

MCLS Workshop: Computing Statistical Power in R Using Simulation

Hugo Lortie-Forgues

June 5, 2022

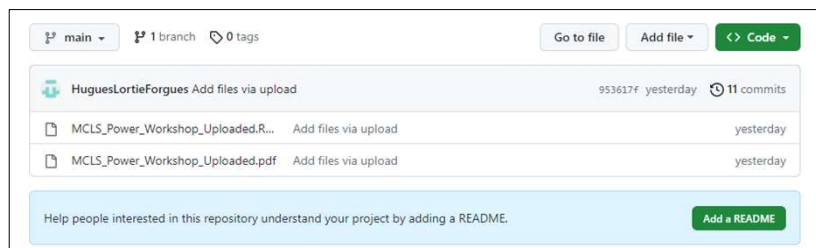


Loughborough
University

1

Material

<https://github.com/HuguesLortieForgues/Workshops>



```
# Useful Functions & concepts used Throughout the workshop
## Function: c() to combine
In R, the "c()" function is used to combine objects into a single vector or list.
Here are a few examples to illustrate the usage of the c() function:

```{r}
vector <- c(1, 2, 3, 4, 5)
vector
```

In this case, the c() function combines the individual elements 1, 2, 3, 4, and 5 into a single vector [1, 2, 3, 4, 5].

Combining multiple vectors into one:

```{r}
vector1 <- c(1, 2, 3)
vector2 <- c(4, 5, 6)
combined_vector <- c(vector1, vector2)
combined_vector
```
```

****Do not run all the code at once****

2

Plan for the workshop

- What is data simulation
- What is power analysis
- What is power analysis using simulation
- Important concepts and R functions for the workshop

- Examples

1. Independent Sample t-test
2. One-Way ANOVA
3. Correlation
4. Paired Sample t-test
5. Repeated-measure ANOVA (3 Levels)
6. Factorial ANOVA (Two Between-Subject Variables)
7. Mixed-design ANOVA (One Between-Subject and One Within-Subject Variable).
8. Simple Regression
9. Multiple Regression with Interaction
10. Regression with Nested Data Using a Mixed-Effect Model

- 14h00 to 15h30
- Break: 30 mins
- 16h00 to 17h30

- I will not talk about how to find the effect size to use in the power analysis.

- My code is not the most efficient/elegant.

3

What is data simulation

- Data simulation is the process of creating synthetic, or artificial, data that mimics the characteristics of real-world data.
 - Sometimes called: Monte-Carlo simulations, Stochastic simulations
- A useful approach to:
 - Calculate power in complex designs
 - Conduct sensitivity analyses (e.g., what if the data is not normal?)
 - Prepare analysis scripts for pre-registration
 - Improve understanding of statistical concepts

4

What is power analysis

- Statistical power is the probability of statistically detecting a treatment effect when one truly exists.

"Obviously, an experiment designed [with low power] is not worth performing. Unfortunately, this particular point escaped the attention of a large number of authors of statistical texts...As a result, important experiments are often performed with the possibility of very frequent most regrettable errors."

Jerzy Neyman (1977)

- Why calculate power? Avoid waste.

5

BELIEF IN THE LAW OF SMALL NUMBERS

AMOS TVERSKY AND DANIEL KAHNEMAN¹

Hebrew University of Jerusalem

People have erroneous intuitions about the laws of chance. In particular, they regard a sample randomly drawn from a population as highly representative, that is, similar to the population in all essential characteristics. The prevalence of the belief and its unfortunate consequences for psychological research are illustrated by the responses of professional psychologists to a questionnaire concerning research decisions.

Research Article

Researchers' Intuitions About Power in Psychological Research



Marjan Bakker¹, Chris H. J. Hartgerink¹, Jelte M. Wicherts¹, and Han L. J. van der Maas²

¹Department of Methodology and Statistics, Tilburg School of Social and Behavioral Sciences, Tilburg University, and ²Department of Psychology, Psychological Methods, University of Amsterdam

aps
PSYCHOLOGICAL SCIENCE

Psychological Science
1-8
© The Author(s) 2016
Reprints and permission:
sagepub.com/journalsPermissions.nav
DOI: 10.1177/09567976166647519
ps.sagepub.com
SAGE

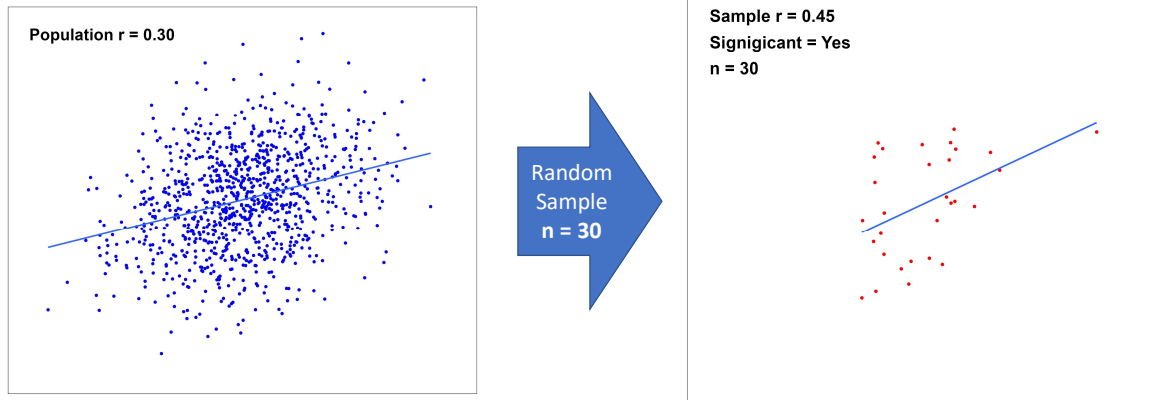
Do Studies of Statistical Power Have an Effect on the Power of Studies?

Peter Sedlmeier and Gerd Gigerenzer
University of Konstanz, Federal Republic of Germany

The long-term impact of studies of statistical power is investigated using J. Cohen's (1962) pioneering work as an example. We argue that the impact is nil; the power of studies in the same journal that Cohen reviewed (now the *Journal of Abnormal Psychology*) has not increased over the past 24 years. In 1960 the median power (i.e., the probability that a significant result will be obtained if there is a true effect) was .46 for a medium size effect, whereas in 1984 it was only .37. The decline of power is a result of alpha-adjusted procedures. Low power seems to go unnoticed: only 2 out of 64 experiments mentioned power, and it was never estimated. Nonsignificance was generally interpreted as confirmation of the null hypothesis (if this was the research hypothesis), although the median power was as low as .25 in these cases. We discuss reasons for the ongoing neglect of power.

