Rapid and comprehensive power analysis in R, with a focus on planned error control for multiple outcomes

SREE

Workshop hosts:
Kristen Hunter, University of New South Wales Sydney
Luke Miratrix, Harvard Graduate School of Education
SREE Webinar, November 2024

Welcome!
Does this work for you in RStudio?
If not, please raise hand as we wait for everyone to log in.

Workshop materials
https://tinyurl.com/SREEPUMP

```
install.packages( “PUMP” )
library( PUMP )
pump_info()
```

Materials modified from: Luke Miratrix
Harvard Graduate School of Education

Introductions

Quick intros (in the chat):

- Name and institution
- Why are you here? (1 sentence)

For me:

- Kristen Hunter, University of New South Wales in Sydney, Australia
- I am here because I am excited to help researchers design studies that are well-powered enough to be effective.

Useful links

- PUMP package on CRAN
  - Contains vignettes
- Shiny app
  - No coding required
- PUMP package on Github
  - Contains the latest version of the package
- Journal of Statistical Software article
  - Note the Technical Appendix
- Contact
  - Kristen Hunter kristen.hunter@unsw.edu.au
  - Luke Miratrix lmiratrix@g.harvard.edu

Workshop materials
https://tinyurl.com/SREEPUMP

Outline

PART I: Designs and models

PART II: Calculating necessary sample sizes

PART III: Multiple outcomes, multiple testing

PART I
Designs and models

Mike Stobe / Getty Images for Westminster Kennel Club

Outline of Part 1

PART I: Designs and models

1. The planning for experiments
2. What are power analyses for?
3. Making the plan: picking a design and analytic model

Goal: Set stage of a power analysis and introduce the PUMP package which will help with power calculations.

PART II: Calculating necessary sample sizes

PART III: Multiple outcomes, multiple testing

A hypothetical experiment

Example: A literacy intervention where we treat at the classroom level.

Our goal: As scientists, we want to objectively ascertain whether this literacy intervention works.

Our secret goal: We believe in our intervention and want to show the world!!

	We are right	We are wrong
Our analysis shows an impact	We have our desired evidence!	We are going to look foolish… eventually
Our analysis does not show an impact	**We have missed an opportunity**, and people will think our intervention doesn't work	**If our estimates are precise enough**, we will have learned something important.

An experiment with noisy estimates is dangerous

A null result due to noise will usually be taken by the world as evidence an intervention doesn't work.

- "Inconclusive" is read (incorrectly) as "Failure"

Getting grant funding to implement an intervention that works, and failing to show that it works:

- Bad for the intervention
- Bad for the researcher who believed in the intervention
- Bad for society for not benefitting from a good idea.

Honest power analyses are important for many parties, including the initiating researchers.

Anthony Behar / Sipa USA / Reuters

What is an experiment's "Plan"?

For our literacy example, a plan is something like:

What will our experiment look like? (Design)

We know our data is hierarchical with 3 levels: students in classrooms in schools.

We will randomize classrooms inside schools.

We will randomize half the classrooms to treatment.

How are we going to analyze our data? (Model)

We plan on fitting a multilevel regression model. We will have fixed effects for each school (our randomization blocks). We will have a random intercept for each classroom (to account for clustering).

PUMP: A code for a plan

We encode our **design** as follows:

- A: How many levels in our experiment (students, classrooms, schools)?
- B: What level will we randomize?

And then, for our **model**, for each level in our model:

- Do we have a (r) random intercept or (f) fixed intercept?
- Do we have a
  - c: constant treatment impact parameter for all units?
  - r: random effect for treatment?
  - f: fixed effect interaction of treatment and group id?

This gives you a plan code (d_m, which stands for design_model) like this:

d2.1_m2rr

Design: 2 level design, randomization at level 1.

Model: We will fit a model with a random intercept and a random treatment effect.

(This model allows for cross-site treatment variation)

PUMP: Supported plans

Designs supported

- Levels: 1, 2, 3
- Randomization level: 1, 2, 3

Models supported

- Assumes mixed effects linear regression models
- Covers most regressions usually seen “in the wild”
- Allows for cross-site treatment variation or not.

Plans supported

- d1.1_m1c
- d2.1_m2fc
- d2.1_m2ff
- d2.1_m2fr
- d2.1_m2rr
- d2.2_m2rc
- d3.1_m3rr2rr
- d3.2_m3fc2rc
- d3.2_m3ff2rc
- d3.2_m3rr2rc
- d3.3_m3rc2rc

Advertisement: Technical Appendix

The Technical Appendix of the JSS article gives detailed information about each design and model.

- Explanation/motivation of the model
- Algebraic equation representation, including reduced form
- Standard error of treatment effect estimate
- Degrees of freedom
- Sample size formula
- Assumptions encoded in the model, such as no cross-site variation
- R code syntax to fit the model

What about power?

Timothy A. Clary/AFP/Getty

Planning an experiment: some questions

We have (or are hoping to get) resources, and given the resources we have some questions:

Q: How powerful is my experiment?

A: The answer here is always “it depends on how effective your intervention actually is,” which is what you are trying to learn by running the experiment. So perhaps this is a silly question.

Q: How many experimental units do we need?

A: As many as you can get. So this is also kind of the wrong question.

- This is the usual “power analysis” question of how big an experiment you need to get 80% power. It still suffers a bit from the problems of the question above. But more on this one later.

Q: Should we have more clusters or bigger clusters? (E.g., is it worth testing a subsample of kids in more classrooms?)

A: This is a good question, but is often not the main question.

- Experimental optimization is a way of figuring out how to most effectively use ones resources. But usually a ballpark estimate will be enough, and is hard enough.

A key question about power: What is the MDE?

Q: What size impact will my experiment reliably detect?

A: The answer is the Minimum Detectable Effect (MDE)

Given a plan for an experiment, and some baseline knowledge of context, we can give a rough but useful answer to this question.

Elements we need:

- The plan: “We are going to randomize 12 schools with about 300 kids per school…”
- The context: “We have some covariates that predict our outcome decently well, and variation is mostly at the student level, not school level…”

A bit more on what the MDE or MDES is

The MDE is “the smallest true impact that an experiment has a good chance of detecting” (Bloom, 1995).

