Power Under Multiplicity Project (PUMP): Estimating Power, Minimum Detectable Effect Size, and Sample Size When Adjusting for Multiple Outcomes

Kristen Hunter* Luke Miratrix† Kristin Porter‡

Abstract

For randomized controlled trials (RCTs) with a single intervention 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, which reduces the probability of detecting effects when they do exist. However, 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 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. Multiple outcomes 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, as researchers change their focus from one outcome to multiple outcomes, different definitions of statistical power emerge. `PUMP` allows researchers to consider a variety of definitions of power, as some may be more appropriate for the goals of their study. The package estimates power for frequentist multi-level mixed effects models, and supports a variety of commonly-used RCT designs and models and multiple testing procedures. 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.

1 Introduction

The `PUMP R` package fills in an important gap in open-source software tools to design multi-level randomized controlled trials (RCTs) with adequate statistical power. With this package, researchers can estimate statistical power, minimum detectable effect size (MDES), and needed sample size for multi-level experimental designs, in which 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 mixed effects regression models, which is currently the prevailing framework for estimating impacts from experiments in education and other social policy research.[1]

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 multiple testing procedure (MTP). MTPs adjust p-values

*Harvard University Department of Statistics
†Harvard Graduate School of Education
‡MDRC

[1]Other options include nonparametric or Bayesian methods, but these are less prevalent in applied research (for example, see Gelman, Hill, and Yajima (2012), Gelman, Hill, and Yajima (2007)).

to reduce the likelihood of spurious findings when researchers are testing for effects on multiple outcomes. 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 et al. (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 include:

- **individual power**: the probability of detecting an effect of a particular size (specified by the researcher) or larger for each hypothesis test. Individual power corresponds to how power is defined when there is focus on a single outcome.
- $1-$**minimal power**: the probability of detecting effects of at least a particular size on at least one outcome. 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.
- **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. 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 when multiplicity adjustments are taken into account. 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. In both of these cases, by not accounting for both the challenges and opportunities arising from multiple outcomes, a researcher may find they have wasted resources, either by designing an underpowered study that cannot detect the desired effect sizes, or by designing an overpowered study that had a larger sample size than necessary. We introduce the `PUMP` package to allow for directly answering questions that take multiple outcomes into account, such as:

- How many schools would I need to detect a given effect on at least three of my five outcomes?
- 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 had impact?

The methods in the PUMP package build on those introduced in Porter (2018). This earlier paper focused only 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 nine additional 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! `R` package (and its related tools in the form of a spreadsheet and Shiny application), which compute power or MDES for multi-level RCTs with a single outcome (Dong and Maynard (2013)). 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); Raudenbush et al. (2011); Spybrook et al. (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.

In order to calculate power, the researcher specifies information about the sample size at each level, the minimum detectable effect size for each outcome, the level of statistical significance, and parameters of the data generating distribution. The minimum detectable effect size is the smallest true effect size the study can detect with the desired statistical significance level, in units of standard deviations. An "effect size" generally refers to the standardized mean difference effect size, which "equals the difference in mean outcomes for the treatment group and control group, divided by the standard deviation of outcomes across subjects within

experimental groups" (Bloom (2006)). Researchers often use effect sizes to standardize outcomes so that outcomes with different scales can be directly compared.

The package includes three core functions:

- `pump_power()` for calculating power given a experimental design and assumed model, parameters, and minimum detectable effect size.
- `pump_mdes()` for calculating minimum detectable effect size given a target power and sample sizes.
- `pump_sample()` for calculating the required sample size for achieving a given target power for a given minimum detectable effect size.

For any of these core functions, the user begins with two main choices. First, the user chooses the assumed design and model of the RCT. The `PUMP` package covers a range of multi-level designs, up to three levels of hierarchy, that researchers typically use in practice, in which research units are nested in hierarchical groups. Our power calculations assume the user will be analyzing these RCTs using frequentist mixed-effects regression models, containing a combination of fixed or random intercepts and treatment impacts at different levels, as we explain in detail in Section 4.1 and in the Technical Appendix. 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 parameters of the data generating distribution.

The package also includes functions that allow users to easily explore power over a range of possible values of 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_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.
- `plot()` on `PUMP`-generated objects to generate plots (including grid outputs).

The authors of the `PUMP` package have also created a web application built with R Shiny. 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://mdrc.shinyapps.io/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. We note, however, that the problem of power estimation for multi-level RCTs is not exclusive to the educational setting. 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 packages 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 is designed to be robust enough to transform high-poverty and high-needs middle and high schools attended by many 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 RCT contains three levels (students within schools within districts) with random assignment at level two (schools). 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. Schools in the active treatment group were assigned to implement the Diplomas Now model, while the schools in the control group continued their existing school programs or implemented other reform strategies of their choosing (Corrin et al. (2016).) 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.

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 measures 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 domains 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 family-wise 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. The resulting multiplicity of statistical hypothesis tests can lead to spurious findings of effects.[2] 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 reduced. Much of the proceeding explanation is borrowed from or parallels discussion found in Porter (2018).

[2]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.

We first remind the reader of how raw, or unadjusted, p-values are calculated in this setting, before introducing adjustments due to multiple testing. Consider that a researcher is interested in testing the impact of an intervention on M outcomes. In our running example of the Diplomas Now study, if we had five outcomes in the attendance group, we would have $M = 5$. We apply a frequentist hypothesis testing framework, and frame impacts in terms of effect sizes (ES). For outcome m, we can test a null hypothesis of no effect, $H_{0_m} : ES_m = 0$, against an alternative hypothesis $H_{1_m} : ES_m \neq 0$ for a two-sided tests or $H_{1_m} : ES_m > 0$ or $H_{1_m} : ES_m < 0$ for a one-sided test. A significance test, such as a two- or one-sided t-test, would then typically be driven by a test statistic given by

$$t_m = \frac{\widehat{ES}_m}{SE(\hat{ES}_m)}, \tag{1}$$

where $SE(\hat{ES}_m)$ is the standard error. A raw p-value would then be computed as the probability of being at least as extreme as the one observed, given that the null hypothesis is true. The term "raw" is used to denote unadjusted p-values, in contrast to values that have been adjusted using a MTP. For a two-sided test, the raw p-value for test m is $p_m = 2 * Pr(T_m \geq |t_m|)$. The `PUMP` package allows for either one-sided or two-sided tests, but we proceed assuming two-sided tests going forward.[3]

When testing a *single* hypothesis under this framework, "researchers typically specify α, the maximum acceptable probability of making a Type I error. A Type I error is the probability of erroneously rejecting the null hypothesis when it is true. The quantity α is also referred to as the significance level. If $\alpha = 0.05$, then the null hypothesis is rejected if the p-value is less than 0.05" (Porter (2018)).

In contrast, "when one tests *multiple* hypotheses under this framework (such that $M > 1$) and one conducts a separate test for each of the hypotheses with $\alpha = 0.05$, there is a *greater* than 5% overall chance of a false positive finding in the study. If the multiple tests are independent, the probability that at least one of the null hypothesis tests will be erroneously rejected is

$$1 - Pr(\text{none of the null hypotheses will be erroneously rejected}) = 1 - (1 - \alpha)^M.$$

Therefore, if researchers are estimating effects on three outcomes (and if these outcomes are independent) the probability of at least one false positive finding is $1 - (1 - 0.05)^3 = 0.14$. If the researchers were instead estimating effects on five independent outcomes, the probability of at least one false positive finding rises to 0.23. This Type I error inflation for independent outcomes demonstrates the crux of the multiple testing problem. In practice, however, the multiple outcomes are usually at least somewhat correlated, which makes the test statistics correlated and reduces the extent of Type I error inflation. Nonetheless, any error inflation can still make it problematic to draw reliable conclusions about the existence of effects above a specified size" (Porter (2018)).