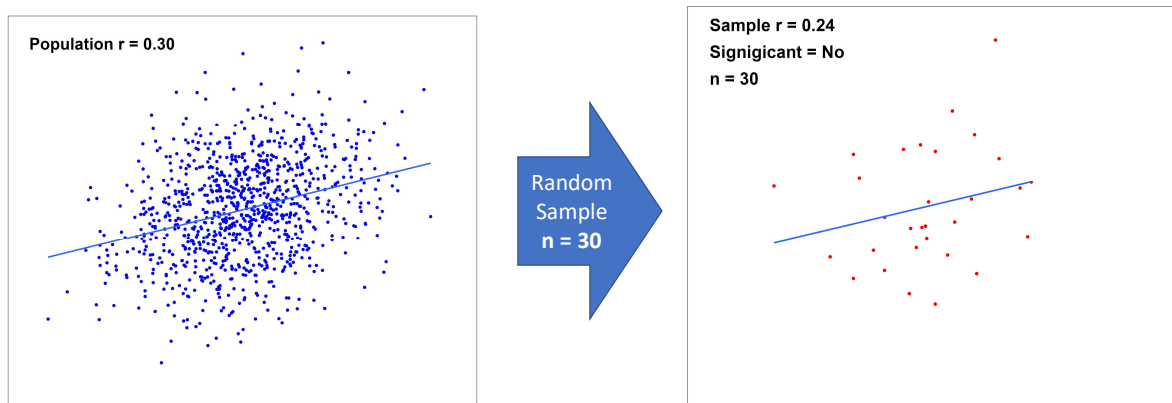
6

What is power analyses using simulation



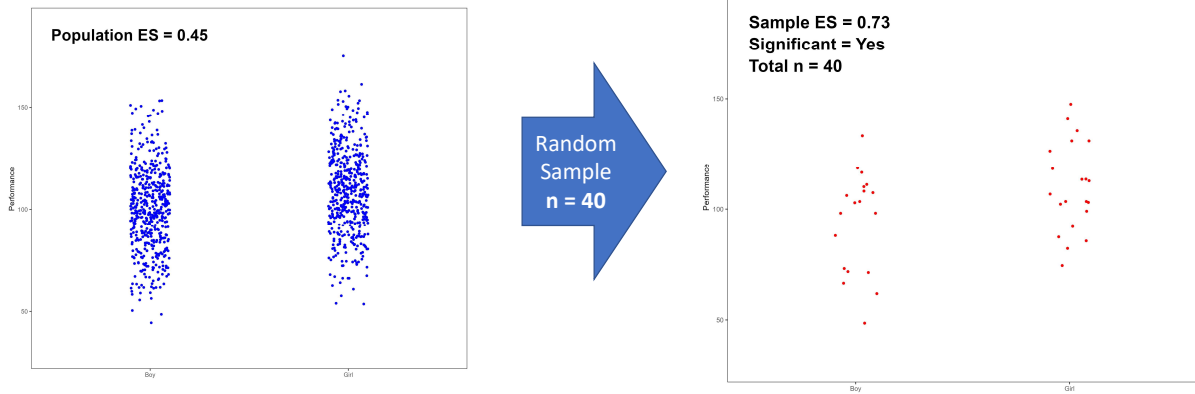
7

What is power analyses using simulation



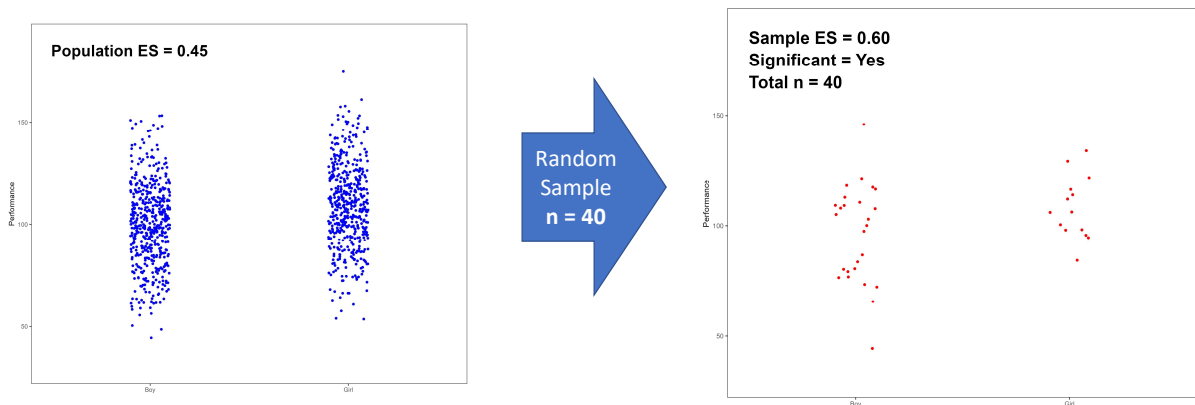
8

What is power analyses using simulation



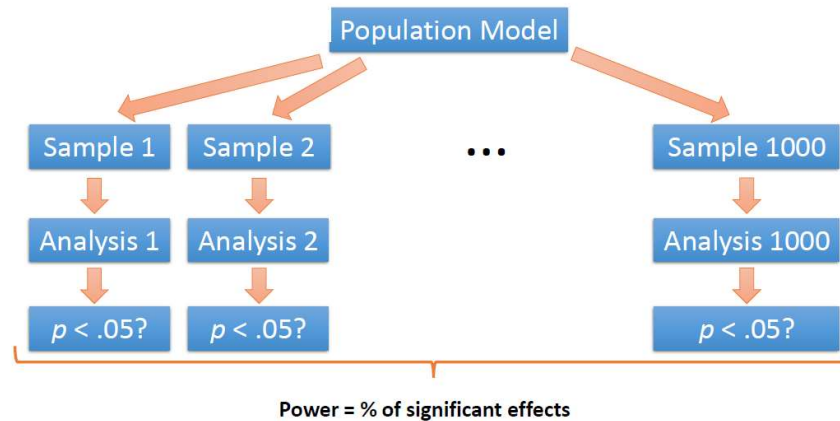
9

What is power analyses using simulation



10

What is power analyses using simulation

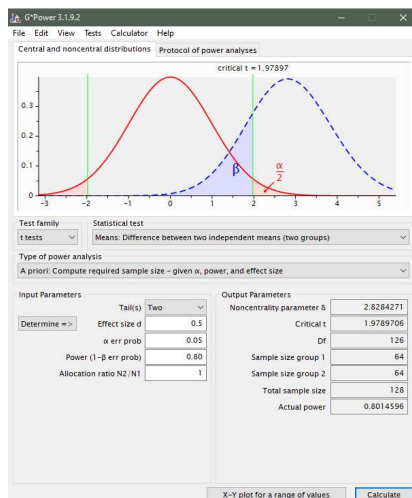


- If I were to repeat the study multiple times, how many of them would be significant?

(Lane, Hennes, & West, 2016)

11

Computing power: Analytical methods



```
install.packages("pwr")
library(pwr)
```

| function | power calculations for |
|-----------------------------|--|
| <code>pwr.2p.test</code> | two proportions (equal n) |
| <code>pwr.2p2n.test</code> | two proportions (unequal n) |
| <code>pwr.anova.test</code> | balanced one way ANOVA |
| <code>pwr.chisq.test</code> | chi-square test |
| <code>pwr.f2.test</code> | general linear model |
| <code>pwr.p.test</code> | proportion (one sample) |
| <code>pwr.r.test</code> | correlation |
| <code>pwr.t.test</code> | t-tests (one sample, 2 sample, paired) |
| <code>pwr.t2n.test</code> | t-test (two samples with unequal n) |

12

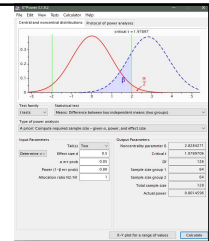
A note on the approach

- Common to:
 - Input the effect size, power needed, and alpha level
 - Obtain required sample size

- What we will do:
 - Input effect size, sample size, and alpha level
 - Obtain statistical power

Simpler/More intuitive

We can redo this multiple time and create a power curve.



