2 = 0.7, ICC.2 = 0.05, ICC.3 = 0.4,
+ rho = 0.4)
```

The results are easily made into a nice table (Table 5) via the `knitr::kable()` command:

```
R> knitr::kable(m, digits = 3, booktabs = TRUE, position = "t!",
+ caption = "MDES estimate.") %>%
+ kableExtra::kable_styling(position = "center")
```

The answers `pump_mdes()` gives are approximate, as we are calculating them via simulation. To control accuracy, we can specify a tolerance (`tol`) of how close the estimated power needs to be to the desired target along with the number of iterations in the search sequence (via `start.tnum`, `tnum`, and `final.tnum`). The search will stop when the estimated power is within `tol` of `target.power`, as estimated by `final.tnum` iterations. Lower `tol` and higher `tnum` values will give more exact results (and take more computational time).

Changing the type of power is straightforward: For example, to identify the MDES for 1-minimal power (i.e., what effect size do we have to assume across all observations such that we will find at least one significant result with 80% power?), we simply update our result with our new power definition:

```
R> m2 <- update(m, power.definition = "min1")
```

```
mdes result: d3.2_m3fc2rc d_m with 5 outcomes
target min1 power: 0.80
```



```
MTP Adjusted.MDES min1.power SE
HO 0.08048574 0.78425 0.01
(5 steps in search)
```

The `update()` method can replace any number of arguments of the prior call with new ones, making exploration of different scenarios very straightforward.<sup>6</sup> Our results show that if we just want to detect at least one outcome with 80% power, we can reliably detect an effect of size 0.08 (assuming all five outcomes have effects of at least that size).

When estimating power for multiple outcomes, it is important to consider cases where some of the outcomes in fact have null, or very small, effects, to hedge against circumstances such as one of the outcomes not being well measured. One way to do this is to set two of our outcomes to no effect with the `numZero` parameter:

```
R> m3 <- update(m2, numZero = 2)

mdes result: d3.2_m3fc2rc d_m with 5 outcomes
target min1 power: 0.80
MTP Adjusted.MDES min1.power SE
HO 0.08971374 0.79125 0.01
(13 steps in search)
```

The MDES goes up, as expected: When there are not effects on some outcomes, there are fewer good chances for detecting an effect. Below we provide a deeper dive into the extent to which `numZero` can affect power estimates.

### 6.3. Determining necessary sample size

The MDES calculator tells us what we can detect given a specific design. We might instead want to ask how much larger our design would need to be in order to achieve a desired MDES. In particular, we might want to determine the needed number of students per school, the number of schools per randomization block, or the number of randomization blocks to detect an effect of a given size. The `pump_sample` method will search over any one of these. The `typesample` parameter specifies the level for which we are determining the sample size:

- `nbar` is the average number of students per school (level one units per level two group),
- `J` is the average number of schools per randomization block (level two units per level three group), and
- `K` is the number of randomization blocks (level three units).

To calculate sample size for a three-level model, we must know the sample size of two of the levels, and then we can determine the needed sample size of the third. For example, if we know that there are 258 students on average per school, and we know that we want to have 10 randomization blocks, how many schools should we put in each randomization block?

<sup>6</sup>The `update()` method re-runs the underlying call of `pump_mdes()`, `pump_sample()`, or `pump_power()` with the revised set of design parameters. You can even change which call to use via the `type` parameter.

Power definition	No adjustment	Holm adjustment
Individual outcome 1	0.7	0.53
Individual outcome 2	0.7	0.52
Individual outcome 3	0.7	0.53
Individual outcome 4	0.7	0.53
Individual outcome 5	0.7	0.53
Mean individual	0.7	0.53
1-minimum		0.81
2-minimum		0.64
3-minimum		0.51
4-minimum		0.39
Complete		0.33

Table 6: Power table.

Below, we instead assume we have a fixed randomization block size of 3, and we calculate how many blocks we would need to achieve a MDES of 0.10 for 1-minimal power (this answers the question of how big of an experiment we need in order to have an 80% chance of finding at least one outcome significant, if all outcomes had a true effect size of 0.10).

```
R> smp <- pump_sample(d_m = "d3.2_m3fc2rc", MTP = "H0",
+ typesample = "K", target.power = 0.80, power.definition = "min1",
+ tol = 0.01, MDES = 0.10, M = 5, nbar = 258, J = 3, Tbar = 0.50,
+ alpha = 0.05, numCovar.1 = 5, numCovar.2 = 3, R2.1 = 0.1,
+ R2.2 = 0.7, ICC.2 = 0.05, ICC.3 = 0.40, rho = 0.4)
```

```
sample result: d3.2_m3fc2rc d_m with 5 outcomes
target min1 power: 0.80
MTP Sample.type Sample.size min1.power SE
H0 K 16 0.797 0.01
(18 steps in search)
```

We would need only 16 blocks, rather than the originally specified 21, giving 48 total schools in our study, to achieve 80% 1-minimal power.

We recommend checking the MDES and sample size outputs, as the estimation error combined with the stochastic search can give results a bit off the target power in some cases. A check is easy to do; simply run the found design through `pump_power()`, which directly calculates power for a given scenario, to see if we recover our original target power (we can use `update()` and set the type to `power` to pass all the design parameters automatically). When we do this, we can also increase the number of iterations to get more precise estimates of power, as well:

```
R> p_check <- update(smp, type = "power", tnum = 50000,
+ long.table = TRUE)
```

As shown in Table 6, when calculating power directly, we get power for all the implemented definitions of power applicable to the design. The first five rows of Table 6 are the powers for

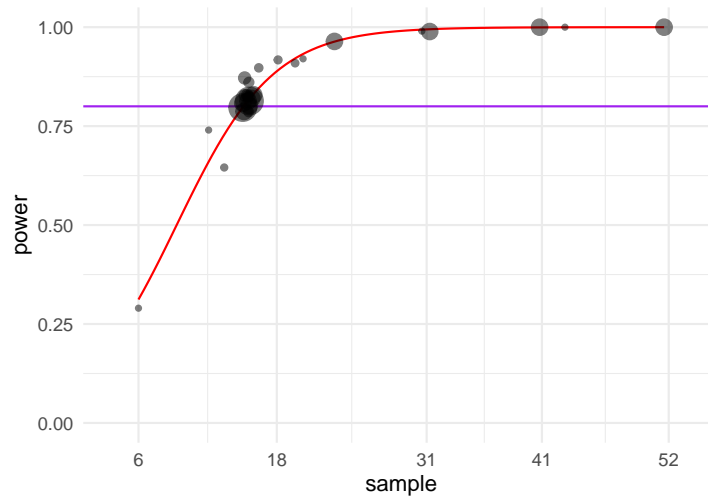


Figure 1: Power estimates as sample size changes.

rejecting each of the five outcomes – they are (up to simulation error) the same since we are assuming the same MDES and other design parameters for each. The “mean individual” is the mean individual power across all outcomes. The first numeric column is power without adjustment, and the second has our power estimate with the listed  $p$  value adjustment (Holm in our example).

The next rows show different multi-outcome definitions of power. In particular, `1-minimum` shows the chance of rejecting at least one hypothesis. The `complete` row shows the power to reject all hypotheses; it is only defined if all outcomes are specified to have a non-zero effect.<sup>7</sup>

In Figure 1 we can look at a power curve of our `pump_sample()` call to assess how sensitive power is to our level two sample size:<sup>8</sup>