MDE = Minimal Detectable *True* Effect

MDES = Minimal Detectable *True* Effect Size (standardized MDE)

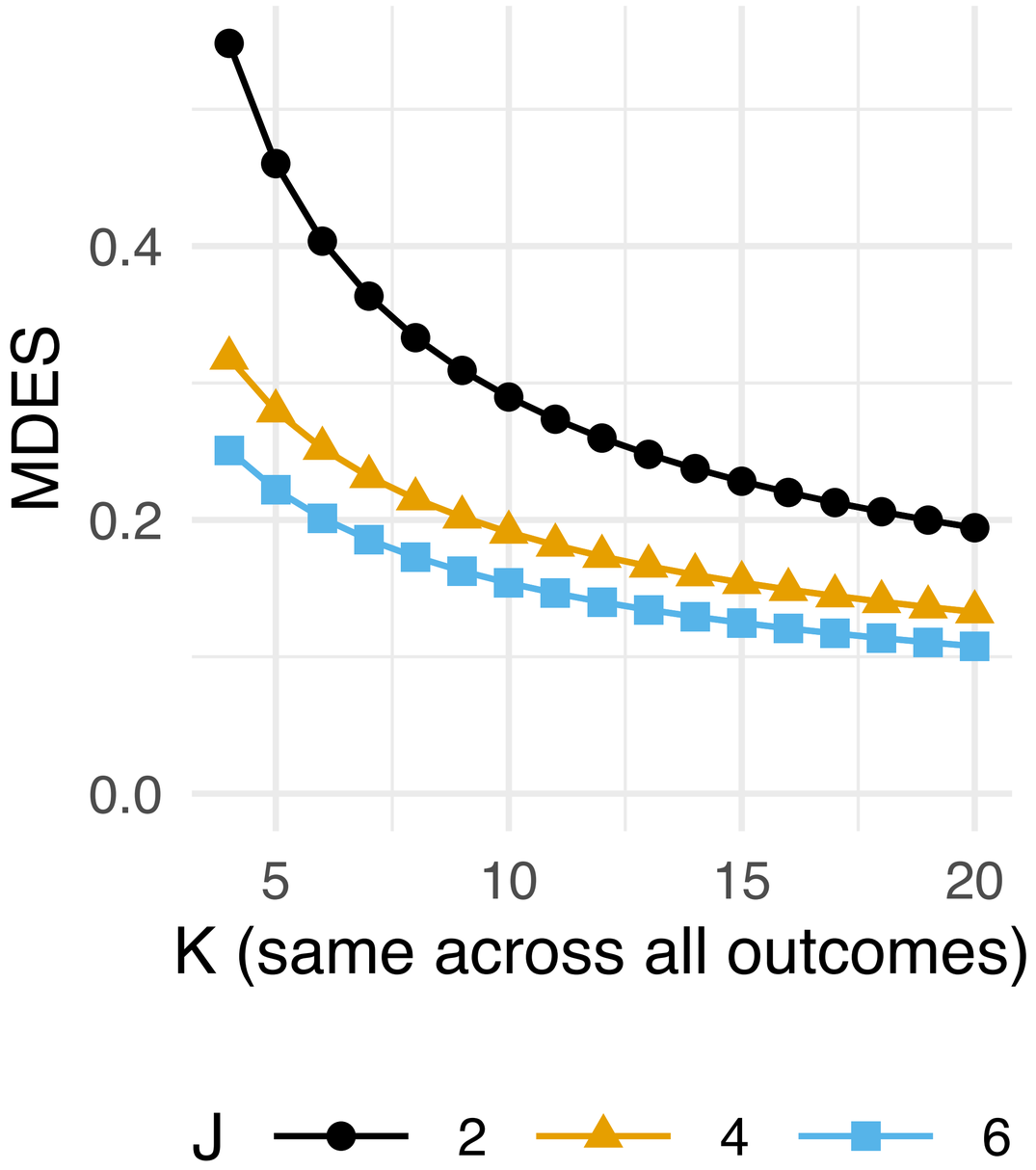
- MDE is measured in the units of the impact: test score points, dollars, etc.
- The MDES is a MDE in standardized units of the outcome: MDES = MDE/standard deviation of the outcome

Define the standard deviation of the outcome as σ. If we are running an experiment with MDES = 0.10σ, then *if the true impact* were 0.10σ, then we will have at least an 80% chance of finding a significant result. For example, you might say “We have an 80% chance to detect that the reading test scores are 0.1 standard deviations higher in the intervention group than the control group.”

For large sample sizes, 80% power, 0.05 alpha: MDE = 2.8 SE, where the SE is the true standard error of our estimator.

Because MDE = 2.8 SE, if MDE = 0.10σ, then SE would be about 0.036σ, and the confidence interval would be 0.14σ wide.

The benefits of more. A key power graph of MDES



Our Design: schools (K) with classrooms (J) with students (n)

We randomize classrooms inside schools. n is set to 23 (we know this from pilot data)

Story:

If we have 10 schools, and 4 classrooms per school, we can reliably detect an effect of a bit less than 0.20 effect size units.

To reliably detect an effect of 0.10 effect size units, we would need about 20 schools and 6 classrooms per school.

And how do I easily calculate power (and make that plot you showed me)?

Karen Betancur / Reuters

PUMP: Power Under Multiplicity Project

Kristen Hunter, Luke Miratrix, Kristin Porter, Zarni Htet

Funded in part by the Institute of Education Sciences (Grant R305D170030)

Image by Jan Steiner from Pixabay

Motivation of PUMP

- PUMP: R package on CRAN
- Stands for “Power Under Multiplicity Project”
- Calculates power for multiple hypotheses in multilevel randomized controlled trials (RCTs)
- Multilevel: hierarchical structure, such as students nested within schools nested within districts
- Assumes frequentist mixed effects linear models

It was originally designed for handling multiple outcomes.

It is also a very useful package for power analyses in general.

Installing and using PUMP

```
# CRAN version
install.packages( "PUMP" )

# Github version (latest version)
devtools::install_github( "https://github.com/MDRCNY/PUMP" )



# three key methods

pump_power( … )   # What is the power of this specific plan?
pump_mdes( … )    # How small an effect can this specific plan detect?
pump_sample( … )  # What sample size do I need to detect an effect of
                  # this specified size?

# "grid" methods
pump_*_grid( … )  # As above, but repeat across a range of design
                  # parameters, e.g., R2 or ICC or sample sizes
```

Notation for the PUMP Package

```
m1 <- pump_mdes(
  d_m =  "d2.1_m2fc",     # choice of plan: design and model
  target.power = 0.80,    # what desired chance of success?
  power.definition = "D1indiv", # (discussed later)
  alpha = 0.05,           # significance level
  J = 20,                 # number of classrooms/blocks
  nbar = 50,              # number of students in each classroom/block
  Tbar = 0.50,            # proportion treated
  numCovar.1 = 5,         # number of covariates at level 1 (students)
  R2.1 = 0.6,             # assumed R^2 of level 1 covariates (students)
  ICC.2 = 0.20,           # intraclass correlation of level 2 (classrooms)
  omega.2 = 1             # impact variation as proportion of ICC.2
)
```

What experiment are we calculating MDE for?

Getting scripts up and running

Let's make sure everyone can run the scripts to make the plots we have seen.

Steps

- Open script 1
- Install PUMP
  - See commented out line at top.
- Run the script. Note, in particular, "pump_info()"
- Change the script to calculate MDES for
  - 30 classrooms, 23 kids per classroom
  - 90% power

Pause: try the code yourself!

Julia Nikhinson/ AP

The results of the original MDES calculator call for a multisite experiment

```
summary( m1 )
## mdes result: d2.1_m2fc d_m with 1 outcomes
##    target D1indiv power: 0.80
##    nbar: 50    J: 20    Tbar: 0.5
##    alpha: 0.05
##    Level:
##      1: R2: 0.6 (5 covariates)
##      2:    fixed effects     rho = 0
##    MTP Adjusted MDES D1indiv.power    SE
##   None      0.100333              0.8 0.01
##   (max.steps = 20   tnum = 1000   start.tnum = 100   final.tnum = 4000
```

These are all the specific design parameters we put in, in case we forgot.

This is our MDES for this design!

Some of this printout is for the multiple outcomes stuff we will do later on.