13

Useful R Functions & Concepts Used Throughout the Workshop

14

Function: c() to combine

Sequence:

The expression `1:10` generates a sequence of numbers from 1 to 10, inclusive.

In R, the `c()` function is used to combine objects into a single vector or list.

Here are a few examples to illustrate the usage of the `c()` function:

```
vector <- c(1, 2, 3, 4, 5)
vector
```

```
## [1] 1 2 3 4 5
```

In this case, the `c()` function combines the individual elements 1, 2, 3, 4, and 5 into a single vector [1, 2, 3, 4, 5].

Combining multiple vectors into one:

```
vector1 <- c(1, 2, 3)
vector2 <- c(4, 5, 6)
combined_vector <- c(vector1, vector2)
combined_vector
```

```
## [1] 1 2 3 4 5 6
```

Here, the `c()` function combines the elements of `vector1` and `vector2` into a single vector [1, 2, 3, 4, 5, 6].

15

Function: rep() to replicate

The `rep()` function is used to create repeated copies of a vector, a value, or a set of values.

When using *times*, you specify the total number of repetitions of the entire object or vector.

```
vector <- c(1, 2, 3)
rep(vector, times = 3)
```

```
## [1] 1 2 3 1 2 3 1 2 3
```

When using *each*, you specify the number of repetitions for each element in the object or vector.

```
vector <- c(1, 2, 3)
rep(vector, each = 3)
```

```
## [1] 1 1 1 2 2 2 3 3 3
```

16

Exercises

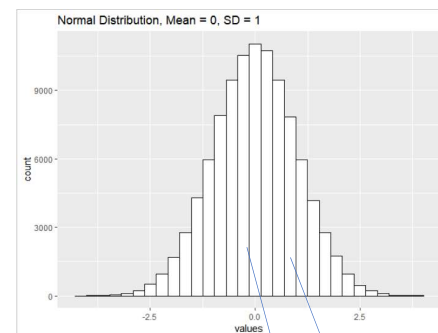
Exercises:

- Using the functions shown above, write the following sequences of numbers
- 1 2 1 2 1 2 1 2
- 1 1 2 2 1 1 2 2
- 1 2 3 4 1 2 3 4 5 6 7 8 5 6 7 8

17

Function: rnorm()

- In R, the **rnorm()** function is used to generate random samples from a normal distribution.



```
## {r}
rnorm(n = 10, mean = 0, sd = 1)
##
```

```
[1] 0.1916863 -0.3030520 0.6559188 0.9119106 -0.3496928 -0.3511093 -0.4980334 0.3911456 -1.1831866 -0.4123472
```

```
## {r}
rnorm(n = 10, mean = 0, sd = 1)
##
```

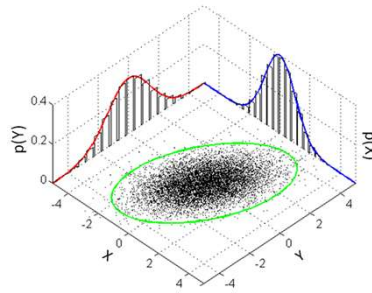
```
[1] 0.58037503 -1.22693306 1.79795871 -0.04016572 0.23934289 -1.67093158 -0.43333902 -0.15586945 0.20587191 0.42659279
```

18

From the MASS package

Function: mvrnorm()

- Generates random samples from a multivariate normal distribution.
 - A generalization of the normal distribution to multiple dimension.



- Useful to create correlated variables

19

Concept: the “for loop”

- Type of looping construct that is used to execute a block of code repeatedly a specified number of times

```

{r}
for (i in 1:10){
  print("hello")
}

```

```

[1] "hello"
[1] "hello"
[1] "hello"
[1] "hello"
[1] "hello"
[1] "hello"
[1] "hello"
[1] "hello"
[1] "hello"
[1] "hello"

```

20

Storing information in a “for loop”

- We can use a variable to store the information as the loop iterates.

```
container <- NULL #Creating an empty container
for(i in 1:10){
  container <- c(container, i)
}
container #Here I print the values contained in the variable "container".
```

```
##{r}
sum(container <= 5)/length(container)

[1] 0.5
```

21

Extracting information from an R output

- library(broom)

```
v1 <- rnorm(n = 10000, mean = 0, sd = 1)
v2 <- rnorm(n = 10000, mean = 0, sd = 1)
cor.test(v1,v2, method = "pearson")
...

Pearson's product-moment correlation

data:  v1 and v2
t = 0.60212, df = 9998, p-value = 0.5471
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.01357995  0.02561876
sample estimates:
      cor 
0.006021718
```

```
v1 <- rnorm(n = 10000, mean = 0, sd = 1)
v2 <- rnorm(n = 10000, mean = 0, sd = 1)
tidy(cor.test(v1,v2, method = "pearson"))$p.value[1]
...

[1] 0.5471062
```

22

Example 1: Independent Sample t-test

| ID | groupID | score |
|----|---------|-------|
| 1 | gr1 | 0.00 |
| 2 | gr1 | 0.11 |
| 3 | gr1 | 1.98 |
| 4 | gr1 | 0.39 |
| 5 | gr1 | 0.29 |
| 6 | gr1 | -1.03 |
| 7 | gr1 | 0.69 |
| 8 | gr1 | 0.26 |
| 9 | gr1 | 1.19 |
| 10 | gr1 | 0.03 |
| 11 | gr1 | -0.72 |
| 12 | gr1 | -0.73 |
| 13 | gr1 | 0.49 |
| 14 | gr1 | 0.56 |
| 15 | gr1 | 0.78 |
| 16 | gr2 | 0.22 |
| 17 | gr2 | 0.80 |
| 18 | gr2 | 0.04 |
| 19 | gr2 | -0.71 |
| 20 | gr2 | 0.39 |
| 21 | gr2 | 0.44 |
| 22 | gr2 | -0.87 |
| 23 | gr2 | 0.68 |
| 24 | gr2 | 0.54 |
| 25 | gr2 | 1.24 |
| 26 | gr2 | 1.19 |
| 27 | gr2 | 1.16 |
| 28 | gr2 | 0.70 |
| 29 | gr2 | -1.10 |
| 30 | gr2 | -1.38 |

```
#Sample size
n_per_group <- 15

#Parameters
MeanGr1 <- 0
SDGr1 <- 1
MeanGr2 <- 0.2
SDGr2 <- 1

#Container
pvalues <- NULL

group1 <- rnorm(n = n_per_group, mean = MeanGr1, sd = SDGr1)
group2 <- rnorm(n = n_per_group, mean = MeanGr2, sd = SDGr2)

score <- c(group1, group2)
groupID <- rep(c("gr1", "gr2"), each = n_per_group)

MyData <- data.frame(groupID, score)

t_model <- t.test(score ~ groupID, data = MyData)
pvalues <- c(pvalues, tidy(t_model)$p.value[1])
```

23

Exercises: Independent Sample t-test

1. Modify the example above to effect size of $d = 0.5$ and compare what you obtain with G*Power.
2. Increase the sample size to 50 per group and see what happens to the power?
3. What happens to power if the sample size of the two groups is very different?
4. Modify the example above to conduct a Wilcoxon rank-sum test (non-parametric version of the t-test)
5. Increase the number of simulated examples from 1000 to 10000 for more accurate results (but a longer running time).
6. Compute the power if the null hypothesis is true (i.e., the means of both groups are equal)
7. Draw a histogram of the p-value obtained when the null hypothesis is true.