```
R> plot(smp)
```

*Remark.* In certain settings, a wide range of sample sizes may result in very similar levels of power. In this case, the algorithm may return a sample size that is larger than necessary. This pattern mainly occurs for sample sizes at lower levels of the hierarchy; e.g., for `nbar` for all models, and for `nbar` and `J` for three-level models. Thus, we recommend always plotting the output so that we can see if there are flat regions in the power curve. In addition, due to the nature of the search algorithm or structural issues, occasionally the algorithm may not converge. For example, in some settings the power curves hit an asymptote, such that even an infinite sample size would not be able to reach a particular target power. For a more detailed discussion of these challenges, see the package sample size vignette.

#### 6.4. Comparing adjustment procedures

It is easy to rerun the above power calculation using the Westfall-Young step-down procedure or other procedures of interest. Alternatively, simply provide a list of procedures you wish to

<sup>7</sup>The package does not show power for these without adjustment for multiple testing, as that power would be grossly inflated and meaningless.

<sup>8</sup>The points on the plots show the evaluated simulation trials, with larger points corresponding to more iterations and greater precision.

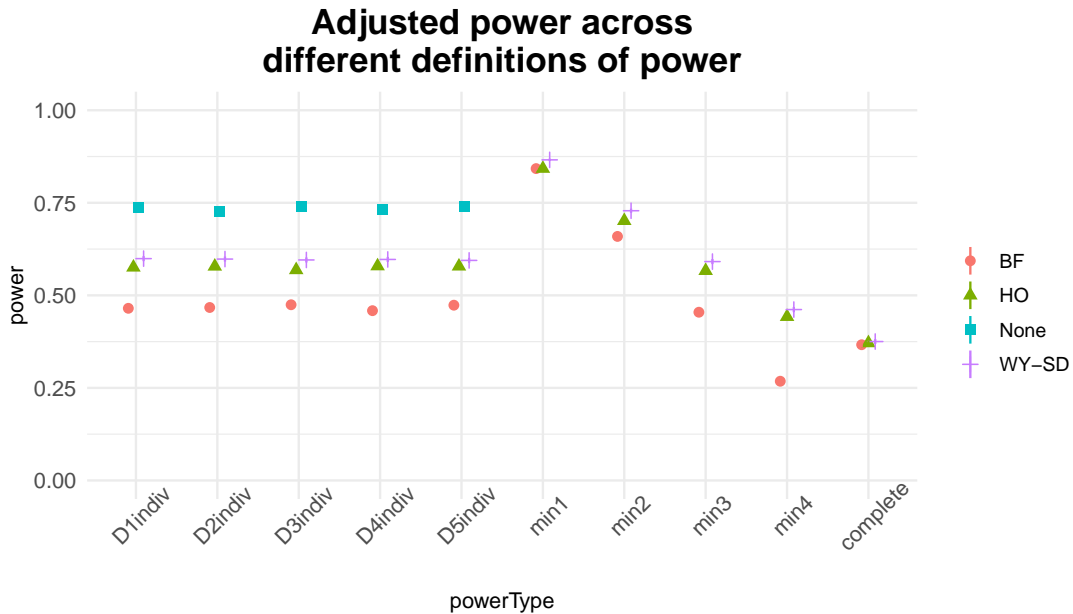


Figure 2: Power estimates across definitions and MTPs.

compare. If you provide a list, the package will re-run the power calculator for each item on the list. Here we obtain power for our scenario using Bonferroni, Holm and Westfall-Young adjustments, and plot the results using the default `plot()` method in Figure 2:

```
R> p2 <- update(p_check, MTP = c("BF", "HO", "WY-SD"), tnum = 5000,
+ parallel.WY.cores = 2)
R> plot(p2)
```

To speed up computation, we reduce `tnum` because of the computationally intensive nature of the WY-SD procedure. We also set `parallel.WY.cores = 2` to parallelize the computation across 2 cores.

The Westfall-Young step-down adjustment, which is more computationally intensive but can be less conservative, exploits the correlation in our outcomes (`rho = 0.4`). However, in this scenario it does not result in a substantial difference in individual power compared to the Holm procedure.

### 6.5. Exploring sensitivity to design parameters

Within the **PUMP** package we have two general ways of exploring design sensitivity. The first is with `update()`, which allows for quickly generating a single alternate scenario. To explore sensitivity to different design parameters more systematically, use the `grid()` functions, which calculate power, MDES, and sample size for all combinations of a set of passed parameter values. There are two main differences between the two approaches. First, the `grid` functions allow for systematic exploration of many possible combinations, while `update()` only allows the user to explore one value at a time. Second, `update()` allows for different values of a parameter for the different outcomes. In contrast, the `grid` functions do not allow design parameters, including MDES, to vary across outcomes, and assumes the same parameter

value across all outcomes. This assumption is made for simplicity of syntax. When faced with contexts where it is believed that these parameters do vary, we recommend using average values for the broader searches, and then double-checking a small set of potential final designs with the `pump_power()` method.

We first illustrate the `update()` approach, and then turn to illustrating `grid()` across three common areas of exploration: ICCs, the correlation of test statistics, and the assumed number of non-zero effects. The last two are particularly important for multiple outcome contexts.

For efficiency of future calculations, before proceeding we save a power call with a smaller `tnum` to reduce computation time.