Same experiment, but using the “FIRC” model

Fixed Intercept, Random Coefficient: A model that allows for cross-site impact variation

```
m2 <- update( m1, d_m = "d2.1_m2fr" )

summary( m2 )
## mdes result: d2.1_m2fr d_m with 1 outcomes
##    target D1indiv power: 0.80
##    nbar: 50    J: 20    Tbar: 0.5
##    alpha: 0.05
##    Level:
##      1: R2: 0.6 (5 covariates)
##      2:    fixed effects    ICC: 0.2     omega: 1
##    rho = 0
##    MTP Adjusted.MDES D1indiv.power    SE
##   None     0.3199859             0.8 0.01
##   (max.steps = 20  tnum = 1000  start.tnum = 100  final.tnum = 4000
```

With PUMP, you can just update a prior calculation to recalculate with a new aspect. Here we convert to a FIRC model from the usual OLS analysis.

Our MDES has gone up a huge amount!! The choice of model can be very important.

Two scenarios show a big difference

These models are targeting different estimands/research questions!

Luke W Miratrix, Michael J Weiss, and Brit Henderson. An applied researcher's guide to estimating effects from multisite individually randomized trials: Estimands, estimators, and estimates. Journal of Research on Educational Effectiveness, 14(1):270–308, 2021.

Cross-site variation

We saw two scenarios, but how much did the amount of cross site variation (omega.2) matter?

Omega is cross-site variation, the amount of treatment variation across sites

We can check by doing a *sensitivity analysis* as follows:

m1 is our original scenario (we called update on it before)

```
m3 <- update_grid( m1,
   d_m = c( "d2.1_m2fr", "d2.1_m2fc" ),
   omega.2 = seq( 0, 1, length.out = 7 ) )
```

We pass lists of options for our various parameters. It will call the original calculator on every combination of the passed parameters.

The rest of the parameters are just copied over from the original call.

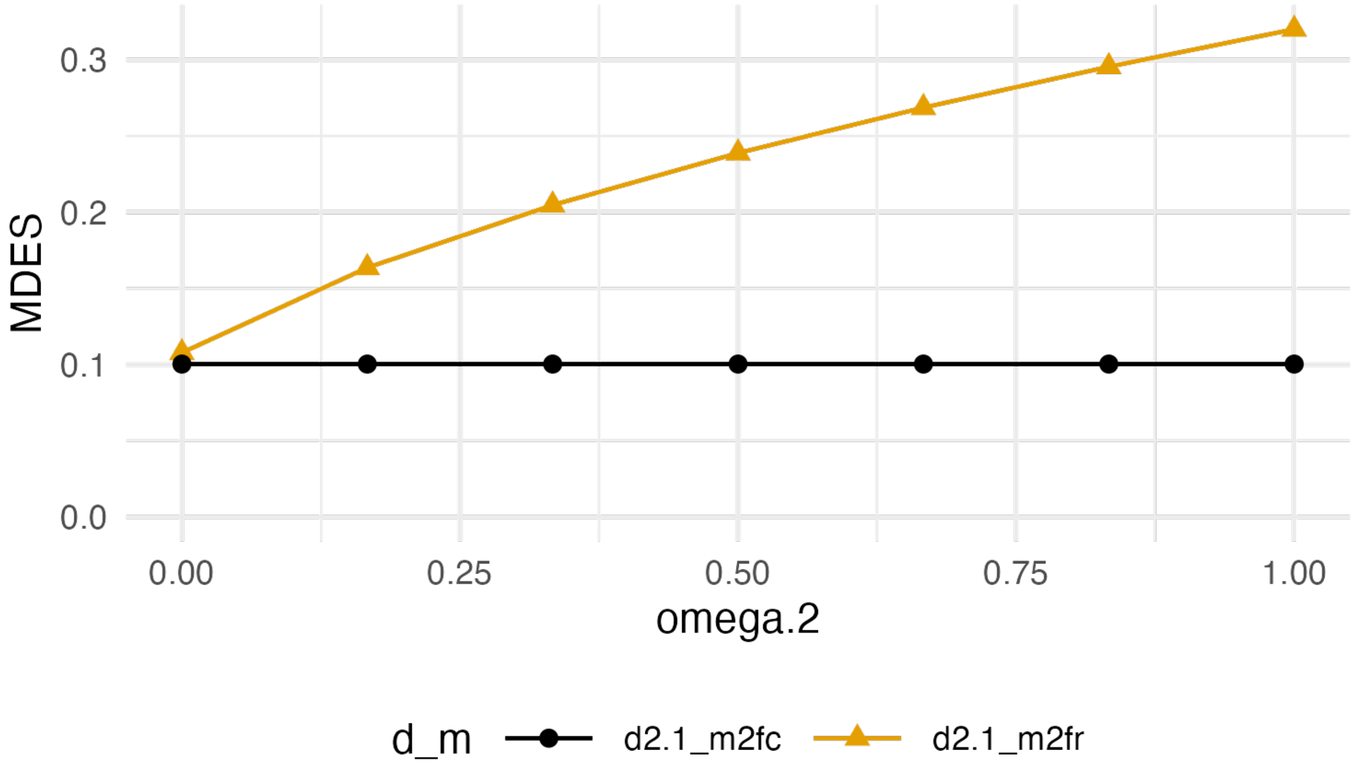
Comparing our two models across different omega.2 values

```
> m3
mdes grid result: d2.1_m2fr d_m with 1 outcomes
 mdes grid result: d2.1_m2fc d_m with 1 outcomes
Varying across d_m, omega.2
   MTP        d_m   omega.2 Adjusted.MDES D1indiv.power
 None d2.1_m2fr 0.0000000    0.09334948            0.8
 None d2.1_m2fr 0.1666667    0.15441119            0.8
 None d2.1_m2fr 0.3333333    0.19741202            0.8
 None d2.1_m2fr 0.5000000    0.23259448            0.8
 None d2.1_m2fr 0.6666667    0.26311382            0.8
 None d2.1_m2fr 0.8333333    0.29044375            0.8
 None d2.1_m2fr 1.0000000    0.31541443            0.8
 None d2.1_m2fc 0.0000000    0.08689094            0.8
 None d2.1_m2fc 0.1666667    0.08689094            0.8
 None d2.1_m2fc 0.3333333    0.08689094            0.8
 None d2.1_m2fc 0.5000000    0.08689094            0.8
 None d2.1_m2fc 0.6666667    0.08689094            0.8
 None d2.1_m2fc 0.8333333    0.08689094            0.8
 None d2.1_m2fc 1.0000000    0.08689094            0.8
```

We get all combinations of the factors we gave "grid"

So here we have 14 MDES calculations, 7 for each design we are considering.

The amount of cross site variation is critical



Better than a table, we plot our result and get our plot at left.

The constant impact model does not care how much cross site variation there is.

The random impact model does!

PART II
Necessary sample sizes

Karen Betancur / Reuters

Outline of Part II

PART I: Designs and models

PART II: Calculating necessary sample sizes

1. Calculating sample size requirements given a desired MDES
2. Doing sensitivity checks to see how answers can vary.

Goal: Get some practice doing power calculations and explorations.