24

Most Rigorous Large-Scale Educational RCTs Are Uninformative: Should We Be Concerned?

Hugues Lortie-Forgues¹ and Matthew Inglis²

There are a growing number of large-scale educational randomized controlled trials (RCTs). Considering their expense, it is important to reflect on the effectiveness of this approach. We assessed the magnitude and precision of effects found in those large-scale RCTs commissioned by the UK-based Education Endowment Foundation and the U.S.-based National Center for Educational Evaluation and Regional Assistance, which evaluated interventions aimed at improving academic achievement in K-12 (141 RCTs; 1,222,024 students). The mean effect size was 0.06 standard deviations. These sat within relatively large confidence intervals (mean width = 0.30 SDs), which meant that the results were typically uninformative (the median Bayes factor was 0.68). We argue that our field needs, as a priority, to understand why educational RCTs typically find small and uninformative effects.

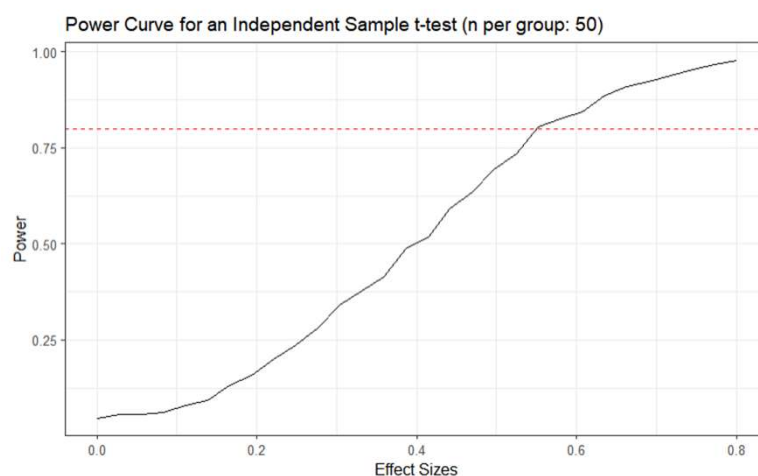
Keywords: educational policy; evaluation; meta-analysis; program evaluation

Lortie-Forgues, H., & Inglis, M. (2019). Rigorous large-scale educational RCTs are often uninformative: Should we be concerned?. *Educational Researcher*, 48(3), 158-166.

25

Power Curve

Also, possible to have sample size on the x-axis.



- See Appendix at the end of the RMarkdown document

26

Example 2: One-Way ANOVA

Independent Sample t-test

```
#Sample size
n_per_group <- 15

#Parameters
MeanGr1 <- 0
SDGr1 <- 1
MeanGr2 <- 0.2
SDGr2 <- 1

#Container
pvalues <- NULL

for(i in 1:1000){
  group1 <- rnorm(n = n_per_group, mean = MeanGr1, sd = SDGr1)
  group2 <- rnorm(n = n_per_group, mean = MeanGr2, sd = SDGr2)

  score <- c(group1, group2)
  groupID <- rep(c("gr1", "gr2"), each = n_per_group)

  MyData <- data.frame(groupID, score)

  t_model <- t.test(score ~ groupID, data = MyData)
  pvalues <- c(pvalues, tidy(t_model)$p.value[1])
}

#Power
sum(pvalues <= 0.05)/length(pvalues)
```

| ID | groupID | score |
|----|---------|-------|
| 1 | gr1 | 0.00 |
| 2 | gr1 | 0.11 |
| 3 | gr1 | 1.98 |
| 4 | gr1 | 0.39 |
| 5 | gr1 | 0.29 |
| 6 | gr1 | -1.03 |
| 7 | gr1 | 0.69 |
| 8 | gr1 | 0.26 |
| 9 | gr1 | 1.19 |
| 10 | gr1 | 0.03 |
| 11 | gr1 | -0.72 |
| 12 | gr1 | -0.73 |
| 13 | gr1 | 0.49 |
| 14 | gr1 | 0.56 |
| 15 | gr1 | 0.78 |
| 16 | gr2 | 0.22 |
| 17 | gr2 | 0.80 |
| 18 | gr2 | 0.04 |
| 19 | gr2 | -0.71 |
| 20 | gr2 | 0.39 |
| 21 | gr2 | 0.44 |
| 22 | gr2 | -0.87 |
| 23 | gr2 | 0.68 |
| 24 | gr2 | 0.54 |
| 25 | gr2 | 1.24 |
| 26 | gr2 | 1.19 |
| 27 | gr2 | 1.16 |
| 28 | gr2 | 0.70 |
| 29 | gr2 | -1.10 |
| 30 | gr2 | -1.38 |
| 31 | gr3 | 0.97 |
| 32 | gr3 | 1.17 |
| 33 | gr3 | 0.55 |
| 34 | gr3 | -0.43 |
| 35 | gr3 | 0.54 |
| 36 | gr3 | 1.24 |
| 37 | gr3 | -0.18 |
| 38 | gr3 | 0.92 |
| 39 | gr3 | 0.66 |
| 40 | gr3 | 1.63 |
| 41 | gr3 | 2.18 |
| 42 | gr3 | 1.63 |
| 43 | gr3 | 1.64 |
| 44 | gr3 | -0.91 |
| 45 | gr3 | -0.56 |

27

Example 2: One-Way ANOVA

Independent Sample t-test

```
#Sample size
n_per_group <- 15

#Parameters
MeanGr1 <- 0
SDGr1 <- 1
MeanGr2 <- 0.2
SDGr2 <- 1

#Container
pvalues <- NULL

for(i in 1:1000){
  group1 <- rnorm(n = n_per_group, mean = MeanGr1, sd = SDGr1)
  group2 <- rnorm(n = n_per_group, mean = MeanGr2, sd = SDGr2)

  score <- c(group1, group2)
  groupID <- rep(c("gr1", "gr2"), each = n_per_group)

  MyData <- data.frame(groupID, score)

  t_model <- t.test(score ~ groupID, data = MyData)
  pvalues <- c(pvalues, tidy(t_model)$p.value[1])
}

#Power
sum(pvalues <= 0.05)/length(pvalues)
```

One-Way ANOVA

```
#Sample Size
n_per_group <- 15

#Parameters
MeanGr1 <- 0
SDGr1 <- 1
MeanGr2 <- 0.2
SDGr2 <- 1
MeanGr3 <- 0
SDGr3 <- 1

#Container
pvalues <- NULL

for(i in 1:1000){
  group1 <- rnorm(n = n_per_group, mean = MeanGr1, sd = SDGr1)
  group2 <- rnorm(n = n_per_group, mean = MeanGr2, sd = SDGr2)
  group3 <- rnorm(n = n_per_group, mean = MeanGr3, sd = SDGr3)

  score <- c(group1, group2, group3)
  groupID <- as.factor(rep(c("gr1", "gr2", "gr3"), each = n_per_group))

  MyData <- data.frame(groupID, score)

  anova.model <- aov(score ~ groupID, data = MyData)

  pvalues <- c(pvalues, tidy(anova.model)$p.value[1])
}

sum(pvalues <= 0.05)/length(pvalues)
```

28

Exercises: One-Way ANOVA

- Explore how having widely different variance (or SD) between groups influences Type 1 error associated with a one-way ANOVA.
 - Note that homogeneity of variances (i.e., assumption of equal variances, or homoscedasticity assumption) is an assumption of one-way ANOVA.