```
R> pow <- update(p_check, tnum = 10000)
```

### *Exploring power with update()*

Update allows for a quick change of some of the set of parameters used in a prior call; we saw `update()` used several times above. As a further example, here we examine what happens if the ICCs are more equally split across levels two and three:

```
R> p_b <- update(pow, ICC.2 = 0.20, ICC.3 = 0.25)
R> print(p_b)
```

```
power result: d3.2_m3fc2rc d_m with 5 outcomes
  MTP D1indiv D2indiv D3indiv D4indiv D5indiv indiv.mean  min1  min2  min3
None  0.2616  0.2563  0.2629  0.2606  0.2577    0.25982   NA   NA   NA
  HO  0.1130  0.1082  0.1110  0.1115  0.1072    0.11018  0.2961  0.1292  0.0679
  min4 complete
      NA      NA
0.0381  0.0274
0.000 <= SE <= 0.002
```

We immediately see that either our assumption of little variation in level two or substantial variation in level three mattered a great deal for power.

When calculating power for a given scenario, it is also easy to vary many of our design parameters by outcome. For example, if we thought we had better predictive covariates for our second outcome, we might try:

```
R> p_d <- update(pow, R2.1 = c(0.1, 0.3, 0.1, 0.2, 0.2),
+   R2.2 = c(0.4, 0.8, 0.3, 0.2, 0.2))
R> print(p_d)
```

```
power result: d3.2_m3fc2rc d_m with 5 outcomes
  MTP D1indiv D2indiv D3indiv D4indiv D5indiv indiv.mean  min1  min2  min3
None  0.4512  0.8783  0.4057  0.3723  0.3680    0.49510   NA   NA   NA
  HO  0.2614  0.6904  0.2287  0.2115  0.2077    0.31994  0.7493  0.4051  0.2327
  min4 complete
      NA      NA
0.1347  0.0991
0.001 <= SE <= 0.002
```

Notice how the individual powers are heavily changed. The  $d$ -minimal powers naturally take the varying outcomes into account as we are calculating a joint distribution of test statistics that will have the correct marginal distributions based on these different design parameter values.

After several calls to `update()`, we may lose track of where we are; to find out, we can always check details with `print_design()` or `summary()`:

```
R> summary( p_d )

power result: d3.2_m3fc2rc d_m with 5 outcomes
MDES vector: 0.1, 0.1, 0.1, 0.1, 0.1
nbar: 258 J: 3 K: 16 Tbar: 0.5
alpha: 0.05
Level:
  1: R2: 0.1 / 0.3 / 0.1 / 0.2 / 0.2 (5 covariates)
  2: R2: 0.4 / 0.8 / 0.3 / 0.2 / 0.2 (3 covariates)   ICC: 0.05   omega: 0
  3: fixed effects   ICC: 0.4   omega: 0
rho = 0.4
MTP D1indiv D2indiv D3indiv D4indiv D5indiv indiv.mean min1 min2 min3
None 0.4512 0.8783 0.4057 0.3723 0.3680 0.49510 NA NA NA
H0 0.2614 0.6904 0.2287 0.2115 0.2077 0.31994 0.7493 0.4051 0.2327
min4 complete
NA NA
0.1347 0.0991
0.001 <= SE <= 0.002
(tnum = 10000)
```

Using `update` allows for targeted comparison of major choices, but if we are interested in how power changes across a range of options, we can do this more systematically with the `grid()` functions, as we do next.

### *Exploring the effect of the ICC*

We above saw that the ICC does affect power considerably. We next extend this evaluation by exploring a range of options for both level two and three ICCs, so we can assess whether our power is sufficient across a set of plausible values. The `update_grid()` call makes this straightforward: We pass our baseline scenario along with lists of parameters to additionally explore.

We can then easily visualize the variation in 1-minimal power by calling `plot()` on the object, shown in Figure 3.