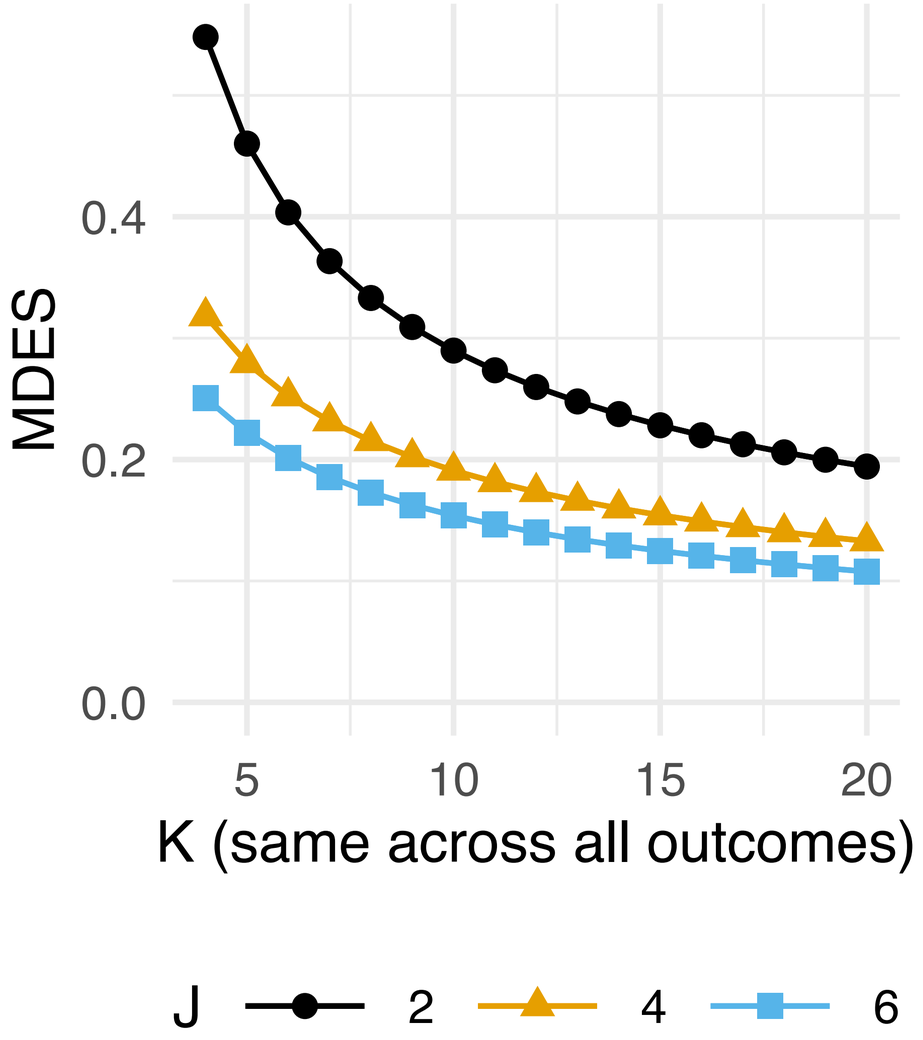
PART III: Multiple outcomes, multiple testing

How big a sample do we need (to detect a given size effect)?

Back to our literacy study.

Say we want to ask how many schools we need to detect an effect size of 0.15.

We can look at our prior MDES graph, or just calculate directly.



Necessary sample size for specific scenario

```
> ss1 <- pump_sample( d_m = "d3.2_m3fc2rc",
+                     typesample = "K",
+                     power.definition = "D1indiv",
+                     MDES = 0.15,
+                     target.power = 0.80,
+                     alpha = 0.05,
+                     nbar = 23, J = 4, Tbar = 0.5 )
> ss1
sample result: d3.2_m3fc2rc d_m with 1 outcomes
  target D1indiv power: 0.80
  MTP Sample.type Sample.size D1indiv.power   SE
 None           K          16            0.8 0.01
```

We can calculate needed sample size for nbar (level 1, students), J (level 2, classrooms) or K (level 3, schools)

We supply the levels we know.

We need 16 schools (with 4 classrooms per school)

But it's better to see a range across scenarios

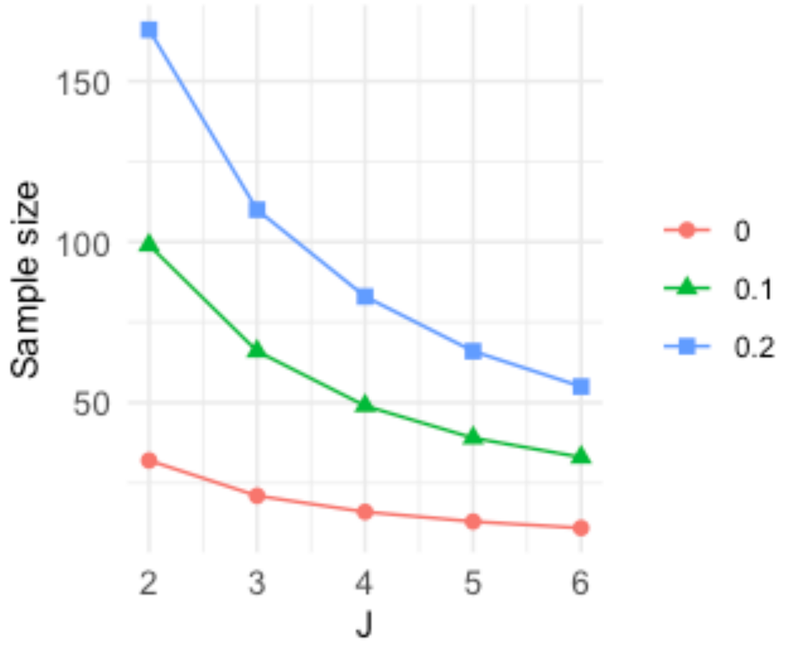
```
> ss2 <- update_grid( ss1, J = c( 2, 4, 6 ) )
> ss2
sample grid result: d3.2_m3fc2rc d_m with 1 outcomes
Varying across J
  MTP           d_m MDES J Sample.type Sample.size D1indiv.power
 None d3.2_m3fc2rc 0.15 2           K          32            0.8
 None d3.2_m3fc2rc 0.15 4           K          16            0.8
 None d3.2_m3fc2rc 0.15 6           K          11            0.8
```

Or a fancy plot across even more scenarios!

```
ss3 <- update_grid( ss1,
                    J = c( 2, 3, 4, 5, 6 ),
                    ICC.2 = c( 0, 0.1, 0.2 ) )

plot( ss3, color = "ICC.2" )
```

"color" tells plot to group our results by this feature, so we can see an "Interaction Plot" style plot.