29

Creating Correlated Variables

```
#Sample size
n_observation <- 100

#Parameters
r_v1v2 <- 0.9 # value of correlation between v1 and v2 in the population

pop_rs <- matrix(c(1.00, r_v1v2,
                  r_v1v2, 1.00), nrow = 2, ncol = 2)

allvar <- mvrnorm(n = n_observation, mu = c(0,0), pop_rs)
var1 <- allvar[,1]
var2 <- allvar[,2]

MyData <- data.frame(var1, var2)
cor.test(var1, var2, data = MyData)
...
```

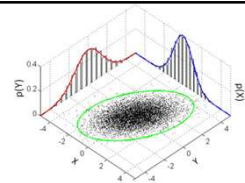
```
Pearson's product-moment correlation

data: var1 and var2
t = 19, df = 98, p-value <2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.84 0.92
sample estimates:
cor
0.89
```

Correlation matrix

| | var1 | var2 |
|------|------|------|
| var1 | 1 | 0.9 |
| var2 | 0.9 | 1 |

| ID | var1 | var2 |
|-----|------|------|
| 1 | 0.13 | 0.49 |
| 2 | 0.60 | 0.13 |
| 3 | 0.71 | 0.70 |
| 4 | 0.01 | 0.40 |
| 5 | 0.16 | 0.91 |
| 6 | 0.94 | 0.24 |
| ... | ... | ... |
| ... | ... | ... |
| 99 | 0.78 | 0.28 |
| 100 | 0.15 | 0.25 |



30

Creating Correlated Variables

```
#Sample size
n_observation <- 100

#Parameters
r_v1v2 <- 0.9 # value of correlation between v1 and v2 in the population

pop_rs <- matrix(c(1.00, r_v1v2,
                  r_v1v2, 1.00), nrow = 2, ncol = 2)

allvar <- mvrnorm(n = n_observation, mu = c(0,0), pop_rs)
var1 <- allvar[,1]
var2 <- allvar[,2]

MyData <- data.frame(var1, var2)

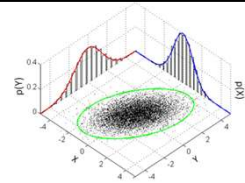
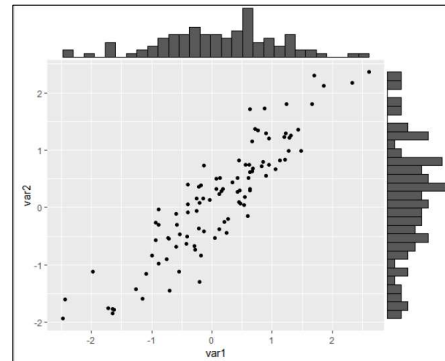
cor.test(var1,var2, data = MyData)
...
```

```
Pearson's product-moment correlation

data: var1 and var2
t = 19, df = 98, p-value <2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.84 0.92
sample estimates:
cor
0.89
```

Correlation matrix

| | var1 | var2 |
|------|------|------|
| var1 | 1 | 0.9 |
| var2 | 0.9 | 1 |



31

Example 3: Correlation

```
#Sample Size
n_observation <- 100

#Parameters
r_v1v2 <- 0.3
pop_rs <- matrix(c(1.00, r_v1v2,
                  r_v1v2, 1.00), nrow = 2, ncol = 2)

#Container
pvalues <- NULL

for(i in 1:1000){

  allvar <- mvrnorm(n = n_observation, mu = c(0,0), pop_rs)
  var1 <- allvar[,1]
  var2 <- allvar[,2]

  MyData <- data.frame(var1, var2)

  cor_model <- cor.test(var1, var2, method = "pearson", data = MyData)

  pvalues <- c(pvalues, tidy(cor_model)$p.value[1])
}

sum(pvalues <= 0.05)/length(pvalues)
...
```

[1] 0.86

32




Example 3: Correlation

Exercises:

- Instead of a Pearson correlation, use a Spearman correlation (i.e., **method = "spearman"**). Is the statistical power similar?
- What happens to the power if we dichotomise var2 using a median split? `var2 <- ifelse(var2 > median(var2), 1, 0)`
- What happens to the power if we introduce random noise in the measurement of a variable (i.e., reduce the reliability of a measure)?
- What is the mean value of correlations that have met the criteria for statistical significance? How does it compare to the correlation value in the population?

33

Quantifying "Promising Trials Bias" in Randomized Controlled Trials in Education

Sam Sims^a , Jake Anders^a , Matthew Inglis^b , and Hugues Lortie-Forgues^b 

^aUCL Centre for Education Policy & Equalising Opportunities, UCL Institute of Education, London, UK;

^bCentre for Mathematical Cognition, Loughborough University, Loughborough, UK

ABSTRACT

Randomized controlled trials have proliferated in education, in part because they provide an unbiased estimator for the causal impact of interventions. It is increasingly recognized that many such trials in education have low power to detect an effect if indeed there is one. However, it is less well known that low powered trials tend to systematically exaggerate effect sizes among the subset of interventions that show promising results ($p < \alpha$). We conduct a retrospective design analysis to quantify this bias across 22 such promising trials, finding that the estimated effect sizes are exaggerated by an average of 52% or more. Promising trial bias can be reduced ex-ante by increasing the power of the trials that are commissioned and guarded against ex-post by including estimates of the exaggeration ratio when reporting trial findings. Our results also suggest that challenges around implementation fidelity are not the only reason that apparently successful interventions often fail to subsequently scale up. Instead, the effect from the initial promising trial may simply be exaggerated.

ARTICLE HISTORY

Received 18 May 2020

Revised 25 April 2022

Accepted 6 May 2022

KEYWORDS

Randomized controlled trials; Type M error; Type S error

34

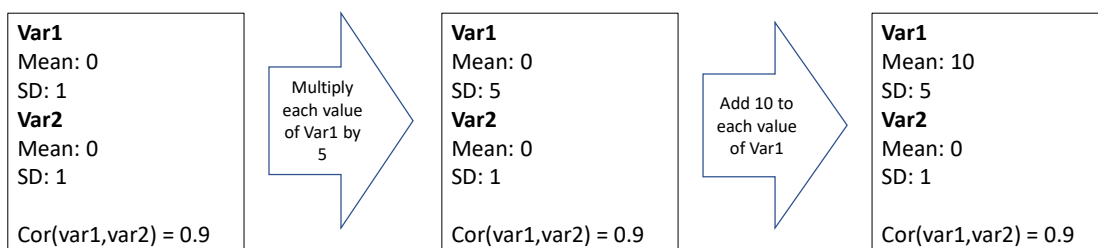
Example 4: Paired Sample t-test

- The correlation between repeated measurement matters

35

Example 4: Paired Sample t-test

- Invariance of the correlation coefficient under linear transformations.
 - Correlation coefficient remains unchanged when we apply linear transformations to the variables involved



Does not work if transformations are non-linear (e.g., square, square root, $1/x$) or multiply by 0.