3.1 Using MTPs to protect against spurious impact findings

As introduced above, multiple testing procedures adjust p-values to counteract the multiple testing problem.[4] We next describe how using a MTP protects against false positives.

Considering multiple outcomes presents both challenges and opportunities. First, we discuss the impact of MTPs on individual power. The power of an individual hypothesis test is the probability of correctly rejecting a null hypothesis when the effect is at least a specified size. We refer to a setting in which the true impact is at least as large as the desired effect size as a "false null" hypothesis, while a "true null" is a setting in which the true impact is zero. In the proceeding explanations, when we refer to rejecting a null hypothesis or detecting an effect, we assume that we are detecting an effect of a certain pre-specified size. If p-values are adjusted upward, one is less likely to reject true nulls, which reduces the probability of Type I errors, or

[3] For a one-sided test, depending on the direction of our alternative hypothesis, the raw p-value for test m is computed as $p_m = Pr(T_m \leq t_m)$ or $p_m = Pr(T_m \geq t_m)$.

[4] Alternatively, MTPs can decrease the critical values for rejecting hypothesis tests. For ease of presentation, this paper focuses only on the approach of adjusting p-values.

false positive findings. At the same time, MTPs increase the probability of a Type II error, or false negative findings, when the test fails to reject a false null. Individual power is $1 - Pr(\text{Type II error})$, so MTPs have the tradeoff of reducing Type I errors but also reducing individual power.

Next, we consider the impact of multiple outcomes on other definitions of power. Applying a MTP reduces power according to all definitions of power relative to the case when no MTP is applied to adjust p-values. However, as discussed previously, having multiple outcomes also allows for a wider variety of definitions of success. Recall that 1−minimal power is the probability of detecting an effect on at least one outcome. Typically, 1−minimal power, even after applying a MTP, is higher than individual power for a hypothesis test on a single, pre-specified outcome. Depending on the study, other definitions of power, such as 1/2 or 1/3-minimal power, may or may not have higher power than the power of a single hypothesis test.

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 explain each of these features of MTPs and provide discussion of the new parameter specifications they require when estimating power.

3.2 Familywise error rate vs. false discovery error rate

Familywise procedures "reframe Type I error as a rate across the entire set or "family" of multiple hypothesis tests. This rate is called the familywise error rate (FWER; Tukey (1953)). The FWER is typically set to the same value as the probability of a Type I error for a single test, e.g., α. MTPs that control the FWER at 5% adjust p-values in a way that ensures that the probability of at least one Type I error across the entire set of hypothesis tests is no more than 5%. The MTPs introduced by Bonferroni (Dunn (1959), Dunn (1961)), Holm (1979), and Westfall and Young (1993) all control the FWER" (Porter (2018)).

MTPs that control false discovery rate (FDR) take an entirely different approach to the multiple testing problem. FDR, introduced by Benjamini and Hochberg (1995), is a less stringent criteria than FWER. It is the expected proportion of all rejected hypotheses that are erroneously rejected. As laid out in Porter (2018), the two-by-two representation in Table 1 is often found in articles on multiple hypothesis testing, and helps to illustrate the difference between FWER and FDR. Let M be the total number of tests. Therefore, we have M unobserved truths: whether or not each null hypothesis is true or false. We also have M observed decisions: whether or not the null hypotheses were rejected, because the p-values were less than α. In Table 1, U, V, W, and X are four possible scenarios: the numbers of true or false hypotheses not rejected or rejected. M_0 and M_1 are the unobservable numbers of true null and false null hypotheses. R is the number of null hypotheses that were rejected, and $M - R$ is the number of null hypotheses that were not rejected.

	Observed decisions		
Unobserved truths	Number not rejected	Number rejected	Total
Number of true null hypotheses	U	V	M_0
Number of false null hypotheses	W	X	M_1
Total	$M - R$	R	M

Table 1: Numbers of hypothesis types and decisions.

In Table 1, V is the number of erroneously rejected null hypotheses, or the number of false positive findings. Therefore, the FWER is equivalent to $Pr(V > 0)$, the probability of at least one false positive finding. As noted in Porter (2018), "recall that Type I error is inflated when testing for effects when no MTPs are applied. Consider the setting when all the outcomes are independent of each other. The Type I error is almost 10% when testing effects on two independent outcomes and 23% when testing effects on five independent outcomes. These Type I error rates both correspond to the FWER. The goal of MTPs that control the FWER is to bring these percentages back down to 5%."

Also in Table 1, the FDR is equal to $E(V/R)$ but is defined to be 0 when $R = 0$, or when no hypotheses are rejected. "As is frequently noted in the literature (e.g., Shaffer (1995); Schochet (2008)), the FWER and FDR have different objectives. Control of the FWER protects researchers from spurious findings and so may be preferred when even a single false positive could lead to the wrong conclusion about the effectiveness of an intervention. On the other hand, the FDR is more lenient with false positives" (Porter (2018)). Researchers may be willing to accept a few false positives, V, when the total number of rejected hypotheses, R, is large. Note that under the complete null hypothesis that all M null hypotheses are true, the FDR is equal to the FWER. Referring back to Table 1, under the complete null we have $V = R$, so

$$\begin{aligned} FDR &= E\left(\frac{V}{R}\right) \\ &= E\left(\frac{V}{R} \mid R = 0\right) Pr(R = 0) + E\left(\frac{V}{R} \mid R > 0\right) Pr(R > 0) \\ &= 0 \times Pr(R = 0) + 1 \times Pr(R > 0) \\ &= Pr(R > 0) = FWER \end{aligned}$$

However, if any effects truly exist, then FWER $\geq$ FDR. As a result of the difference in objective between FWER and FDR, in the case where there is at least one false null hypothesis (at least one true effect), a MTP that controls the FDR at 5% will have a Type I error rate that is greater than 5%.

A side remark is that MTPs may provide either "weak control" or "strong control" of the error rate they target. A MTP "provides weak control of the FWER or the FDR at level α 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 α if the control is guaranteed 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)).[5]

3.3 Single-step vs. stepwise procedures

An additional feature of a MTP that affects its statistical power is whether it is a "single-step" or "stepwise" procedure. "Single-step procedures adjust each p-value independently of the other p-values. For example, the Bonferroni MTP multiplies all raw p-values by M. Therefore, one p-value adjustment does not depend on other p-value adjustments, only on the number of tests" (Porter (2018)).

In contrast, "stepwise procedures first order raw p-values (or test statistics), and then adjust according to the order of the tests. The adjustments depend on the null hypotheses already rejected in previous steps. For example, the Holm MTP — the stepwise counterpart to the Bonferroni MTP — orders raw p-values from smallest to largest. The procedure then multiplies the smallest p-value by M, the second smallest p-value by $M - 1$, and so on. The Holm MTP, like most other stepwise procedures, also enforces monotonicity: each adjusted p-value is greater than or equal to the previous adjusted p-value, and enforces that any p-values is not greater than one. Overall, stepwise MTPs allow for less adjustment than single-step MTPs in later steps, and therefore preserve more power (for outcomes in the later steps). The Bonferroni and Westfall-Young single-step procedure are single-step; the Holm and Benjamini-Hochberg MTPs and the Westfall-Young step-down procedure are stepwise" (Porter (2018)). 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,

[5]The single-step and step-down Westfall Young MTPs (which we discuss below) 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, Dudoit, and Speed (2003)).

equivalently, the number of false null hypotheses. "Researchers may be inclined to assume that there will be effects on all outcomes, as hypotheses of effects probably drive the selection of outcomes in the first place.(...) However, if the researchers are incorrect, the probability of detecting the extant effects can be diminished, sometimes substantially" (Porter (2018)).