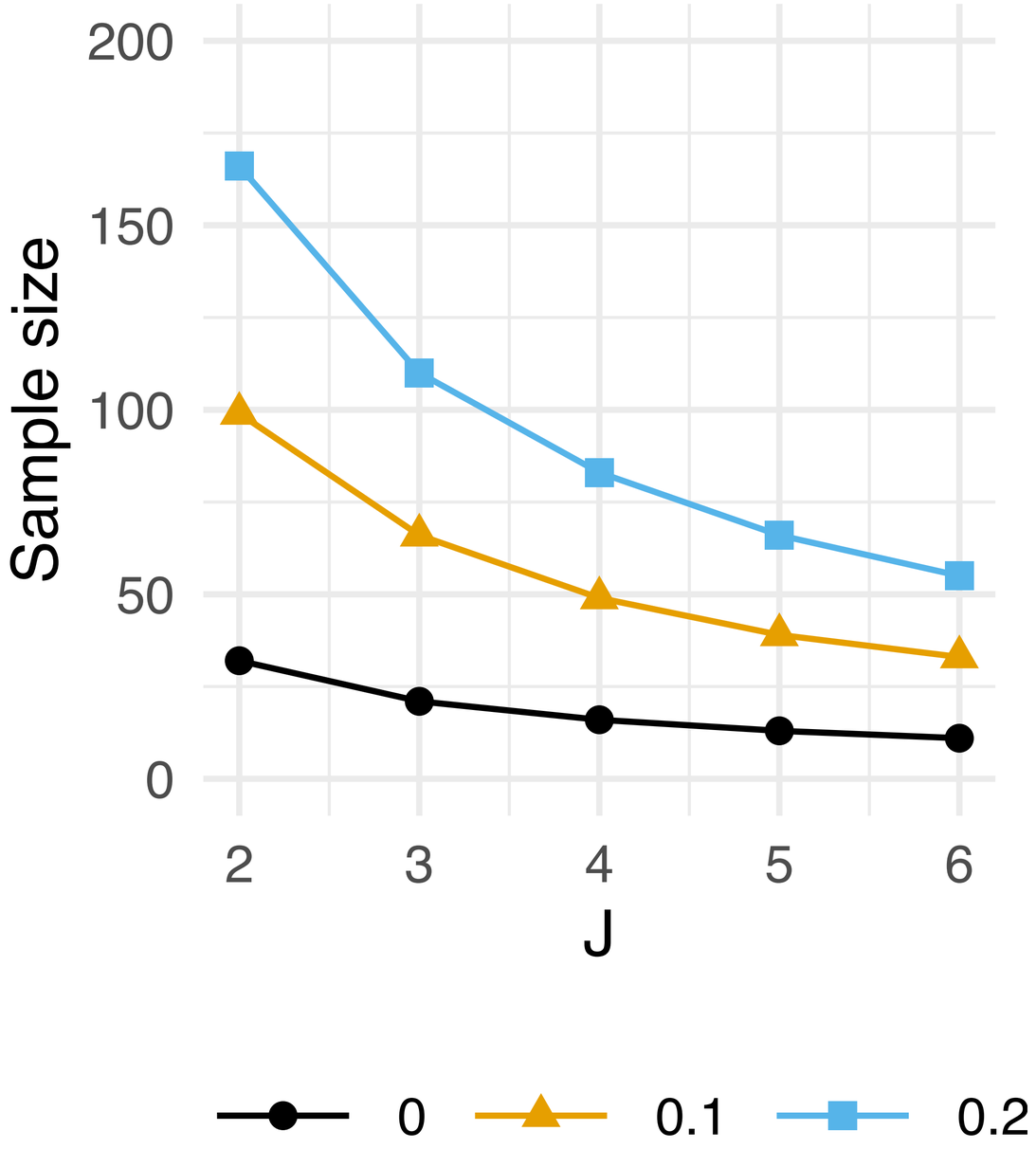



```
# Fancier
library( ggthemes() )
mytheme = theme_minimal() +
    theme( legend.position = "bottom",
           legend.direction = "horizontal",
           legend.key.width = unit(1,"cm"),
           panel.border = element_blank() )

plot( ss3, color = "ICC.2" ) +
    scale_y_continuous( limits = c(0, 200 ) ) +
    mytheme +
    scale_color_colorblind()
```

This makes the same plot as above, but the stuff after the plot() command makes our plot slightly prettier.

ICC is a killer, and we missed it the first go-around!



By doing a sensitivity analysis we can quickly discover what features of our context are most dangerous to our power.

This also tells us the trade-off between J and K: More J does substantially reduce K.

What about classroom size?…

ICC = intraclass correlation, a measure of variation in outcomes at a particular level

Lab: Exploring a trade-off

Let's say you are considering subsampling classrooms as this allows you to budget for more schools.

TASK

- Load and run the script to do the prior analysis
- Modify the script to explore the tradeoff between nbar (average classroom size) and K (number of schools needed).

QUESTIONS

- If we only sample 8 kids per classroom, how many schools do we need? (say we have 4 classrooms per school)
- Could we get away with 8 schools if our classrooms were big enough?

Wrap up

- If we only sample 8 kids per classroom, how many schools do we need? (say we have 4 classrooms per school)

- Could we get away with 8 schools if our classrooms were big enough?

Eduardo Munoz / Reuters

Break

(Code not working? Raise hand.)

Lokman Vural Elibol / Anadolu/ Getty

Anthony Behar/ Sipa USA/ Reuters

PART III
Multiple outcomes, multiple testing

Timothy A. Clary/ AFP/ Getty

Outline of Part III

PART I: Designs and models

PART II: Calculating necessary sample sizes

PART III: Multiple outcomes, multiple testing

1. What is multiple testing correction?
2. How do we even define power with multiple outcomes?
3. How can we design experiments with these factors?

Goal: Build understanding of these new terms and concepts, and see how they can play out in practice.

The multiple testing problem

Say you run 20 experiments on a treatment that has **no impact.**

Each experiment still has a 1 in 20 chance of **erroneously rejecting the null.**

So across the 20 experiments, you will likely find at least 1 significant effect!

Now consider 20 outcomes:

- If there is independent noise in the outcomes, then this is kind of like the 20 experiments.
- We have a problem.

The procedure you may have heard of: Bonferroni

Say you have 4 outcomes/hypothesis tests.

You have an alpha of 0.05.

A **family wise error control** is to ensure you only have a 0.05 chance of erroneously rejecting **any** of your 4 tests.

The way to do this:

- Divide your 0.05 by 4, and only reject each test if you are below 0.05/4
- Analogously, multiply all your p-values by 4.
- This is conservative (it makes p-values bigger than they need to be) but is guaranteed.

Four multiple testing procedures

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 1: Summary of MTP procedures.

Holm is a less conservative version of Bonferroni.

WY is computationally intensive, but less conservative.

BH controls the *false discovery rate*, and is thus a more liberal (error prone) approach

Multiple outcomes require adjustment

- With **multiple outcomes** in a study, must use a multiple testing procedure (MTP) to obtain valid conclusions.
- The use of MTPs changes statistical **power** (multiplying all your p-values by 4, or something, will make it harder to reject anything!)

Problem:

Current practice for determining statistical power does not take the use of MTPs into account

So how do you decide how to design your experiment?

The effect of test correction

I have 5 outcomes, with a 2-level blocked design.

My power to detect the effect for any individual outcome:

Without any adjustment (no MTP): 0.81.

Using Bonferroni adjustment: 0.6.

Having multiple outcomes has reduced my power…or has it? Stay tuned!

Eduardo Munoz / Reuters

Multiple outcomes means multiple definitions of power

How do we define power if we have ***multiple*** hypotheses/outcomes?