36

Example 4: Paired Sample t-test

| ID | Time | Value |
|----|------|-------|
| 1 | T1 | -0.61 |
| 2 | T1 | -0.05 |
| 3 | T1 | -1.74 |
| 4 | T1 | -1.52 |
| 5 | T1 | 0.53 |
| 6 | T1 | 0.49 |
| 7 | T1 | 0.51 |
| 8 | T1 | -0.52 |
| 9 | T1 | -0.49 |
| 10 | T1 | 1.09 |
| 1 | T2 | -2.07 |
| 2 | T2 | 1.05 |
| 3 | T2 | -0.43 |
| 4 | T2 | 0.17 |
| 5 | T2 | 0.21 |
| 6 | T2 | 0.51 |
| 7 | T2 | 0.61 |
| 8 | T2 | 1.28 |
| 9 | T2 | -1.01 |
| 10 | T2 | 1.31 |

```
##Sample Size
n_participant <- 10

#Population Mean for Each Cell in the Design
mean_time1 <- 0
mean_time2 <- 0.5

#Common Population Standard Deviation
sd_all <- 1

#Common population correlation among within-subjects factors
r_within <- 0.5

pop_rs <- matrix(c(1.00, r_within,
                  r_within, 1.00), nrow = 2, ncol = 2)

#Container
pvalues <- NULL

for(i in 1:1000){

  allvar<-mvrnorm(n_participant, mu = c(0,0), pop_rs, empirical = FALSE)

  time1 <- (allvar[,1] * sd_all) + mean_time1
  time2 <- (allvar[,2] * sd_all) + mean_time2

  value <- c(time1, time2)

  condition <- rep(c("T1", "T2"), each = n_participant)
  ID <- as.factor(rep(1:n_participant, times = 2))

  MyData <- data.frame(ID, Condition, value)

  t_model <- t.test(value ~ Condition, data = MyData, paired = TRUE)
  pvalues <- c(pvalues, tidy(t_model)$p.value[1])

}

#Power
sum(pvalues <= 0.05)/length(pvalues)

...

[1] 0.28
```

37

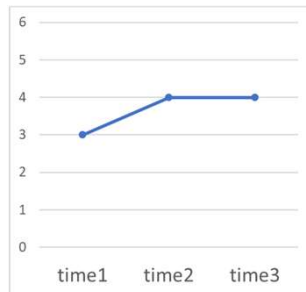
Example 4: Paired Sample t-test

Exercise:

- What happens to the power of the test when the correlation between the two time-point decreases?
- What if you incorrectly use an Independent Sample t-test to analyse this data?

38

Example 5: Repeated-measure ANOVA



```
#Sample Size
n_participant <- 10

#Population Mean for Each Cell in the Design
mean_time1 <- 3
mean_time2 <- 4
mean_time3 <- 4

#Common Population Standard Deviation
sd_all <- 1

#Common population correlation among within-subjects factors
r_within <- 0.2

rvalues <- matrix(c(1.00, r_within, r_within,
                  r_within, 1.00, r_within,
                  r_within, r_within, 1.00), nrow = 3, ncol = 3)

#Container
pvalues <- NULL

for(i in 1:100){

  allVar<-mvrnorm(n_participant, mu = c(0,0,0), rvalues, empirical = FALSE)
  time1 <- (allVar[,1] * sd_all) + mean_time1
  time2 <- (allVar[,2] * sd_all) + mean_time2
  time3 <- (allVar[,3] * sd_all) + mean_time3

  Value <- c(time1, time2, time3)
  Time <- rep(c("T1", "T2", "T3"), each = n_participant)
  ID <- as.factor(rep(1:n_participant, times = 3))

  MyData <- data.frame(ID, Time, Value)

  fitAnova<-aov(Value~Time + Error(ID/Time), data=MyData)
  pvalues <- c(pvalues, tidy(fitAnova)$p.value[2])

}

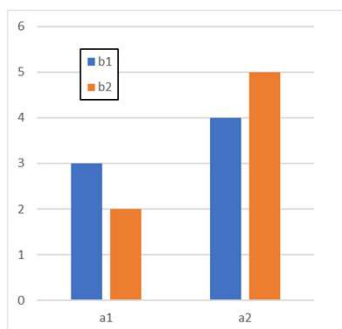
#Power
sum(pvalues <= 0.05)/length(pvalues)

...

[1] 0.67
```

39

Example 6: ANOVA with Two Between-Subject Variables



| ID | IV_a | IV_b | DV |
|-----|------|------|------|
| 1 | a1 | b1 | 0.06 |
| 2 | a1 | b1 | 0.49 |
| 3 | a1 | b1 | 0.78 |
| 4 | a1 | b1 | 0.82 |
| ... | ... | ... | ... |
| 11 | a1 | b2 | 0.11 |
| 12 | a1 | b2 | 0.05 |
| 13 | a1 | b2 | 0.12 |
| 14 | a1 | b2 | 0.37 |
| ... | ... | ... | ... |
| 21 | a2 | b1 | 0.66 |
| 22 | a2 | b1 | 0.17 |
| 23 | a2 | b1 | 0.91 |
| 24 | a2 | b1 | 0.10 |
| ... | ... | ... | ... |
| 31 | a2 | b2 | 0.67 |
| 32 | a2 | b2 | 0.40 |
| 33 | a2 | b2 | 0.24 |
| 34 | a2 | b2 | 0.23 |
| ... | ... | ... | ... |

```
#Sample Size
n_per_cell <- 10

#Population Mean for Each Cell in the Design
mean_a1b1 <- 3
mean_a1b2 <- 2
mean_a2b1 <- 4
mean_a2b2 <- 5

#Common Population Standard Deviation
sd_all <- 1

#Container
pvalues_IV_a <- NULL
pvalues_IV_b <- NULL
pvalues_Interaction <- NULL

for(i in 1:100){

  IV_a <- as.factor(rep(c("a1", "a2"), each = n_per_cell*2))
  IV_b <- as.factor(rep(c("b1", "b2", "b1", "b2"), each = n_per_cell))

  a1b1 <- rnorm(n_per_cell, mean = mean_a1b1, sd = sd_all)
  a1b2 <- rnorm(n_per_cell, mean = mean_a1b2, sd = sd_all)
  a2b1 <- rnorm(n_per_cell, mean = mean_a2b1, sd = sd_all)
  a2b2 <- rnorm(n_per_cell, mean = mean_a2b2, sd = sd_all)

  DV <- c(a1b1, a1b2, a2b1, a2b2)

  MyData <- data.frame(IV_a, IV_b, DV)

  fitAnova <- aov(DV ~ IV_a * IV_b, data = MyData)

  pvalues_IV_a <- c(pvalues_IV_a, tidy(fitAnova)$p.value[1])
  pvalues_IV_b <- c(pvalues_IV_b, tidy(fitAnova)$p.value[2])
  pvalues_Interaction <- c(pvalues_Interaction, tidy(fitAnova)$p.value[3])

}

#Power IV a
sum(pvalues_IV_a <= 0.05)/length(pvalues_IV_a)
#Power IV b
sum(pvalues_IV_b <= 0.05)/length(pvalues_IV_b)
#Power Interaction a x b
sum(pvalues_Interaction <= 0.05)/length(pvalues_Interaction)

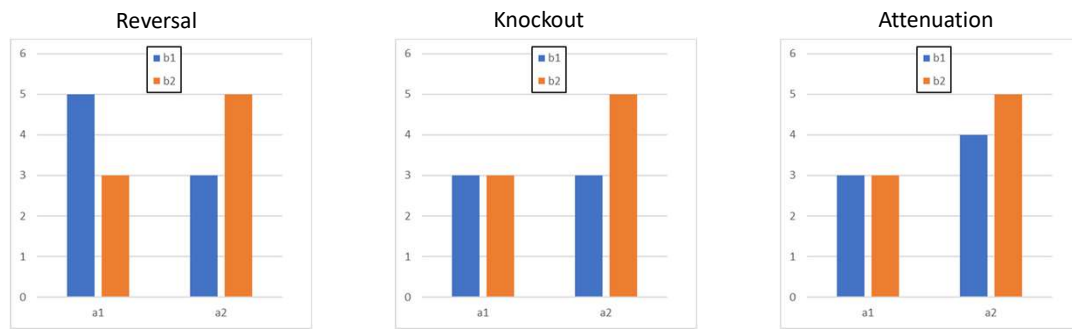
...

[1] 1
[1] 0.06
[1] 0.89
```