As noted in Porter (2018), it is important to point out that 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.) and complete power. While individual power is defined based on the probability of correctly rejecting false nulls, the definition is loosened here and includes the probability of erroneous rejections of true nulls. 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 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 if we assume all five outcomes are false nulls, or if 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.

There is an additional technical note about the calculation of complete power.[6] 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 α, the intersection-union test is also a level α 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 if 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 nulls 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 α. For example, if we have two independent tests with Type I error α, then if both of are true nulls, the probability of a Type I error for the omnibus test (the probability of rejecting both null hypotheses) is α^2 (Deng, Xu, and Wang (2008)).

3.4 Correlation between test statistics

The final feature of a MTP that affects its statistical power is whether or not it takes into account the correlation between test statistics. "The Bonferroni and Holm procedures strongly control the FWER in all cases, even when the test statistics are correlated, but they adjust p-values more than is necessary in that case. Along with the Bonferroni and Holm MTPs, the Benjamin-Hochberg MTP also does not take correlations into account.[7] In contrast, both of the Westfall-Young MTPs rely on the estimation of the joint distribution of test statistics when the complete null hypothesis (that there are not effects on any of the outcomes) is true. This joint distribution of the test statistics is estimated from the study's data" (Porter (2018)). For example, random permutations of the treatment indicator break the association between treatment status and outcome. Repeating these permutations a large number of times results in a distribution of test statistics under the complete null. "Because the actual data are used to generate this null distribution, correlations

[6] Complete power has also been referred to as "conjunctive power" (Bretz, Hothorn, and Westfall (2010)) and "all pairs power" (Ramsey (1978)).

[7] The Benjamini-Hochberg procedure was originally shown to control the FDR for independent test statistics. However, Benjamini and Yekutieli (2001) showed that it also controls the FDR for true null hypotheses with "positive regression dependence." This condition is satisfied for most applications in practice.

among the test statistics are captured. Then observed test statistics can be compared with the distribution of test statistics under the complete null hypothesis" (Porter (2018)).[8]

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. 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)). Then, "if there are no covariates in the impact models or if the R^2's of the covariates are equivalent in all impact models, then the correlations between the test statistics are equal to the correlations between the outcomes. However, having different R^2's across the impact models reduces the correlations between the residuals and therefore between test statistics.[9] Models of outcomes that are highly correlated are more likely to have residuals that are highly correlated because baseline covariates will tend to have similar R^2's. The gaps between the correlations between outcomes and the correlations between residuals — and therefore the test statistics — may be wider for moderately or weakly correlated outcomes. In any case, the upper bounds of correlations between the test statistics are the correlations between the outcomes" (Porter (2018)).

4 Estimating power, MDES and sample size in studies with multiple outcomes

4.1 Power estimation approach

We take an innovative simulation-based approach to estimating power, as introduced in Porter (2018). This approach is then also applied to estimate MDES and sample size. In order to estimate power 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 quite 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 to calculate estimated power. 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 far 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 size. We define ψ_m to be the treatment effect for outcome m, with M total outcomes. 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: $H_A : \psi_1 > 0.125, \psi_2 > 0.2, \psi_3 > 0.1, \psi_4 > 0.1, \psi_5 > 0.05$. Another possible joint alternative hypothesis is one where only the first two outcomes have effects above the desired sizes: $H_A : \psi_1 > 0.125, \psi_2 > 0.2, \psi_3 = \psi_4 = \psi_5 = 0$. Once our joint alternative hypothesis is specified, we would generate simulated data under this hypothesis. To simulate data, we need to specify the full set of parameters as mentioned in Section 5.3 that allow for data generation. The Technical Appendix contains more details about the assumed data-generating process. For example, for the Diplomas Now experiment, we would assume a specific data generating process to allow us to simulate synthetic students, schools, and districts, including covariates, outcomes, and treatment assignment. This process would involve specifying parameter values such as R^2, the amount of outcome

[8] Instead of using test statistics, the Westfall-Young MTPs can alternatively compare raw p-values with the estimated joint null distribution of p-values.

[9] For example, one of the multiple outcomes may have a baseline covariate with a high R^2 while another may have a baseline covariate with a smaller R^2. Also, block dummies may explain more variation in some outcomes than in others.

variation explained by covariates at a particular level, and 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` library (Bates et al. (2015)), and `interacted_linear_estimators()` from the `blkvar` library.[10] 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 we could run a 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 $\mathbf{F}$, and is of dimension $tnum \times M$. One row corresponds to one set of simulated raw p-values from regressions for the 5 attendance outcomes of interest for Diplomas Now.

5. *Adjust p-values.* For each row, corresponding to one simulated dataset, 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 $\mathbf{G}$ of adjusted p-values. 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 procedures. One row corresponds to one set of simulated *adjusted* p-values for the 5 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 α (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 $tnum \times M$). 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 `tnum` rows in which the null hypothesis m was rejected (the mean of column m of $\mathbf{H}$). We would have 5 different individual power values for Diplomas Now, corresponding to each outcome of interest.

- *d-minimal power* is the proportion of the $tnum$ rows in which at least d of the M hypotheses were rejected.[11] For Diplomas Now, we could consider 1-minimal power through 4-minimal power.

- *Complete power* is the proportion of the $tnum$ 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 $\mathbf{H}$.) 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 5 outcomes, this criteria is a relatively strict indicator of success.

Above, we outlined a full simulation-based approach for calculating power. This approach would be computationally intensive because of the need to generate and analyze a full simulated dataset at each iteration. We can simplify this process by skipping the simulation of data and modeling steps. Given an assumed model and correlation structure for the test statistics, we can directly sample from $f(t_1, \ldots, t_M)$, 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:

[10] This package is currently under development on GitHub; see https://github.com/lmiratrix/blkvar

[11] Note that others refer to 1−minimal power simply as "minimal power" (e.g., Maurer and Mellein (1988); Chen et al. (2011); Westfall, Tobias, and Wolfinger (2011)), "disjunctive power" (e.g., 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 $d > 1$.

1. **Generate** draws of *test statistics* $t_1, \ldots, t_M$ *under the joint alternative hypothesis.* This step produces a $tnum \times M$ 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.*

We now describe how to sample from $f(t_1, \ldots, t_M)$ directly. First, we assume a particular research design and model. In our example based on the Diplomas Now study, the research design is a 3-level experiment, with randomization at level 2. We plan for analyzing our data with a linear regression model with fixed intercepts at the district level, random intercepts at the school level, and a constant treatment effect across schools and districts. As previously, denote ψ_m as the treatment effect for outcome m. We express treatment effects in terms of effect sizes:

$$ES_m = \frac{\psi_m}{\sigma_m}$$

where σ_m is the standard deviation of outcome Y_m in the control group. In order to calculate power, we also need the standard error of the impact in effect size units, which we denote as

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

The quantity Q_m is a consequence of the assumed model, the number of units at different levels, the percent of units treated, the assumed R^2, and other 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 districts; 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 districts; and the amount of impact variation relative to the amount of mean variation. Some parameters, such as the percent 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 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:

$$t_m = \frac{\hat{ES}_m}{\hat{Q}_m}$$

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

Note that 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 dataset under the joint alternative hypothesis and calculate a set of M observed test statistics. Then, we would permute the single simulated dataset, say $B = 3,000$ times, assuming the joint null hypothesis, and calculate test statistics on each of these permuted datasets. 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 dataset, and repeat the process. If we were to generate $tnum = 10,000$ datasets under the joint alternative hypothesis, for each of these datasets we generate $B = 3,000$ permuted datasets under the joint null, so we would have to generate $10,000 \times 3,000$ datasets!