Individual power: probability of rejecting a particular null hypothesis (e.g., chance of finding an impact on *reading* in particular*)*

1-Minimal power: probability of rejecting at least one null hypothesis (e.g., finding impact on at least one of *reading, math, or science*)

d-Minimal power: probability of rejecting at least *d* null hypotheses (e.g., finding impact on at least two of *reading, math, or science*)

Complete power: probability of rejecting all the null hypotheses (e.g. finding impact on all of *reading, math and science*)

All valid options–the choice depends on how we want to define success!

Factors affecting power

With at least one outcome:

- Plan: design of the study and assumed model
- $\bar{n}, J, K$: number of level 1/2/3 units
- $\bar{T}$: proportion of units treated
- R^2: explanatory power of covariates (and also the number of covariates)
- ICC: intraclass correlation (ratio of the variance at a particular level to overall variance)
- ***Omega*:** cross-site impact variation

Unique to **multiple** outcomes:

- M: number of outcomes/tests
- proportion of outcomes for which there are truly effects
- ρ: correlation between test statistics

Also choices:

- the definition of power used
- the multiple testing procedure (MTP) used

Our multisite experiment (with 1 outcome)

```
library( PUMP )

p1 <- pump_power(
  d_m = "d2.1_m2fc",        # choice of design and model
  MDES = 0.10,              # assumed effect size
  M = 1,                    # number of outcomes
  J = 10,                   # number of classrooms/blocks
  nbar = 275,               # average number of students per classroom
  Tbar = 0.50,              # proportion treated
  alpha = 0.05,             # significance level
  numCovar.1 = 5,           # number of covariates at level 1
  R2.1 = 0.1,               # assumed R^2 of level 1 covariates
  ICC.2 = 0.05,             # intraclass correlation
  rho = 0.4                 # test statistic correlation
)
```

Our multisite experiment (with 1 outcome)

```
power result: d2.1_m2fc d_m with 1 outcomes
  MDES vector: 0.1
  nbar: 275  J: 10 Tbar: 0.5
  alpha: 0.05
  Level:
    1: R2: 0.1 (5 covariates)
    2:   fixed effects   ICC: 0.05   omega: 0
  rho = 0.4
  MTP    D1indiv          SE1  df1
 None 0.8091271 0.03526523 2734
  (tnum = 10000)
```

We have designed this experiment to have 80% power with 1 outcome.

Our multisite experiment (with multiple outcomes)

Bonferroni correction

```
library( PUMP )

p2 <- pump_power(
  d_m = "d2.1_m2fc",          # choice of design and model
  MTP = "BF",                 # multiple testing procedure
  MDES = rep( 0.10, 5 ),      # assumed effect size, one per outcome
  M = 5,                      # number of outcomes
  J = 10,                     # number of classrooms
  nbar = 275,                 # average number of students per classroom
  Tbar = 0.50,                # proportion treated
  alpha = 0.05,               # significance level
  numCovar.1 = 5,             # number of covariates at level 1
  R2.1 = 0.1,                 # assumed R^2 of level 1 covariates
  ICC.2 = 0.05,               # intraclass correlation
  rho = 0.4                   # test statistic correlation
)
```

Comparing different powers with Bonferroni adjustment

The prior slide produces the following values

MTP	D1indiv	D2indiv	…	indiv.mean	min1	min2	min3	…	complete
None	0.81	0.81	…	0.81	NA	NA	NA	…	NA
BF	0.60	0.60	…	0.60	0.92	0.8	0.64	…	0.49

Note min1 is **larger** than even the nominal unadjusted 80%!

Individual power is same for each outcome, because we modeled them as all having the same impact, etc. — we could change that

Complete power is low: it is hard to reject every single one of our tests

What if only 2 of my 5 outcomes are actually impacted by treatment?

```
> p3 <- update(p2, numZero = 3)
> summary(p3)
power result: d2.1_m2fc_d_m with 5 outcomes (3 zeros)
  MDES vector: 0.1, 0.1, 0, 0, 0
  nbar: 275  J: 10  Tbar: 0.5
  alpha: 0.05
  Level:
    1: R2: 0.1 (5 covariates)
    2:   fixed effects   ICC: 0.05 omega: 0
  rho = 0.4
  MTP D1indiv D2indiv indiv.mean   min1   min2 complete         SE1         SE2
 None  0.8119  0.8110    0.81145     NA     NA       NA 0.03526523 0.03526523
   BF  0.6034  0.5994    0.60140 0.7904 0.4278       NA          NA          NA
        SE3         SE4         SE5  df1
 0.03526523 0.03526523 0.03526523 2734
         NA          NA          NA   NA
   0.002 <= SE <= 0.002
```

Q: How many schools do I need to have an 80% chance to find significance on at least one outcome?

A:

We specify we are looking at min1 power, the probability of detecting an effect on at least 1 outcome.

```
ss <- pump_sample(
  target.power = 0.8,                # target power
  power.definition = "min1",         # power definition
  typesample = "J",                  # type of sample size procedure
  tol = 0.01,                        # tolerance
  d_m =  "d2.1_m2fc", MTP = "BF",
  MDES = 0.1, M = 3, nbar = 350, Tbar = 0.50, alpha = 0.05,
  numCovar.1 = 5, R2.1 = 0.1, ICC.2 = 0.05, rho = 0.4
)
```

Our experiment with 1 outcome needed 10 classrooms to achieve 80% individual power.