40

Power for statistical interactions

- Power needed depends on the shape of the interaction



Higher Power
(Smaller N needed)



<https://approachingblog.wordpress.com/2018/01/24/powering-your-interaction-2/>
<http://datacolada.org/17>

41

Power for statistical interactions

Statistical Modeling, Causal Inference, and Social Science

Home Authors Blogs We Read Sponsors

Here's the title of my talk at the New York R conference, 20 Apr 2018: Gaydar and the fallacy of objective measurement

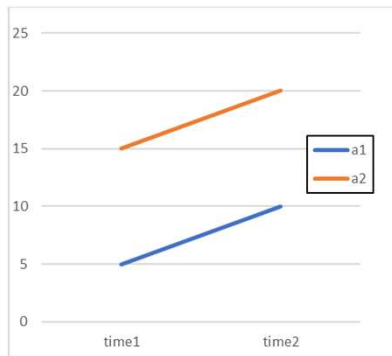
You need 16 times the sample size to estimate an interaction than to estimate a main effect

Posted on March 15, 2018 9:11 AM by Andrew

<https://statmodeling.stat.columbia.edu/2018/03/15/need-16-times-sample-size-estimate-interaction-estimate-main-effect/>

42

Example 7: Mixed-design ANOVA



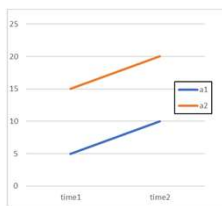
| ID | IV_a | Time | DV |
|-----|------|-------|------|
| 1 | a1 | time1 | 0.44 |
| 2 | a1 | time1 | 0.16 |
| 3 | a1 | time1 | 0.57 |
| 4 | a1 | time1 | 0.31 |
| ... | ... | ... | ... |
| 1 | a1 | time2 | 0.53 |
| 2 | a1 | time2 | 0.62 |
| 3 | a1 | time2 | 0.89 |
| 4 | a1 | time2 | 0.89 |
| ... | ... | ... | ... |
| 11 | a2 | time1 | 0.63 |
| 12 | a2 | time1 | 0.60 |
| 13 | a2 | time1 | 0.61 |
| 14 | a2 | time1 | 0.74 |
| ... | ... | ... | ... |
| 11 | a2 | time2 | 0.81 |
| 12 | a2 | time2 | 0.59 |
| 13 | a2 | time2 | 0.89 |
| 14 | a2 | time2 | 0.26 |
| ... | ... | ... | ... |

The correlation matrix

| | A1_T1 | A1_T2 | A2_T1 | A2_T2 |
|-------|-------|-------|-------|-------|
| A1_T1 | | | | |
| A1_T2 | | | | |
| A2_T1 | | | | |
| A2_T2 | | | | |

43

Example 7: Mixed-design ANOVA



| ID | IV_a | Time | DV |
|-----|------|-------|------|
| 1 | a1 | time1 | 0.44 |
| 2 | a1 | time1 | 0.16 |
| 3 | a1 | time1 | 0.57 |
| 4 | a1 | time1 | 0.31 |
| ... | ... | ... | ... |
| 1 | a1 | time2 | 0.53 |
| 2 | a1 | time2 | 0.62 |
| 3 | a1 | time2 | 0.89 |
| 4 | a1 | time2 | 0.89 |
| ... | ... | ... | ... |
| 11 | a2 | time1 | 0.63 |
| 12 | a2 | time1 | 0.60 |
| 13 | a2 | time1 | 0.61 |
| 14 | a2 | time1 | 0.74 |
| ... | ... | ... | ... |
| 11 | a2 | time2 | 0.81 |
| 12 | a2 | time2 | 0.59 |
| 13 | a2 | time2 | 0.89 |
| 14 | a2 | time2 | 0.26 |
| ... | ... | ... | ... |

```
#Sample Size
n_per_cell <- 10
n_total <- 20

#Population Mean for Each Cell in the Design
mean_a1_Time1 <- 5
mean_a1_Time2 <- 10
mean_a2_Time1 <- 15
mean_a2_Time2 <- 20

#Common correlation among within-subjects factors
corr <- 0.2

#Common Standard Deviation
sd_var <- 10

#Containers
pvalueBetween <- NULL
pvalueWithin <- NULL
pvalueInteraction <- NULL
```

```
For(i in 1:100){
  rvalues <- matrix(c(1.00, corr, 0.00, 0.00,
                    corr, 1.00, 0.00, 0.00,
                    0.00, 0.00, 1.00, corr,
                    0.00, 0.00, corr, 1.00), nrow = 4, ncol = 4)

  allVar<-mvrnorm(n_per_cell, mu = c(0,0,0,0), rvalues, empirical = FALSE)

  a1_Time1 <- (allVar[,1] * sd_var) + mean_a1_Time1
  a1_Time2 <- (allVar[,2] * sd_var) + mean_a1_Time2
  a2_Time1 <- (allVar[,3] * sd_var) + mean_a2_Time1
  a2_Time2 <- (allVar[,4] * sd_var) + mean_a2_Time2

  Group <- as.factor(rep(c("A1", "A2"), each = n_per_cell*2))

  Time <- as.factor(rep(c("T1", "T2", "T1", "T2"), each = n_per_cell))

  Value <- c(a1_Time1, a1_Time2, a2_Time1, a2_Time2)

  #This creates a vector of 40 numbers: c(1 to 10, 1 to 10, 11 to 20, 11 to 20)
  ID <- as.factor(c(rep(1:n_per_cell, times = 2), rep((n_per_cell+1):n_total, times = 2)))

  MyData <- data.frame(ID, Group, Time, Value)

  rptAnova<-aov(Value~(Group*Time)+Error(ID/Time), data=MyData)

  pvalueBetween <- c(pvalueBetween, tidy(rptAnova)$p.value[1])
  pvalueWithin <- c(pvalueWithin, tidy(rptAnova)$p.value[3])
  pvalueInteraction <- c(pvalueInteraction, tidy(rptAnova)$p.value[4])
}

#Power Between Factor A
sum(pvalueBetween <= 0.05)/length(pvalueBetween)
#Power Within Factor Time
sum(pvalueWithin <= 0.05)/length(pvalueWithin)
#Power Interaction A x Time
sum(pvalueInteraction <= 0.05)/length(pvalueInteraction)

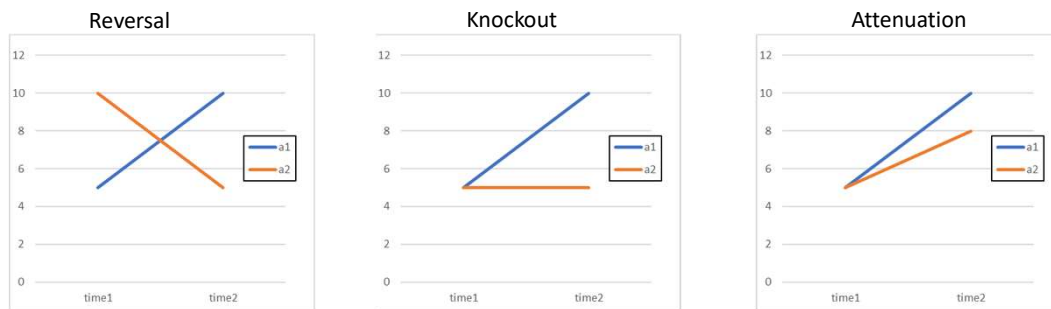
...

[1] 0.83
[1] 0.47
[1] 0.05
```