```
R> gridICC <- update_grid(pow, ICC.2 = seq(0, 0.30, 0.05),
+   ICC.3 = seq(0, 0.60, 0.20))
R> plot(gridICC, power.definition = "min1")
```

Note that in addition to `update_grid()`, there are also base functions `pump_power_grid()`, `pump_mdes_grid()`, and `pump_sample_grid()`.

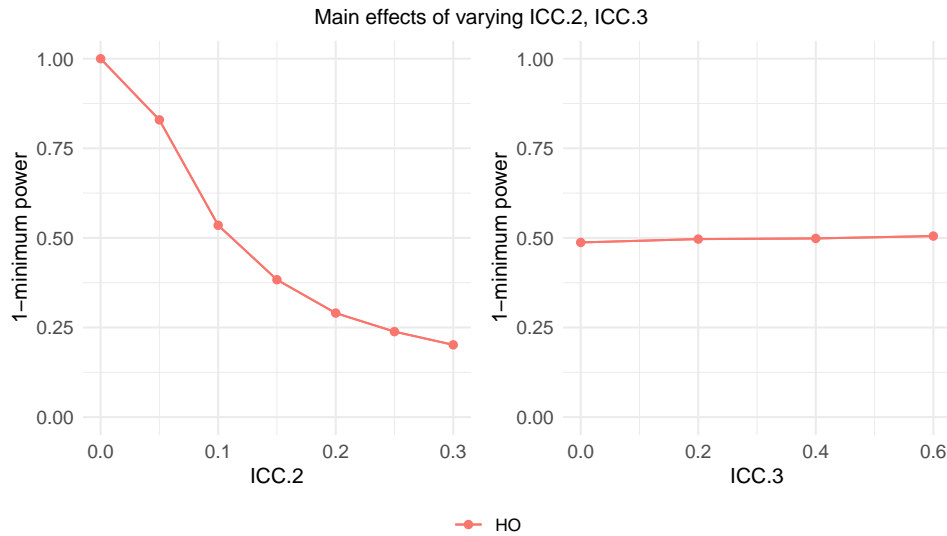


Figure 3: Power estimates as ICC changes.

We see that higher  $ICC_2$  radically reduces power to detect anything and  $ICC_3$  does little. To understand why, we turn to our standard error formula for this design and model:

$$Q_m = \sqrt{\frac{ICC_2(1 - R_2^2)}{\bar{T}(1 - \bar{T})JK} + \frac{(1 - ICC_2 - ICC_3)(1 - R_1^2)}{\bar{T}(1 - \bar{T})JK\bar{n}}}.$$

In the above, the  $\bar{n} = 258$  students per group makes the second term very small compared to the first, regardless of the  $ICC_2$  or  $ICC_3$  values. The first term, however, is a direct scaling of  $ICC_2$ ; changing it will change the standard error, and therefore power, a lot. To understand patterns for other designs, all provided designs and models implemented in the package are discussed, along with corresponding formula such as these, in the technical appendix.

For grid searches we recommend reducing the number of iterations, via `tnum`, to speed up computation. As `tnum` shrinks, we will get increasingly rough estimates of power, but even these rough estimates can help us determine trends.

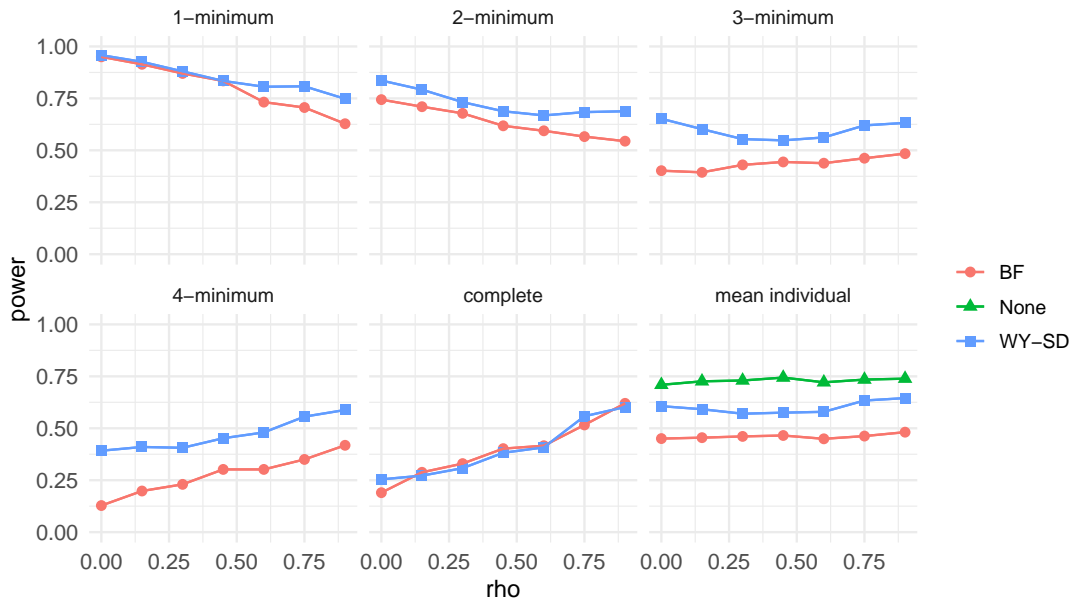
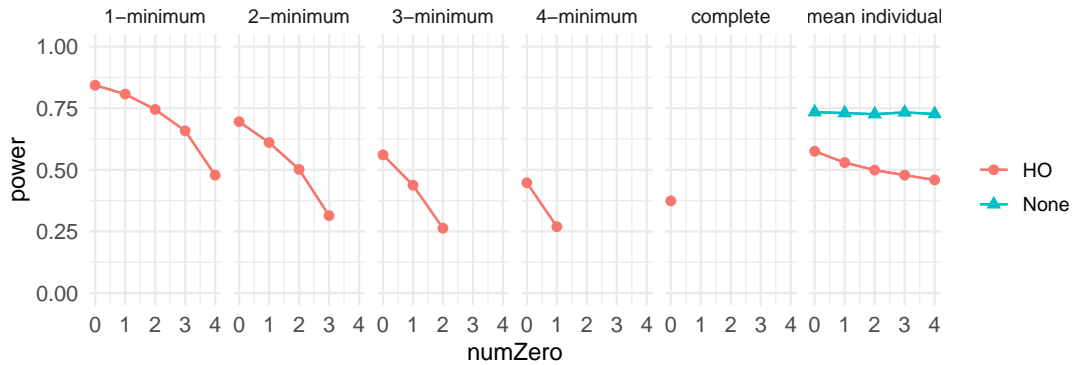
### Exploring the effect of $\rho$

The correlation of test statistics,  $\rho$ , is a critical parameter for how power will play out across the multiple tests. The effect of the correlation parameter may vary across definitions of power. For example, consider 2-minimal power: On one hand, correlated statistics make individual adjustment less severe, and on the other correlation means we succeed or fail all together. We can explore this relationship relatively easily by letting `rho` vary as so:

```
R> gridRho <- update_grid(pow, MTP = c("BF", "WY-SD"),
+   rho = seq(0, 0.9, by = 0.15), tnum = 500, B = 3000)
```

We then plot our results in Figure 4.

```
R> plot(gridRho)
```

Figure 4: Power estimates as  $\rho$  changes.Figure 5: Power estimates as  $\text{numZero}$  changes.

First, we see that the Westfall-Young single-step procedure does result in somewhat higher power than Bonferroni for many power definitions. Second, the effect on individual adjustment is flat, as anticipated. Third, 1-minimal power and 2-minimal power fall as  $\rho$  increases, while complete power climbs. See [Porter \(2018\)](#) for further discussion of the effect of correlation on power; while the paper focuses on the multisite randomized trial context, the lessons learned there apply to all designs, as the only substantive differences between different design and modeling choices is in how we calculate the unadjusted distribution of their test statistics.

### *Exploring the effect of null outcomes*

We finally explore varying the number of outcomes with no effects. This exploration is an important way to hedge a design against the possibility that some number of the identified outcomes are measured poorly, or are simply not impacted by treatment. We use a grid search, varying the number of outcomes that have no treatment impact via the `numZero` design parameter in Figure 5:



```
R> gridZero <- update_grid(pow, numZero = 0:4, M = 5)
R> plot(gridZero, nrow = 1)
```

There are other ways of exploring the effect of weak or null effects on some outcomes. In particular, the `pump_power()` and `pump_sample()` methods allow the researcher to provide an MDES vector with different values for each outcome, including 0s for some outcomes. The `grid()` functions, by contrast, take a single MDES value for the non-null outcomes, with a separate specification of how many of the outcomes are 0. (This single value plus `numZero` parameter also works with `pump_power()` if desired.)

### *Checking the correlation between test statistics*

Section 3 discussed that **PUMP** takes the correlation between test statistics as a parameter. We generally use the assumed correlation between outcomes as a proxy for the correlation between test statistics, even though that approximation may not be exact. To make sure this assumption is not substantially affecting the power estimates, as a final step we use a built-in function to check the correlation between test statistics given our assumed model. The approach used by the checker tool mirrors the package validation approach. We generate  $S$  iterations of simulated data, and for each iteration calculate test statistics for each outcome, resulting in a matrix of dimension  $S \times M$ . Then, we estimate the pairwise correlations between the columns of this matrix. Due to the repeated simulation and analysis steps, the correlation checker can take several minutes (or longer) to run. Models with random effects or impacts have a much greater computation time than those without.

By default, the simulation process assumes the same correlation structure between all variables that vary by outcome: Covariates, random effects, and random impacts. Thus, if we have five outcomes, it assumes that each outcome has a single covariate, and those covariates have correlation  $\rho$ . One setting that would likely have this correlation structure is if the outcomes were test scores on different subjects (math, science, reading, etc.), and each outcome has a corresponding pre-test score as a covariate.

However, in the Diplomas Now example we instead assume that covariates are shared across outcomes. Thus, we set our correlation structure to have a correlation of 1 between covariates across outcomes. The parameters `rho.V`, `rho.X`, and `rho.C` correspond to the correlation matrices for covariates at levels three, two, and one. For more information about other parameters for simulating data, including all the correlation matrices, see the package vignette on simulating data.