When we skip the step of simulating data, then for each iteration t in $1, \ldots, tnum$ 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 0. 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. Westfall-Young procedures are computationally intensive, so the approach of skipping the simulated data step is particularly helpful here. This approach substantially reduces computational time by drawing test statistics directly rather than permutating the data.

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, 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_mdes()` and `pump_sample()`, the user provides a particular target power, say 80%. The method then conducts 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

$$MDES = MT_{df} \times SE/\sigma_m$$

where MT_{df}, the "multiplier," is the sum of two t statistics with degrees of freedom df. For one-tailed tests, $MT_{df} = t^{\star}_{\alpha} + t^{\star}_{1-\beta}$ where α is the Type I error rate and β is the desired power. For two-tailed tests, $MT_{df} = t^{\star}_{\alpha/2} + t^{\star}_{1-\beta}$. 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 bounds by manipulating the α and β values in the above. First, to calculate the preliminary lower bound, we apply the formula above as given, assuming individual unadjusted power will give the smallest MDES; to calculate the preliminary upper bound, we apply the formula using α/M to correspond to a Bonferroni correction. We also adjust β 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; 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, including the correlation between test statistics, 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 beyond the initial bounds for the next step. With each iteration we increase `tnum` to increase

precision as we narrow in on the final answer; 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 the 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 corner 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 on CRAN.

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 datasets according to the assumed design and model, calculating p-values, and calculating an empirical estimate of power. Using a binomial distribution we constructed Monte Carlo confidence intervals for the power estimates 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 the Appendix, and full validation code and results are in a supplementary github repository pump_validate. For some scenarios, we have some apparent discrepancies from PowerUp, 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. These discrepancies are noted in the appendix.

5 User choices

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 chooses an assumed model, including whether intercepts and treatment effects 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 homogenous 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.

d_m	Design	Model	PowerUp	Params
d1.1_m1c	d1.1	m1c	n/a	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	n/a	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	n/a	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

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 treatment impacts:

- Whether level two and level three intercepts are:
  - fixed: we have a separate intercept for each unit.
  - random: 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 effects are:
  - constant: we model all units within a group as having the same single average impact.
  - fixed: 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: we allow variation as with fixed, but model the collection of treatment impacts as Normally distributed around a grand mean mean impact. This is implicitly allowing for the sample as being representative of a larger super-population, in terms of treatment impact estimation.

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, either f or r and then the assumption for the treatment impacts, c, f, or r. For example, `m3ff2rc` means at level 3, 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. `d2.1_m3fc`. The Diplomas Now model, for example, is `d3.2_m3fc2rc`.

The full list of supported design and model combinations is below. The user can see the list by calling `pump_info()`, which provides the designs and models, MTPs, power definitions, and model parameters. We also include the corresponding names from the PowerUP! package where appropriate. For more details about each combination of design and model, see the Technical Appendix.

5.2 Multiple testing procedures

The supported multiple testing procedures were covered in more detail in Section 3. Here we provide a review of the multiple testing procedures supported by the `PUMP` package:

- *Bonferroni*: 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*: 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*: 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 α.

- *Single-step Westfall-Young*: 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*: 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.

For a more detailed explanation of each MTP, see Appendix A of Porter (2018).

The following table from Porter (2018) summarizes the important features for each of the MTPs supported by `PUMP`.

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 2: Summary of MTP procedures.

5.3 Model parameters

The table below shows the parameters that influence Q_m, the standard error, for different designs and models.

A few parameters warrant more explanation.

- The quantity ICC is the unconditional Intraclass Correlation, and gives a measure of variation at different levels of the model. For each outcome, 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.
- For each outcome, the quantity omega (ω) for each level is the ratio between impact variation at that level and variation in intercepts (including covariates) at that level. It is a measure of treatment impact heterogeneity.
- The R^2 expressions are the percent of variation at a particular level predicted by covariates specific to that level. For simplicity we assume covariates at a level are group mean centered, so only covariates at a particular level explain variance at that level.

For precise formulae of these expressions, see the Technical Appendix, which outlines the assumed data-generating process, and the resulting expressions for ICC, ω, and R^2.

In addition to design parameters, there are additional parameters that control the precision 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. Note that the `pump_mdes()` and `pump_sample()` have multiple `tnum` parameters controlling the precision of the search.
- `B` is the number of Westfall-Young permutations. Again, there is a tradeoff 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. The default of `1` does not result in parallel computation. Parallelization is done using `parApply` from the `parallel` package.

Parameter	Description
nbar	harmonic mean of level 1 units per level 2 unit (students per school)
J	harmonic mean of number of level 2 units per level 3 unit (schools per district)
K	number of level 3 units (districts)
Tbar	proportion of units assigned to treatment
numCovar.1	number of level 1 (individual) covariates
numCovar.2	number of level 2 (school) covariates
numCovar.3	number of level 3 (district) covariates
R2.1	percent of variation explained by level 1 covariates
R2.2	percent of variation explained by level 2 covariates
R2.3	percent of variation explained by level 3 covariates
ICC.2	level 2 intraclass correlation
ICC.3	level 3 intraclass correlation
omega.2	ratio of variance of level 2 average impacts to level 2 random intercepts
omega.3	ratio of variance of level 3 average impacts to level 3 random intercepts

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 rho, the correlation of the test statistics.

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.

We next discuss selection of all needed design parameters and modeling choices. 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 a few summarizing elements: the individual-level standard errors, the degrees of freedom, and the correlation structure of the test statistics. Once we have these elements, regardless of the design, we can proceed.

Analytic model. We first 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"). We note that 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.

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 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. 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).

ICCs. We also need a measure of where variation occurs: the individual, the school, or the randomization block level. We capture this with Intraclass Correlation Coefficients (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 0.40. We set a relatively high level three ICC as we expect our school type by district blocks to isolate variation; in particular we might believe middle and high school attendance rates would be markedly different.

Correlation of outcomes. We finally need to specify the number 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 block. We believe the attendance measures would be fairly related, so we select `rho = 0.40` 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. 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^2s 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 ρs rather than a single ρ; 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 effects? To answer this question we calculate the minimal 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:

```
m <- pump_mdes(
  d_m = "d3.2_m3fc2rc",       # choice of design and analysis strategy
  MTP = "HO",                  # multiple testing procedure
  target.power = 0.80,           # desired power
  power.definition = "D1indiv", # power type
  M = 5,                         # number of outcomes
  J = 3,                         # number of schools per block
  K = 21,                        # number districts
  nbar = 258,                    # average number of students per school
  Tbar = 0.50,                   # prop treated
  alpha = 0.05,                  # significance level
  numCovar.1 = 5,                # number of covariates at level 1
  numCovar.2 = 3,                # number of covariates at level 2
  R2.1 = 0.1, R2.2 = 0.7,        # explanatory power of covariates for each level
  ICC.2 = 0.05, ICC.3 = 0.4,     # intraclass correlation coefficients
  rho = 0.4 )                    # how correlated outcomes are
```

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

```
knitr::kable( m, digits = 3, booktabs = TRUE,
              position = "h!", caption = "MDES Estimate" ) %>%
  kableExtra::kable_styling( position = "center" )
```

Table 3: MDES Estimate

MTP	Adjusted.MDES	D1indiv.power
HO	0.104	0.792

The answers `pump_mdes()` gives are approximate as we are calculating them via monte carlo 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 tolerance 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 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:

```
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    0.08188259      0.804 0.002363978
#>  (13 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.[12] 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:

```
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    0.08913163      0.791 0.002553031
#>  (16 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. Therefore, an increased MDES (for the nonzero outcomes) is required to achieve the same level of desired power (80%). Below we provide a deeper dive into the extent to which `numZero` can effect 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, or the number of blocks to detect an effect of a given size. The `pump_sample` method will search over any one of these.