MTP	Sample.type	Sample.size	min1.power
BF	J	6	0.81

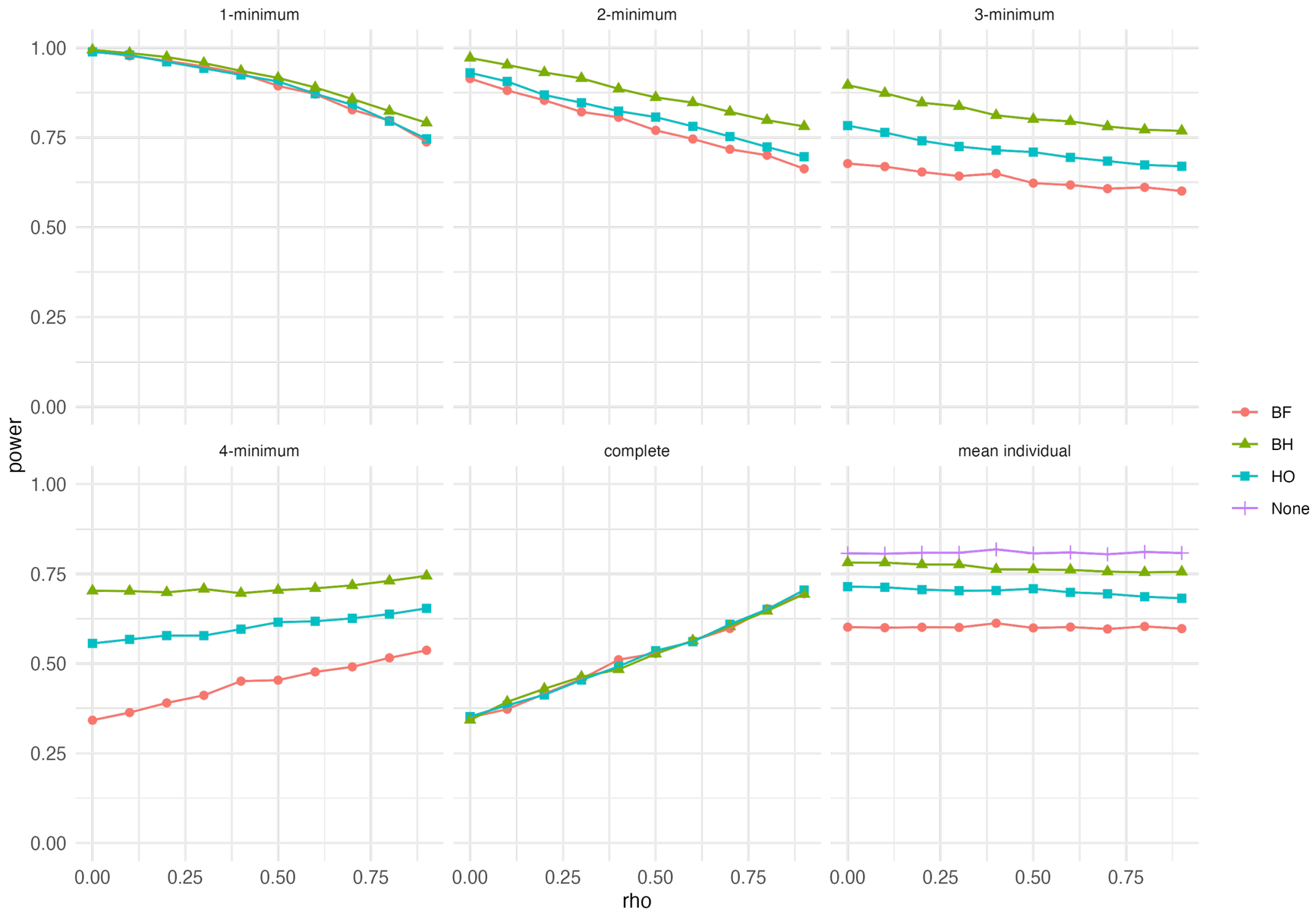
Assessing sensitivity

As with single outcomes, we can use the grid function to assess sensitivity to different model and design parameters.

```
pgrid <- update_grid(
   p2,
   # vary parameter values
   rho = seq( 0, 0.9, by = 0.1 ),
   # compare multiple MTPs
   MTP = c( "BF", "HO", "BH" )
)
```

Assessing sensitivity

```
plot( pgrid, var.vary = 'rho' )
```



Wrapping Up

Timothy A. Clary / AFP / Getty

How does PUMP work?
And some warnings.

https://www.flickr.com/photos/jdoakey/5213764593

How the PUMP package works

Simulation-based approach for power

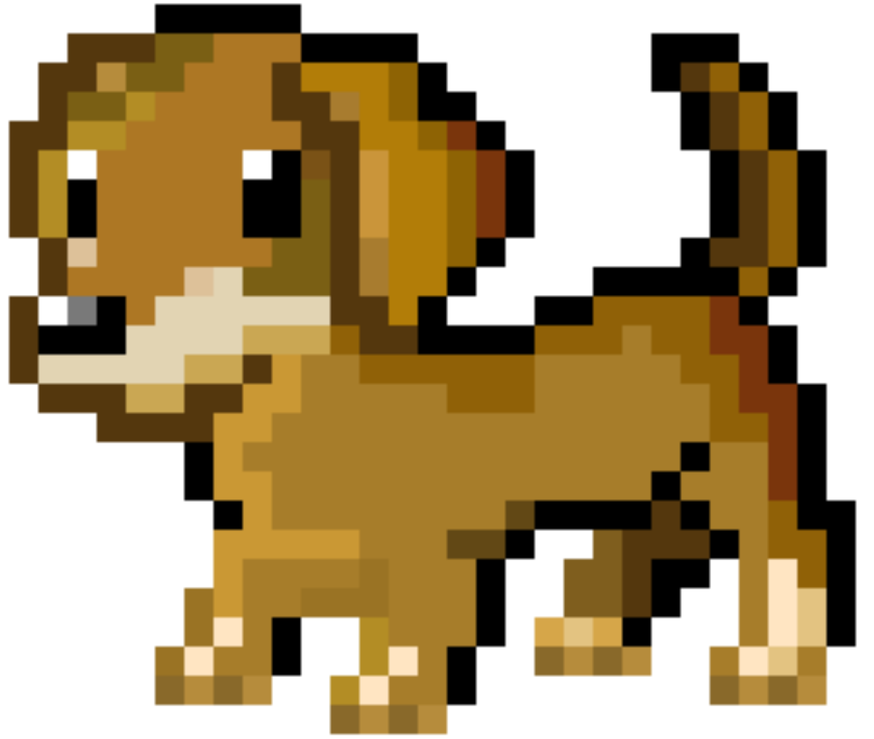
- We simulate data from a hypothetical experiment with the assumed design & parameters.
- We analyze it using the assumed model.
- We get the p-value.
- Repeat! Then calculate an estimate of power.

Because we use a simulation based approach, we get an estimate of the approximate power, and it can change slightly when you re-run the code.

Search-based approach for MDES and sample size

- We iterate through possible sample size values and find the smallest one with the appropriate power. (Same for MDES)

Answers are approximate



The answers are calculated by simulation.

pump_mdes and pump_sample search for meds or sample sizes to achieve a target *estimated* power.

Each time you run the code, you might get slightly different results.

That being said: power analyses are **approximate by nature,** as they are built on guessed assumptions.

Sample sizes for lower levels can be very unstable

For a cluster randomized experiment, even infinite-sized classrooms might not be enough—if there is too much variation at the higher levels.

If you are close to this edge, then you can get extreme sample sizes.

This tells you that you need to focus on more upper level units.

Not a warning: some bonus features of the package

Functions to simulate data from multilevel RCTs

- Useful for running simulations or just getting a practice dataset you can use for demonstration purposes.

Function to estimate the approximate correlation between ***test statistics*** based on the correlation between ***outcomes*** (using a simulation approach)