```
R> covariate.corr.matrix <- gen_corr_matrix(M = 5, rho.scalar = 1)
R> cor.tstat <- check_cor(pow, rho.V = covariate.corr.matrix,
+   rho.X = covariate.corr.matrix, rho.C = covariate.corr.matrix,
+   n.sims = 500)
```

The function outputs a correlation matrix. Given that we assume all the pairwise correlations are the same, we can take the mean across them to get a single estimate of the correlation between test statistics.

```
R> est.cor <- mean(cor.tstat[lower.tri(cor.tstat)])
R> print(est.cor)
```

```
[1] 0.3172791
```

We find that the estimated correlation between test statistics is close, but not equal, to our assumed correlation between outcomes, 0.4. If there was a substantial discrepancy between these two values, we would want to run a final check on power using the found correlation to determine if the discrepancy affected our power estimates.

## 6.6. Calls and methods

For user reference, we wrap up with a brief summary of the package functionality.

*Package calls.* First, we summarize the main package calls.

The base functions are:

- `pump_power()` for calculating power given an experimental design and assumed model and MDES.
- `pump_mdes()` for calculating MDES given a target power and sample sizes.
- `pump_sample()` for calculating the required sample size at a given level for achieving a given target power for a given MDES and sample sizes at other levels.

Exploratory functions are:

- `pump_power_grid()`, `pump_mdes_grid()`, and `pump_sample_grid()` for calculating the given output over a range of possible parameter values.
- `update()` to re-run an existing calculation with a small number of parameters updated.
- `update_grid()` to re-run an existing calculation but over a grid of possible parameter values.

*Methods.* Second, we summarize methods that can be applied to **PUMP**-generated objects. The **PUMP** package returns two types of S3 objects.

- ‘`pumpresult`’ objects are returned from single scenario calls: `pump_power()`, `pump_mdes()`, `pump_sample()`, and calls to `update()`.
- ‘`pumpgridresult`’ objects are returned from grid calls: `pump_power_grid()`, `pump_mdes_grid()`, `pump_sample_grid()`, and calls to `update_grid()`.

The package has a variety of methods that can be called directly on ‘`pumpresult`’ objects.

- `print()` displays a concise summary of the most relevant inputs and results of the call.
- `summary()` prints a more extensive output, containing a full summary of both the full list of inputs and the results of the call.
- `print_context()` provides a summary of the user inputs, including the design and model and the parameter values.
- `plot()` returns different plots tailored to whether the results are for power, MDES, or sample size:

- For power objects, it displays power across all power definitions and MTPs.
  - For MDES and sample size objects, by default it displays a power curve showing how power changes as sample size or MDES changes.
  - For MDES and sample size objects, the user can instead request a diagnostic plot of the power search algorithm using `type = "search"`.
- `power_curve()` returns a data frame of power values over a range of MDES or sample size values.
  - `search_path()` returns the search history of the search algorithm for MDES and sample size calls.
  - `transpose_power_table()` converts a power table between wide and long formats.
  - `as.data.frame()` casts the object to a data frame of the results.
  - `gen_sim_data()` generates a set of simulated data using a data-generating process from the assumed design, model, and parameters. For more details about functions to simulate data, see the package vignette on simulating data.
  - `check_cor()` checks the correlation between test statistics using a simulation approach.

Many of the above methods also apply to ‘`pumpgridresult`’ objects, although some are not relevant to grid objects. The main difference in behavior between the ‘`pumpresult`’ and ‘`pumpgridresult`’ objects is the output of `plot()` function. For an example of `plot()` called on a ‘`pumpresult`’ object, see Figure 2. In contrast, for an example of `plot()` called on a grid object, see Figure 3. For grid objects, the `plot()` function plots a facet wrap displaying how power changes across all MTPs, power definitions, and varying parameters provided during the grid call. If the user wants a smaller set of results, they can specify a single `power.definition`, or use `var.vary` to only plot variation in one parameter value. If the grid call varied multiple parameters, then each plot averages power across all other factors to plot main effects. For example, in Figure 3, the first plot averages over all values of `ICC.3` to show how power varies with just `ICC.2`, and the second plot does the opposite.

## 7. Conclusion

We introduce the Power Under Multiplicity Project (**PUMP**) R package, which estimates power for multi-level randomized control trials with multiple outcomes. **PUMP** allows users to estimate power, MDES, and sample size requirements for a wide variety of commonly used RCT designs and models across different definitions of power and applying different MTPs. The functionality of **PUMP** fills an important gap, as existing tools do not allow researchers to conduct power, MDES or sample size calculations when applying a MTP in a RCT.

The main advantage of the **PUMP** package is to provide easily accessible estimation procedures so that users can properly account for power when making adjustments for multiple hypothesis testing. However, one of the additional strengths of the package is the ease with which a user can explore the effect of different designs, models, and parameter assumptions on power, MDES or sample size. Even if a user is only interested in a single outcome, **PUMP**

provides useful functionality for more robust power calculations. A user can and should try a range of parameter values to determine the sensitivity of the power of their study to different assumptions; this package simplifies that process.

In addition to this paper, there is a variety of supporting information.