Assuming we have three schools per block, we first 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 do 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).

```
smp <- pump_sample(
  d_m = "d3.2_m3fc2rc",
  MTP = "HO",
  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
#>   HO           K          15    0.80025 0.01
#>  (13 steps in search)
```

We would need 15 blocks, rather than the originally specified 21, giving 45 total schools in our study, to achieve 80% 1−minimal power.

We recommend checking MDES and sample-size calculators, as the estimation error combined with the stochastic search can give results a bit off the target 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

[12] 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.

originally targeted 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:

```
p_check <- update( smp, type = "power", tnum = 100000,
                   long.table = TRUE )
```

Table 4: Power table

power	None	HO
individual outcome 1	0.7	0.52
individual outcome 2	0.7	0.52
individual outcome 3	0.7	0.52
individual outcome 4	0.7	0.53
individual outcome 5	0.7	0.53
mean individual	0.7	0.52
1-minimum		0.81
2-minimum		0.64
3-minimum		0.50
4-minimum		0.39
complete		0.32

When calculating power directly, we get power for all the implemented definitions of power applicable to the design.

In the above, the first five rows are the powers for 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 column is power without adjustment, and the second has our power with the listed p-value adjustment.

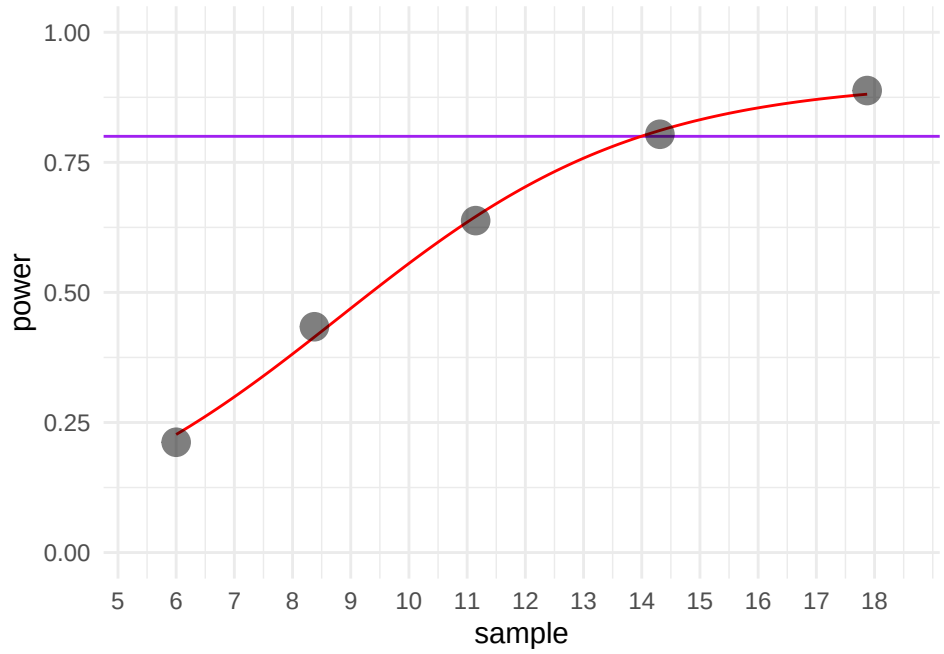
The next rows show different multi-outcome definitions of power. In particular, `1-minimum` shows the chance of rejecting at least one hypotheses. The `complete` row shows the power to reject all hypotheses; it is only defined if all outcomes are specified to have a non-zero effect.[13]

We can look at a power curve of our `pump_sample()` call to assess how sensitive power is to our level two sample size:[14]

```
plot_power_curve( smp )
```

[13]The package does not show power for these without adjustment for multiple testing, as that power would be grossly inflated and meaningless.

[14]The points on the plots show the evaluated simulation trials, with larger points corresponding to more iterations and greater precision.

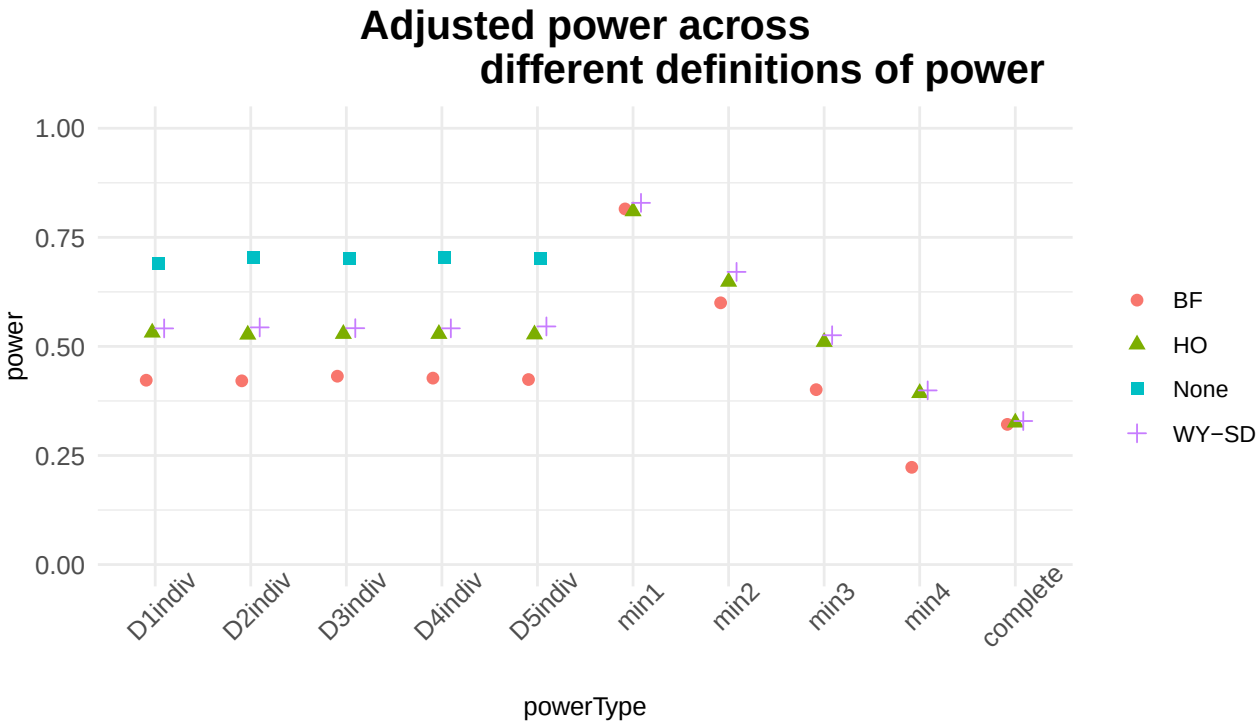


Remark. In certain settings, a wide range of sample sizes may result in very similar levels of power. In this case, the algorithm may return a sample size that is larger than necessary. This pattern does not occur for the sample size at the highest level of the hierarchy, and only for occurs for sample sizes at lower levels of the hierarchy; e.g. for `nbar` for all models, and for `nbar` and `J` for three level models. In addition, due to the nature of the search algorithm, occasionally the algorithm may not converge. 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 using the Westfall-Young Stepdown procedure (this procedure is much more computationally intensive to run), or other procedures of interest. Alternatively, simply provide a list of procedures you wish to compare. If you provide a list, the package will re-run the power calculator for each item on the list; this can make the overall call computationally intensive. Here we obtain power for our scenario using Bonferroni, Holm and Westfall-Young adjustments, and plot the results using the default `plot()` method:

```
p2 <- update( p_check,
              MTP = c( "BF", "HO", "WY-SD" ),
              tnum = 10000,
              parallel.WY.cores = 2 )
plot( p2 )
```



To speed up computation, we set `parallel.WY.cores = 2` to parallelize the computation. We also reduce `tnum` to decrease computation time.