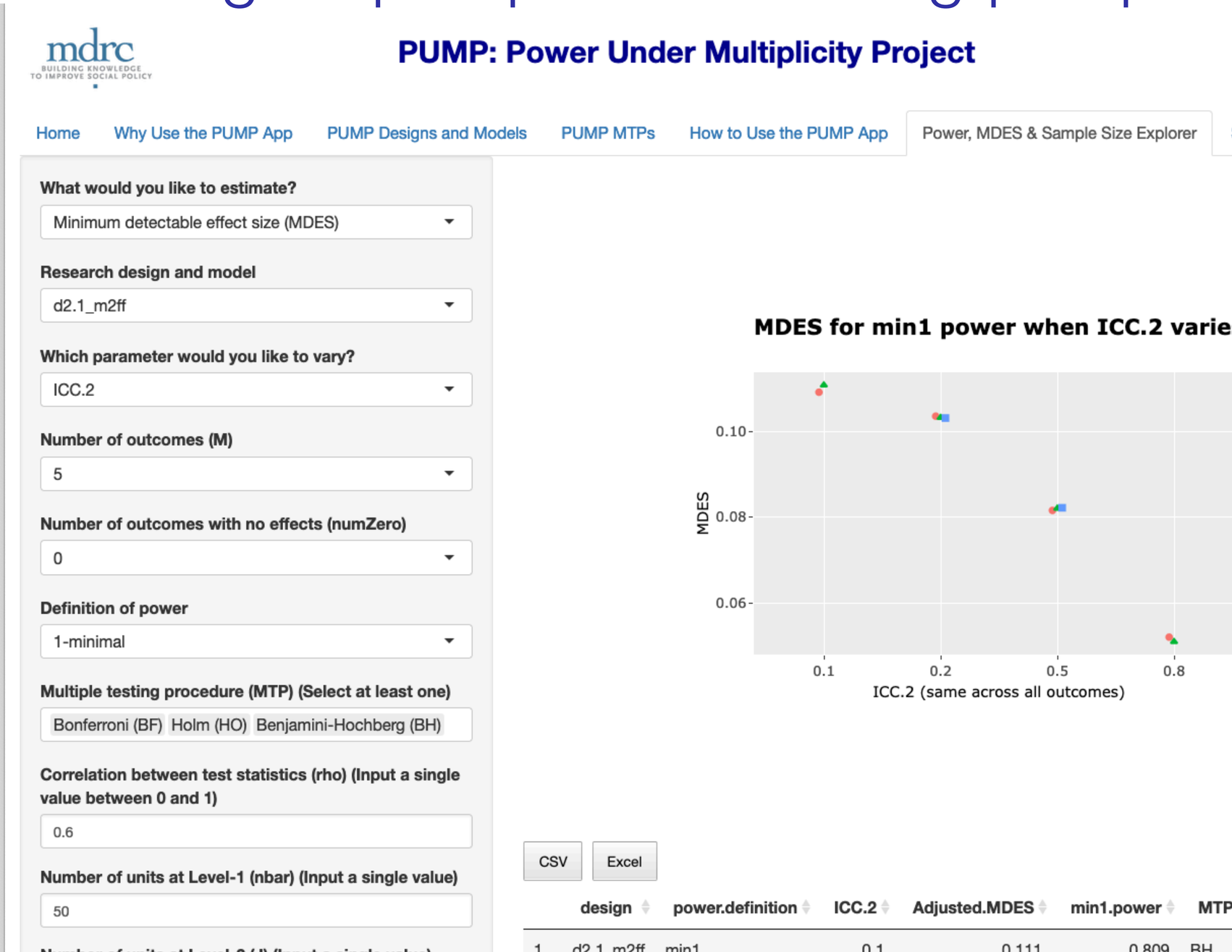
- The correlation of test statistics (rho) is a *weird design parameter*
- This tool allows for estimating it in practice

An alternative to the R coding: https://public.mdrc.org/pump/

Web-based interface to the power calculator we have been discussing.

Slightly less flexibility, but:

- Can download resulting tables and figures easily
- Easy to change parameters
- Sensitivity checks or specific checks of individual scenarios.



Useful links

- PUMP package on CRAN
  - Contains vignettes
- Shiny app
  - No coding required
- PUMP package on Github
  - Contains the latest version of the package
- Journal of Statistical Software article
  - Note the Technical Appendix
- Contact
  - Kristen Hunter kristen.hunter@unsw.edu.au
  - Luke Miratrix lmiratrix@g.harvard.edu

Workshop materials
https://tinyurl.com/SREEPUMP

Workshop materials
https://tinyurl.com/SREEPUMP

Kate Moran / New York Times

Reference slides

Reference: Parameter definitions

- Cross-site variation: Omega2 is the ratio of variation of impacts to residuals for level 2
- Intraclass correlation: ICC2 is the ratio of the variance at level 2 divided by the overall variance of the individual outcomes

Reference: the correlation between test statistics

Correlation of test statistics (Rho) is a consequence of how coupled your outcomes are:

- Outcomes are basically the same? High rho
- Outcomes are different constructs? Low rho
- Wrinkle: Your outcomes are a function of mechanisms at different levels of data, which means correlation might be hard to conceptualize overall.

How do you pick rho? A tricky design parameter.

- We guess! (And this is an area for future development)
- But it seems to not have major impact on power, except for extreme values.
- Pick a rho, and then do a sensitivity check.

A further simulation option:

- pump has a method to generate data where you specify the correlations of your outcomes, and it tells you the correlation of your test statistics.
- But this means specifying even more parameters to get from pilot data.

Reference: Some parameters to note (with defaults)

pump_power

- tnum (10000) - Number of simulation trials
- B (1000) - For Westfall-Young, number of permutations

pump_sample / pump_mdes

- max.steps (20) - Max number of steps in search
- tol (0.01) - How close estimated power should be to target to count. This is approximate, but controls overall precision of estimates.

Note: Smaller tol (tolerance) will require higher tnum (but package will dynamically set it after giving a warning, if needed)

Reference: multiple testing procedures

Bonferroni

- simple
- most conservative

Holm

- step down version of Bonferroni
- less conservative for larger p-values than Bonferroni

Benjamini-Hochberg

- step up procedure
- controls the false discovery rate (less conservative)

Westfall-Young (single step and step down versions)

- permutation-based approach
- takes into account correlation structure of outcomes
- computationally intensive
- not overly conservative

Reference: Step up/down procedures vs single-step procedures

Consider the following p-values: 0.01, 0.014, 0.25, 0.90

Under Bonferroni, we reject the first, but not the second. Each test is treated separately.

But somehow it seems like having a second p-value so similarly low to the first should count for something…

A *step down* or *step up* procedure rejects tests sequentially by size of p-value, allowing for more lenient rejection of the second, third, etc., test.

Reference: Family wise error vs false discovery rate

Family wise error

- I want to control the chance I make *any* mistake

False discovery rate

- I want to ensure that the average proportion of mistakes I make is below some rate.
- This is a less restrictive requirement for my testing.

Example: Say I had 20 tests to make

- A 10% family wise error rate means I want only a 10% chance of rejecting any null
- A 10% FDR means if I end up rejecting 10 of the 20 tests, I would be fine if 1 of the 10 were erroneous.

Reference: How the PUMP package works

For simple designs and one outcome, we often have a formula for power

It would be difficult (in some cases impossible) to derive explicit formulas for every design, model, number of outcomes, MTP, and definition of power

Instead, we use simulation! A full simulation approach would be:

1. Simulate data according to the alternative hypotheses
2. Calculate test statistics under the alternative hypotheses
3. Use these test statistics to calculate p-values
4. Calculate power using the distribution of p-values

Reference: How the PUMP package works

We can simplify this approach by skipping step 1

Given:

- design and model
- correlation between test statistics for different hypotheses

We know the joint alternative distribution of test statistics!

Results in **simpler** and **faster** power calculations

Simulation approach to calculating power:

1. ***Sample test statistics*** under the alternative hypotheses.
2. Use these test statistics to calculate p-values.
3. Calculate power using the distribution of p-values.

Note: because we use simulations to calculate power, estimates are approximate, but the user can increase the number of test statistic draws to increase precision.

Reference: How pump_sample and pump_mdes work

Both these methods work by using a *search*

Determine a sample size that is way too small and way too big.

Iteratively check sample sizes in the middle until one that is just right is found.

This is kind of like fitting multilevel models, in that you can have convergence issues (which it will warn you about)

See bottom of printout to see how long search was, etc.