- The package is available on CRAN, <https://CRAN.R-project.org/package=PUMP>.
- The code is available on GitHub, <https://github.com/MDRCNY/PUMP>.
- An online interface is available at <https://public.mdrc.org/pump/>.
- The technical appendix contains detailed information about each design and model, the assumed data generating process, and precise descriptions of parameters such as ICC and  $\omega$ . It is a useful reference not just for users of the package, but also as a general summary of frequentist multi-level models.
- The package has an additional vignette on understanding sample size calculations, which can present unique challenges when calculating sample sizes below the top level.
- The package has supplementary functions that allow a user to simulate data from multi-level models. Although these functions are not directly related to the power calculations, we provide them as a potentially useful tool. A short vignette explains these functions.
- The code and results for validating the package are in a separate repository, [https://github.com/MDRCNY/pump\\_validate](https://github.com/MDRCNY/pump_validate).

## Acknowledgments

Development of this package was supported by a grant from the Institute of Education Sciences (R305D170030). Kristin Porter was employed by MDRC throughout the grant period for this project. Kristen Hunter was previously affiliated with the Harvard University Department of Statistics while conducting this work. We would like to thank Zarni Htet at MDRC for his contributions to the code for this project and for creating the accompanying Shiny application. We also acknowledge the Diplomas Now team at MDRC. We would like to thank members of the Harvard CARES lab for their feedback on the manuscript.

## References

- Bang H, Jung S, George SL (2005). “Sample Size Calculation for Simulation-Based Multiple-Testing Procedures.” *Journal of Biopharmaceutical Statistics*, **15**, 957–967. doi:10.1080/10543400500265710.
- Bates D, Mächler M, Bolker B, Walker S (2015). “Fitting Linear Mixed-Effects Models Using **lme4**.” *Journal of Statistical Software*, **67**(1), 1–48. doi:10.18637/jss.v067.i01.
- Benjamini Y, Hochberg Y (1995). “Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing.” *Journal of the Royal Statistical Society B*, **57**, 289–300. doi:10.1111/j.2517-6161.1995.tb02031.x.

- Berger RL (1982). “Multiparameter Hypothesis Testing and Acceptance Sampling.” *Technometrics*, **24**(4), 295–300. doi:10.2307/1267823.
- Berger RL, Hsu JC (1996). “Bioequivalence Trials, Intersection-Union Tests and Equivalence Confidence Sets.” *Statistical Science*, **11**(4), 283–319. doi:10.1214/ss/1032280304.
- Bloom HS (2006). “The Core Analytics of Randomized Experiments for Social Research.” *Technical report*, MDRC. URL <https://www.mdrc.org/publication/core-analytics-randomized-experiments-social-research>.
- Bretz F, Hothorn T, Westfall P (2010). *Multiple Comparisons Using R*. Chapman & Hall/CRC. doi:10.1201/9781420010909.
- Bulus M, Dong N, Kelcey B, Spybrook J (2022). “**PowerUpR**: Power Analysis Tools for Multi-level Randomized Experiments.” R package version 1.1.0, URL <https://CRAN.R-project.org/package=PowerUpR>.
- Chang W, Cheng J, Allaire JJ, Sievert C, Schloerke B, Xie Y, Allen J, McPherson J, Dipert A, Borges B (2023). **shiny**: *Web Application Framework for R*. R package version 1.8.0, URL <https://CRAN.R-project.org/package=shiny>.
- Chen J, Luo J, Liu K, Mehrotra D (2011). “On Power and Sample Size Computation for Multiple Testing Procedures.” *Computational Statistics & Data Analysis*, **55**, 110–122. doi:10.1016/j.csda.2010.05.024.
- Corrin W, Sepanik S, Rosen R, Shane A (2016). “Addressing Early Warning Indicators: Interim Impact Findings from the Investing in Innovation (I3) Evaluation of Diplomas Now.” *Technical report*, MDRC. URL <https://www.mdrc.org/publication/addressing-early-warning-indicators>.
- Deng X, Xu J, Wang C (2008). “Improving the Power for Detecting Overlapping Genes from Multiple DNA Microarray-Derived Gene Lists.” *BMC Bioinformatics*, **9**. doi:10.1186/1471-2105-9-s6-s14.
- Dong N, Maynard R (2013). “PowerUp!: A Tool for Calculating Minimum Detectable Effect Sizes and Minimum Required Sample Sizes for Experimental and Quasi-Experimental Design Studies.” *Journal of Research on Educational Effectiveness*, **6**(1), 24–67. ISSN 1934-5747. doi:10.1080/19345747.2012.673143.
- Dudoit S, Shaffer JP, Boldrick JC (2003). “Multiple Hypothesis Testing in Microarray Experiments.” *Statistical Science*, **18**(1), 71–103. doi:10.1214/ss/1056397487.
- Dunn OJ (1959). “Estimation of the Medians for Dependent Variables.” *The Annals of Mathematical Statistics*, **30**(1), 192–197. doi:10.1214/aoms/1177706374.
- Dunn OJ (1961). “Multiple Comparisons among Means.” *Journal of the American Statistical Association*, **56**(293), 52–64. doi:10.1080/01621459.1961.10482090.
- Ge Y, Dudoit S, Speed TP (2003). “Resampling-Based Multiple Testing for Microarray Data Analysis.” *Test*, **12**, 1–77. doi:10.1007/bf02595811.

- Gelman A, Hill J, Yajima M (2007). *Data Analysis Using Regression and Multi-level/Hierarchical Models*. Cambridge University Press.
- Gelman A, Hill J, Yajima M (2012). “Why We (Usually) Don’t Have to Worry about Multiple Comparisons.” *Journal of Research on Educational Effectiveness*, **5**, 189–211. doi:10.1080/19345747.2011.618213.
- Hedges LV, Rhoads C (2010). “Statistical Power Analysis in Education Research.” *Technical report*, National Center for Special Education Research. URL <http://eric.ed.gov/ERICWebPortal/detail?accno=ED509387>.
- Holm S (1979). “A Simple Sequentially Rejective Multiple Test Procedure.” *Scandinavian Journal of Statistics*, **6**(2), 65–70.
- Maurer W, Mellein B (1988). “On New Multiple Test Procedures Based on Independent *P*-Values and the Assessment of Their Powers.” In P Bauer, G Hommel, E Sonnemann (eds.), *Multiple Hypotheses Testing*, pp. 48–66. Springer-Verlag.