The more sophisticated (and less conservative) adjustment exploits the correlation in our outcomes (`rho = 0.4`) to provide higher individual power. Note, however, that we do not see elevated rates for 1−minimal power. Accounting for the correlation of the test statistics when adjusting p-values can drive some power (individual power) up, but on the flip side 1−power can be driven down as the lack of independence between tests gives fewer chances for a significant result. See Porter (2018) for further discussion; 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.

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, `update()` allows for different values of a parameter for the different outcomes; the `grid` approach, by contrast, is more limited in this regard, and assumes the same parameter value across different outcomes. Second, the `grid` functions are a powerful tool for systematically exploring many possible combinations, while `update()` only allows the user to explore one value at a time.

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

6.5.1 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:

```
p_b <- update( p_check, ICC.2 = 0.20, ICC.3 = 0.25 )
print( p_b )
#> power result: d3.2_m3fc2rc d_m with 5 outcomes
#>   MTP D1indiv D2indiv D3indiv D4indiv D5indiv indiv.mean    min1    min2
#>  None 0.24513 0.24236 0.24333 0.24256 0.24491   0.243658      NA      NA
#>    HO 0.09800 0.09737 0.09711 0.09704 0.09799   0.097502 0.26677 0.11533
#>     min3    min4 complete
#>       NA      NA       NA
#>  0.05815 0.03112  0.02399
#>  0.000 <= SE <= 0.001
```

We immediately see that our assumption of substantial variation in level three matters 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:

```
p_d <- update( p_check,
               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 ) )
print( p_d )
#> power result: d3.2_m3fc2rc d_m with 5 outcomes
#>   MTP D1indiv D2indiv D3indiv D4indiv D5indiv indiv.mean    min1    min2
#>  None 0.43265 0.85325 0.38075 0.34358 0.34337   0.470720      NA      NA
#>    HO 0.24398 0.65388 0.21066 0.18851 0.18856   0.297118 0.71219 0.37367
#>     min3    min4 complete
#>       NA      NA       NA
#>  0.21062 0.12208  0.08596
#>  0.000 <= SE <= 0.001
```

Notice how the individual powers are heavily impacted. 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 `update()`s, we may lose track of where we are; to find out, we can always check details with `print_design()` or `summary()`:

```
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: 15   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
#>  None 0.43265 0.85325 0.38075 0.34358 0.34337   0.470720      NA      NA
#>    HO 0.24398 0.65388 0.21066 0.18851 0.18856   0.297118 0.71219 0.37367
#>     min3    min4 complete
#>       NA      NA       NA
```

```
#>  0.21062 0.12208  0.08596
#>  0.000 <= SE <= 0.001
#>  (tnum = 100000)
```

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.

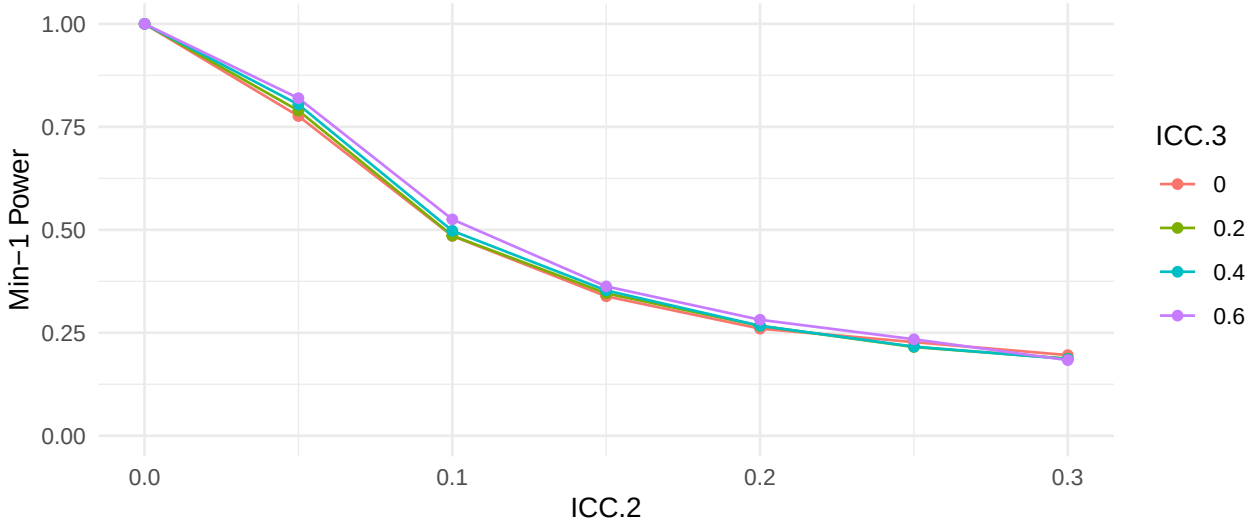
6.5.2 Exploring the impact of the ICC

We above saw that the ICC does impact 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:

```
grid <- update_grid( p_check,
            ICC.2 = seq( 0, 0.3, 0.05 ),
            ICC.3 = seq( 0, 0.60, 0.20 ),
            tnum = 5000 )

grid$ICC.3 <- as.factor( grid$ICC.3 )
grid <- dplyr::filter( grid, MTP == "HO" )

ggplot2::ggplot( grid, aes( ICC.2, min1, group = ICC.3, col = ICC.3 ) ) +
    geom_line() + geom_point() +
    labs( y = "Min-1 Power" ) +
  coord_cartesian( ylim = c(0,1) )
```



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

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:

$$SE(\hat{\tau}) = \sqrt{\frac{\text{ICC}_2(1-R_2^2)}{\bar{T}(1-\bar{T})JK} + \frac{(1-\text{ICC}_2-\text{ICC}_3)(1-R_1^2)}{\bar{T}(1-\bar{T})JK\bar{n}}}.$$

In the above, the $\bar{n} = 258$ students per group makes the second term very small compared to the first, regardless of the ICC.3 value. The first term, however, is a direct scaling of ICC.2; changing it will change

the standard error, and therefore power, a lot. All provided designs and models implemented in the package are discussed, along with corresponding formula such as these, in our technical supplement accompanying this paper and package.

For grid searches we recommend reducing the number of permutations, 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.

The `grid()` functions provide easy and direct ways of exploring how power changes as a function of the design parameters. We note, however, that in order to keep syntax simple, they do not allow different design parameters, including MDES, by outcome. This is to keep package syntax simpler. 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.