- Miratrix L, Weiss M, Henderson B (2021). “An Applied Researcher’s Guide to Estimating Effects from Multisite Individually Randomized Trials: Estimands, Estimators, and Estimates.” *Journal of Research on Educational Effectiveness*, **14**(1). doi:10.1080/19345747.2020.1831115.
- Porter KE (2018). “Statistical Power in Evaluations That Investigate Effects on Multiple Outcomes: A Guide for Researchers.” *Journal of Research on Educational Effectiveness*, **11**, 267–295. doi:10.1080/19345747.2017.1342887.
- Ramsey PH (1978). “Power Differences between Pairwise Multiple Comparisons.” *Journal of the American Statistical Association*, **75**, 479–487. doi:10.2307/2286584.
- Raudenbush SW, Bryk AS (2002). *Hierarchical Linear Models: Applications and Data Analysis Methods*. Sage Publications.
- R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Schochet PZ (2008). “Guidelines for Multiple Testing in Impact Evaluations of Educational Interventions. Final Report.” *Technical report*, Mathematica Policy Research. URL <http://eric.ed.gov/ERICWebPortal/contentdelivery/servlet/ERICServlet?accno=ED502199>.
- Schochet PZ (2016). “Statistical Theory for the RCT-YES Software: Design-Based Causal Inference for RCTs.” URL [https://www.mathematica.org/~media/publications/pdfs/education/statistical\\_theory.pdf](https://www.mathematica.org/~media/publications/pdfs/education/statistical_theory.pdf).
- Schochet PZ, Pashley NE, Miratrix LW, Kautz T (2021). “Design-Based Ratio Estimators and Central Limit Theorems for Clustered, Blocked RCTs.” *Journal of the American Statistical Association*, pp. 1–12. doi:10.1080/01621459.2021.1906685.
- Senn S, Bretz F (2007). “Power and Sample Size When Multiple Endpoints Are Considered.” *Pharmaceutical Statistics*, **6**, 161–170. doi:10.1002/pst.301.

- Spybrook J, Bloom HS, Congdon R, Hill CJ, Martinez A, Raudenbush SW (2011). “Optimal Design Plus Empirical Evidence: Documentation for the **Optimal Design** Software Version 3.0.” *Technical report*, William T. Grant Foundation. URL <http://wtgrantfoundation.org/resource/optimal-design-with-empirical-information-od>.
- Spybrook J, Hedges L, Borenstein M (2014). “Understanding Statistical Power in Cluster Randomized Trials: Challenges Posed by Differences in Notation and Terminology.” *Journal of Research on Educational Effectiveness*, **7**. doi:10.1080/19345747.2013.848963.
- Weiss MJ, Bloom HS, Verbitsky-Savitz N, Gupta H, Vigil AE, Cullinan DN (2017). “How Much Do the Effects of Education and Training Programs Vary Across Sites? Evidence From Past Multisite Randomized Trials.” *Journal of Research on Educational Effectiveness*, **10**(4), 843–876. doi:10.1080/19345747.2017.1300719.
- Westfall PH, Tobias RD, Wolfinger RD (2011). *Multiple Comparisons and Multiple Tests Using SAS*. The SAS Institute.
- Westfall PH, Young SS (1993). *Resampling-Based Multiple Testing: Examples and Methods for P-Value Adjustment*. John Wiley & Sons.

## A. Technical details for multiple testing procedures

*Weak and strong control.* A MTP “provides weak control of the FWER or the FDR at level  $\alpha$  if the control can only be guaranteed when all null hypotheses are true, e.g., when the effects on all outcomes are zero. A MTP provides strong control of the FWER or FDR at level  $\alpha$  if the control is guaranteed even when some null hypotheses are true and some are false, e.g., when there may be effects on at least some outcomes. Of course, strong control is preferred” (Porter 2018). The single-step and step-down Westfall Young MTPs always provide at least weak control of the FWER. In order for these procedures to provide strong control of the FWER, they require the assumption of subset pivotality (Ge, Dudoit, and Speed 2003). The distribution of the unadjusted test statistics or  $p$  values is said to have subset pivotality if for any subset of null hypotheses, the joint distribution of the test statistics or of the  $p$  values for the subset is identical to the distribution under the complete null. A consequence of this assumption is that the permutation of test statistics or  $p$  values can be done under the complete null hypothesis rather than under the unknown partial hypothesis (Ge *et al.* 2003).

*Minimal and complete power definitions.* As noted in Porter (2018), under the assumption that some effects are truly null, we must change our notion of power for  $d$ -minimal powers (e.g., 1-minimal power, 1/3-minimal power, etc.). While individual power is defined based on the probability of correctly rejecting false nulls, the definition for  $d$ -minimal power includes the probability of erroneous rejections of the true nulls in the set. For example, 1/3-minimal power is defined as the probability of detecting effects on at least 1/3 of the *total outcomes*  $M$ , regardless of the number of outcomes with true effects. That is, 1/3-minimal power is not defined as the probability of detecting effects among the  $M$  outcomes on which the effects truly exist. This reframing of power is necessary for power to be consistent. If  $d$ -minimal power were defined based on rejecting only false nulls, then the value and interpretation would change depending on what assumption the researcher is making about the number of false nulls, which is an unknown quantity. For example, with  $M = 5$  outcomes, the probability of detecting at least one effect would be very different depending on whether we assume all five outcomes are false nulls, or whether we assume only two of them are false nulls.

Complete power, which is the probability of detecting effects on all outcomes, has similar issues. We define complete power only in the context where all effects are assumed to be false nulls; if any outcomes are assumed to be true nulls, then complete power is undefined.

*Calculating complete power.* There is an additional technical note about the calculation of complete power, which has also been referred to as “conjunctive power” (Bretz *et al.* 2010) and “all pairs power” (Ramsey 1978). To calculate complete power, we do not need to adjust the  $p$  values, and can instead reject each individual test based on the unadjusted  $p$  values. Complete power is the power of the omnibus test constructed by whether or not we reject all the null hypotheses. This test was originally introduced as the intersection-union test because the null hypothesis is expressed as a union and the alternative hypothesis is expressed as an intersection (Berger 1982; Berger and Hsu 1996). Berger (1982) showed that if all the individual tests are level  $\alpha$ , the intersection-union test is also a level  $\alpha$  test. To provide some intuition, we do not need to adjust  $p$  values for complete power because it is a special case where we must reject *all* the hypothesis tests. Thus, there is no way for the omnibus test to be rejected by chance because of a favorable configuration (Chen *et al.* 2011). For example, consider a case in which we have four tests, with two false nulls and two true nulls.



If we consider 3-minimal power, we just need one of the two true negatives to be rejected by chance alone, and there are two ways for this to occur. For complete power, there is only one way for us to reject all of the nulls. The downside of an intersection-union test is that it is conservative: The FWER is generally less than  $\alpha$ . For example, if we have two independent tests with type I error  $\alpha$ , then if both of are true negatives, the probability of a type I error for the omnibus test (the probability of rejecting both null hypotheses) is  $\alpha^2$  (Deng, Xu, and Wang 2008).

## B. Validation

In order to validate that our power estimates are working as intended, we compared three different methods of estimating power:

- **PUMP**
- Monte Carlo simulations
- **PowerUpR**

First, for all types of power definitions and adjustments, we compare **PUMP** to the estimated power obtained from full Monte Carlo simulations. We follow the simulation approach outlined in detail in Section 4.1. For each of  $S$  iterations, we simulate data and calculate  $p$  values. After completing all iterations, we calculate power and a 95% confidence interval for the true power value, assuming a conservative standard error of  $\sqrt{0.25/S}$ . For individual, unadjusted power, we also compute values from **PowerUpR**.

To validate the estimates, we first check that the **PUMP** and **PowerUpR** estimates match. In some settings we expect some discrepancies between these values because **PUMP** has different assumptions than *PowerUp!* for certain models. For details about differences between **PUMP** and *PowerUp!* assumptions, see the Technical Appendices. Second, we check that the **PUMP** estimate is within the Monte Carlo confidence interval.

We also validate MDES and sample size calculations. For MDES, we choose one default scenario for each design and model, then input the already-calculated individual power and see if the output MDES is the same as the original input MDES. Similarly, for sample size validation, we input the already-calculated individual power and see if the output sample size (`nbar`, `J`, and `K` depending on design) is the same as the original input sample size. Our unit testing code conducts additional tests of this nature.

### B.1. Simulation parameters

In order to validate that the method works in a wide range of scenarios, we vary the following parameters. For most scenarios, we vary only one parameter at a time. Thus, to test varying  $\rho$ , we set  $\rho = 0.2$  with all other parameters being set to the default values, and try another scenario with  $\rho = 0.8$  with all other parameters being set to the default values. Table 7 shows the default parameter values and the other values we apply in order to test out the effect of varying that parameter.

We do not vary:

- `M = 3`.
- `J` and `K` are fixed across all scenarios for a given design and model.

Parameter	Default	Other values
school size $\bar{n}$	50	75, 100
$R^2$	0.1	0.6
$\rho$	0.5	0.2, 0.8
MDES	(0.125, 0.125, 0.125)	(0.125, 0, 0)
ICC	0.2	0.7
$\omega$	0.1	0.8

Table 7: Validation parameters.

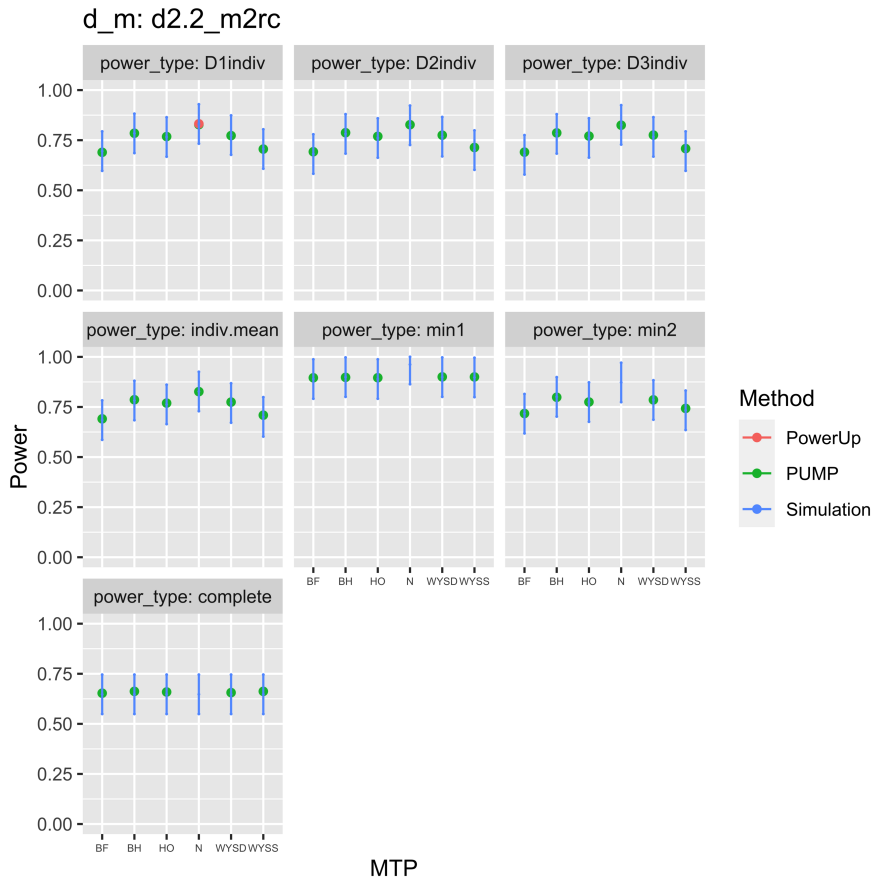


Figure 6: Validation plot.

## B.2. Validation results

Figure 6 is an example of a graph we use for validation. The green dots are **PUMP** estimate of power, the red dot is the **PowerUpR** estimate of power, and the 95% confidence intervals based on the Monte Carlo simulations are shown in blue. To validate that **PUMP** produces the expected result, we want to see the red and green points match, and for the green point to be within the blue interval. Figure 6 shows the results across different types of power and different MTPs. We repeat this plot for a variety of different parameter values for each design and model.

MTP	Adjusted MDES	D1indiv Power	Target MDES
Bonferroni	0.122	0.447	0.125
BH	0.127	0.578	0.125
Holm	0.125	0.540	0.125

Table 8: MDES validation.

MTP	Sample.type	Sample.size	D1indiv.power
Bonferroni	J	21	0.500
BH	J	21	0.580
Holm	J	20	0.544

Table 9: Sample size validation.

For MDES and sample size calculations we put in our found power, and then see if the `pump_mdes()` function returns the MDES we originally plugged in to achieve this power. In Table 8, the first column shows the calculated MDES, the middle column is the power we plugged into the calculation, and the last column shows the MDES that we are targeting. Thus, ideally we want the first and last columns to match.

Similarly, we validate our sample size calculations. Using our found power, we see if `pump_sample()` returns the original sample size. In Table 9, we are targeting a sample size of  $J = 20$ .

### Affiliation:

Kristen B. Hunter  
 University of New South Wales  
 School of Mathematics and Statistics & uDASH  
 Sydney, NSW, Australia  
 E-mail: [kristen.hunter@unsw.edu.au](mailto:kristen.hunter@unsw.edu.au)

Luke Miratrix  
 Harvard Graduate School of Education  
 Cambridge, MA, United States of America  
 E-mail: [lmiratrix@g.harvard.edu](mailto:lmiratrix@g.harvard.edu)

Kristin Porter  
 K.E. Porter Consulting LLC  
 Berkeley, CA, United States of America  
 E-mail: [kristin.porter@keporterconsulting.com](mailto:kristin.porter@keporterconsulting.com)