6.5.3 Exploring the impact of rho

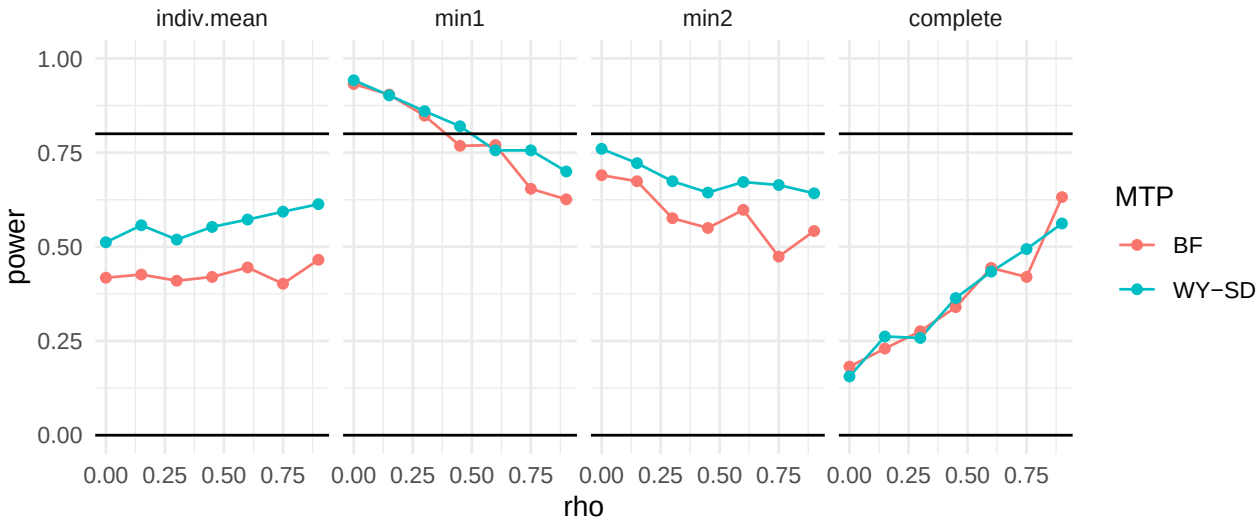
The correlation of test statistics, ρ, is a critical parameter for how power will play out across the multiple tests. For example, with Westfall-Young, we saw that the correlation can improve our individual power, as compared to Bonferroni. We might not know what will happen to 2−minimal power, however: 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 question relatively easily by letting `rho` vary as so:

```
gridRho <- update_grid( p_check,
              MTP = c( "BF", "WY-SD" ),
              rho = seq( 0, 0.9, by = 0.15 ),
              tnum = 500,
              B = 10000 )
```

We then plot our results.

```
gridL <- dplyr::filter( gridRho, MTP != "None" ) %>%
  tidyr::pivot_longer( cols = c(indiv.mean, min1, min2, complete),
                       names_to = "definition", values_to = "power" ) %>%
  dplyr::mutate( definition = factor( definition,
          levels = c("indiv.mean", "min1", "min2", "complete" ) ) )

ggplot2::ggplot( gridL, aes( rho, power, col = MTP ) ) +
  facet_grid( . ~ definition ) +
  geom_line() + geom_point() +
  geom_hline( yintercept = c( 0, 0.80 ) ) +
  theme_minimal() +
  coord_cartesian( ylim = c( 0, 1 ) )
```



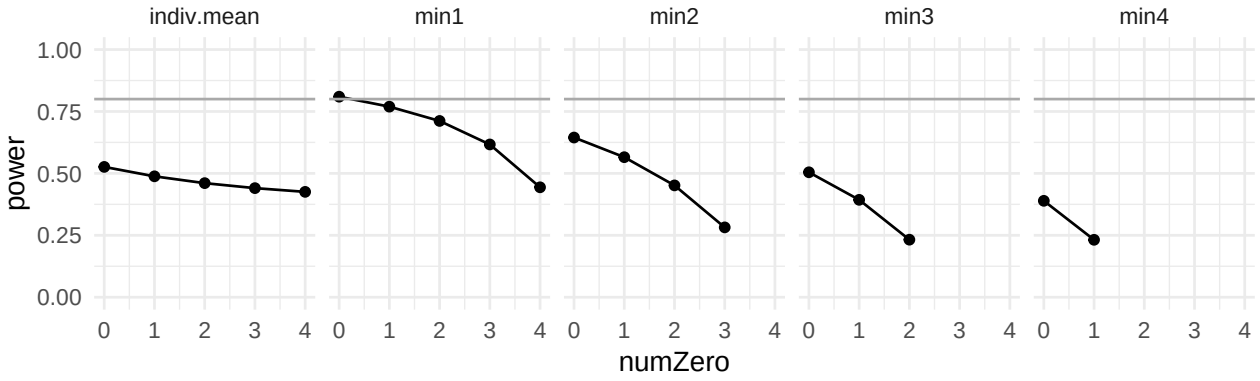
First, we see the benefit of the Westfall-Young single-step procedure is minimal, as compared to Bonferroni. Second, the impact on individual adjustment is flat, as anticipated. Third, across a very broad range of rho, we maintain good 1−minimal power. Complete power climbs as correlation increases, and 2−minimal power is generally unchanged.

6.5.4 Exploring the impact 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:

```
gridZero <- update_grid( p_check,
                         numZero = 0:4,
                        M = 5 )
```

We then can make a plot as we did above:



There are other ways of exploring the impact 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.)

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. An online interface is also available at https://mdrc.shinyapps.io/pump/.

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 impact of different designs, models, and 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 code is available on github, https://github.com/MDRCNY/PUMP.
- The Technical Appendix contains detailed information about each design and model, the assumed data generating process, and understanding parameters such as ICC and ω. It is a useful reference not just for users of the package, but also as a general summary of multi-level models.
- The package has an additional vignette on understanding sample size calculations, which present unique challenges.
- 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.

Acknowledgements

We acknowledge the Diplomas Now team at MDRC. Development of this package was supported by a grant from the Institute of Education Sciences (R305D170030). We would like to thank members of the Harvard CARES lab for their feedback on the manuscript.

8 References

Bang, H., S. Jung, and S. L. George. 2005. "Sample Size Calculation for Simulation-Based Mulitple-Testing Procedures." *Journal of Biopharmaceutical Statistics* 15: 957–67.

Bates, D., M. Mächler, B. Bolker, and S. Walker. 2015. "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software* 67 (1): 1–48.

Benjamini, Y., and Y. Hochberg. 1995. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society B* 57: 289–300.

Benjamini, Y., and D. Yekutieli. 2001. "The Control of the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing Under Dependency." *The Annals of Statistics* 29: 1165–88.

Berger, Roger L. 1982. "Multiparameter Hypothesis Testing and Acceptance Sampling." *Technometrics* 24 (4): 295–300.

Berger, Roger L., and Jason C. Hsu. 1996. "Bioequivalence Trials, Intersection–Union Tests and Equivalence Confidence Sets." *Statistical Science* 11 (4): 283–319.

Bloom, Howard S. 2006. "The Core Analytics of Randomized Experiments for Social Research." MDRC.

Bretz, F, T. Hothorn, and P. Westfall. 2010. *Multiple Comparisons Using R*. Chapman & Hall/CRC.

Chen, J., J. Luo, K. Liu, and D. Mehrotra. 2011. "On Power and Sample Size Computation for Multiple Testing Procedures." *Computational Statistics and Data Analysis* 55: 110–22.

Corrin, W., S. Sepanik, R. Rosen, and A. Shane. 2016. "Addressing Early Warning Indicators: Interim Impact Findings from the Investing in Innovation (I3) Evaluation of Diplomas Now." MDRC.

Deng, Xutao, Jun Xu, and Charles Wang. 2008. "Improving the Power for Detecting Overlapping Genes from Multiple DNA Microarray-Derived Gene Lists." *BMC Bioinformatics* 9.

Dong, Nianbo, and Rebecca Maynard. 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.

Dudoit, S., J. P. Shaffer, and J. C. Boldrick. 2003. "Multiple Hypothesis Testing in Microarray Experiments." *Statistical Science* 18 (1): 71–103.

Dunn, Olive Jean. 1959. "Estimation of the Medians for Dependent Variables." *The Annals of Mathematical Statistics* 30 (1): 192–97.

———. 1961. "Multiple Comparisons Among Means." *Journal of the American Statistical Association* 56 (293): 52–64.

Ge, Y., S. Dudoit, and T. P. Speed. 2003. "Resampling-Based Multiple Testing for Microarray Data Analsysis." *Test* 12: 1–77.

Gelman, A., J. Hill, and M. Yajima. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press.

———. 2012. "Why We (Usually) Don't Have to Worry about Multiple Comparisons." *Journal of Research on Educational Effectiveness* 5: 189–211.

Hedges, Larry V., and Christopher Rhoads. 2010. "Statistical Power Analysis in Education Research." National Center for Special Education Research. http://www.eric.ed.gov/ERICWebPortal/detail?accno=ED509387.

Holm, S. 1979. "A Simple Sequentially Rejective Multiple Test Procedure." *Scand. J. Statist.* 6 (2): 65–70.

Maurer, W., and B. Mellein. 1988. "On New Multiple Test Procedures Based on Independent P-values and the Assessment of Their Powers." In *Multiple Hypotheses Testing*, edited by P. Bauer, G. Hommel, and E. Sonnermann, 48–66. Springer-Verlag.

Porter, Kristin E. 2018. "Statistical Power in Evaluations That Investigate Effects on Multiple Outcomes: A Guide for Researchers." *Journal of Research on Educational Effectiveness* 11: 267–95.

Ramsey, P. H. 1978. "Power Differences Between Pairwise Multiple Comparisons." *Journal of American Statistical Association* 75: 479–87.

Raudenbush, S. W., J. Spybrook, R. Congdon, X. Liu, A. Martinez, H. Bloom, and C Hill. 2011. "Optimal Design Plus Empirical Evidence (Version 3.0)." http://wtgrantfoundation.org/resource/optimal-design-with-empirical-information-od.

Schochet, Peter Z. 2008. "Guidelines for Multiple Testing in Impact Evaluations of Educational Interventions. Final Report." Mathematica Policy Research, Inc. P.O. Box 2393, Princeton, NJ 08543-2393. http://www.eric.ed.gov/ERICWebPortal/contentdelivery/servlet/ERICServlet?accno=ED502199.

Senn, Stephen, and Frank Bretz. 2007. "Power and Sample Size When Multiple Endpoints Are Considered." *Pharmaceutical Statistics* 6: 161–70. https://doi.org/10.1002/pst.301.

Shaffer, Juliet Popper. 1995. "Multiple Hypothesis Testing." *Annual Review of Psychology* 46 (1): 561–84.

Spybrook, Jessica, H. S. Bloom, Richard Congdon, Carolyn J. Hill, Andres Martinez, and Stephen W. Raudenbush. 2011. "Optimal Design Plus Empirical Evidence: Documentation for the 'Optimal Design' Software Version 3.0." http://wtgrantfoundation.org/resource/optimal-design-with-empirical-information-od.

Tukey, J. W. 1953. "The Problem of Multiple Comparisions." Princeton University.

Westfall, Peter H, R. D. Tobias, and R. D. Wolfinger. 2011. *Multiple Comparisons and Multiple Tests Using SAS*. The SAS Institute.

Westfall, Peter H, and S Stanley Young. 1993. *Resampling-Based Multiple Testing: Examples and Methods for P-value Adjustment*. Vol. 279. John Wiley & Sons.

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

Table 5: Validation parameters

9 Appendix: Validation

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

- `PUMP`
- `PowerUpR`
- Monte Carlo simulations

We compute values from `PowerUpR` for individual, unadjusted power, as `PowerUpR` only provides power estimates in that setting. For all other 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. After repeatedly simulating data and calculating p-values, we calculate power and a 95% confidence interval, assuming a conservative standard error estimate of $\sqrt{0.25/S}$.

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 `PowerUpR` for certain models. For details about differences between `PUMP` and `PowerUpR` assumptions, see the Technical Appendixs. 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.

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 ρ, 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 9 shows the default parameter values, and the other values we try to test out varying that parameter.

We do not vary:

- `M = 3`
- `J` and `K` are fixed for each scenario
- Scalar grand mean of control outcomes
- Correlations between school random effects and impacts
- `rho` informs all correlations; we keep the same correlation between covariates, residuals, impacts, random effects for all levels and across all outcomes.

Validation results

Figure 9 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 red point to be within the blue intervals. Figure 9 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.

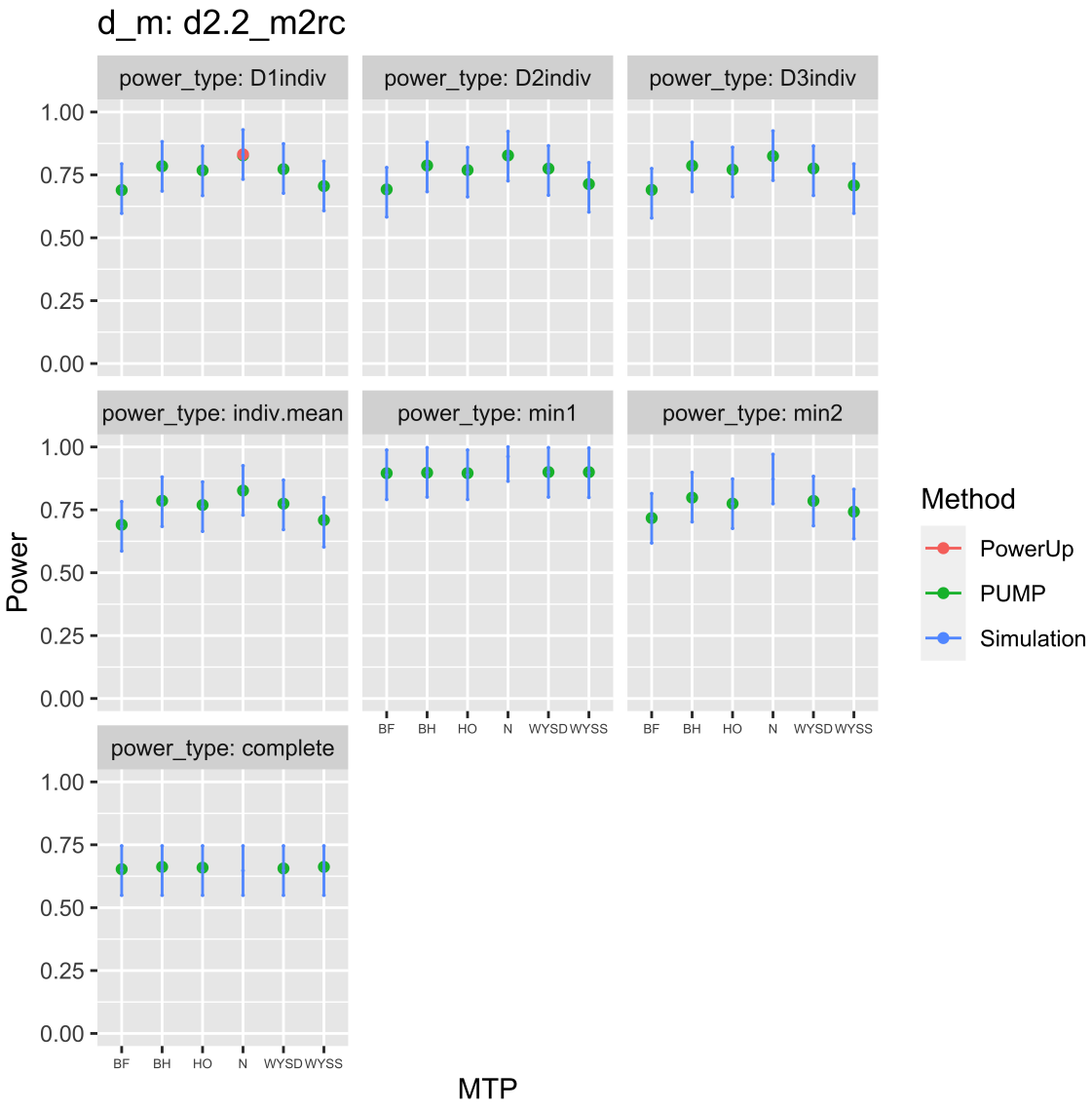


Figure 1: Validation plot

Next, we validate 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 9, 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 $J = 20$.

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 6: 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 7: Sample size validation