44

Power for statistical interactions

- Power needed depends on the shape of the interaction



Higher Power
(Smaller N needed)



<https://approachingblog.wordpress.com/2018/01/24/powering-your-interaction-2/>
<http://datacolada.org/17>

45

Example 8: Simple Regression

```
Call:
lm(formula = y ~ x, data = MyData)

Residuals:
    Min       1Q   Median       3Q      Max
-3.4233 -0.6666 -0.0092  0.6773  3.6651

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.0000     0.0100   100.00  <2e-16 ***
x            0.2981     0.0100    29.81  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1 on 9998 degrees of freedom
Multiple R-squared:  0.08161, Adjusted R-squared:  0.08152
F-statistic: 888.4 on 1 and 9998 DF, p-value: < 2.2e-16
```

```
#Sample Size
n_observation <- 82

#Parameters
b0 <- 1 #Intercept
b1 <- 0.3 #Slope
res_se <- 1 #Residual Standard Deviation (or Residual Standard Error)

#Container
pvalues_b0 <- NULL
pvalues_b1 <- NULL

for(i in 1:1000){
  x <- rnorm(n = n_observation, mean = 0, sd = 1)
  y <- b0 + b1*x + rnorm(n = n_observation, mean = 0, sd = res_se)

  MyData <- data.frame(x, y)

  fit <- lm(y ~ x, data = MyData)
  pvalues_b0 <- c(pvalues_b0, tidy(fit)$p.value[1])
  pvalues_b1 <- c(pvalues_b1, tidy(fit)$p.value[2])
}

#Power B0
sum(pvalues_b0 <= 0.05)/length(pvalues_b0)

#Power B1
sum(pvalues_b1 <= 0.05)/length(pvalues_b1)
...

[1] 1
[1] 0.756
```

46

Example 9: Multiple Regression

```
Call:
lm(formula = y ~ x1 * x2, data = MyData)

Residuals:
    Min       1Q   Median       3Q      Max
-3.998 -0.668  0.004  0.673  4.258

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.00455    0.00317   317.3  <2e-16 ***
x1           0.19717    0.00317    62.3  <2e-16 ***
x2           0.30365    0.00317    95.9  <2e-16 ***
x1:x2        0.39991    0.00316   126.6  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1 on 99996 degrees of freedom
Multiple R-squared:  0.225,    Adjusted R-squared:  0.225
F-statistic: 9.69e+03 on 3 and 99996 DF,  p-value: <2e-16
```

```
##Sample Size
n_observation <- 80

#population parameters
b0 <- 1 #Intercept
b1 <- 0.2 #Slope b1
b2 <- 0.3 #Slope b2
b3 <- 0.4 #Slope b3 (interaction)
res_se <- 1 #Residual Standard Deviation (or Residual Standard Error)

#containers
pvalues_b0 <- NULL
pvalues_b1 <- NULL
pvalues_b2 <- NULL
pvalues_b3 <- NULL

for(i in 1:100){
  x1 <- rnorm(n_observation, mean = 0, sd = 1)
  x2 <- rnorm(n_observation, mean = 0, sd = 1)
  y <- b0 + b1*x1 + b2*x2 + b3*x1*x2 + rnorm(n_observation, mean = 0, sd = res_se)

  MyData <- data.frame(x1, x2, y)

  fit <- lm(y ~ x1 * x2, data = MyData)

  pvalues_b0 <- c(pvalues_b0, tidy(fit)$p.value[1])
  pvalues_b1 <- c(pvalues_b1, tidy(fit)$p.value[2])
  pvalues_b2 <- c(pvalues_b2, tidy(fit)$p.value[3])
  pvalues_b3 <- c(pvalues_b3, tidy(fit)$p.value[4])
}

#Power B0
sum(pvalues_b0 <= 0.05)/length(pvalues_b0)

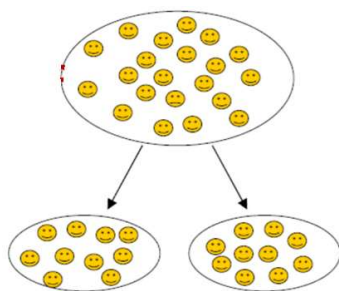
#Power B1
sum(pvalues_b1 <= 0.05)/length(pvalues_b1)

#Power B2
sum(pvalues_b2 <= 0.05)/length(pvalues_b2)

#Power B3 (interaction)
sum(pvalues_b3 <= 0.05)/length(pvalues_b3)
```

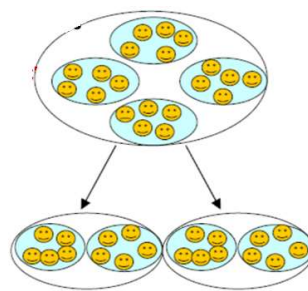
47

Example 10: Regression with Nested Data



Intervention group

Control group



Intervention group

Control group

| ID | Condition_ID | Cluster_ID | DV |
|------|--------------|------------|------|
| 1 | 1 | 1 | 0.34 |
| 2 | 1 | 1 | 0.37 |
| 3 | 1 | 1 | 0.60 |
| 4 | 1 | 1 | 0.42 |
| ... | ... | ... | ... |
| 1997 | 1 | 20 | 0.52 |
| 1998 | 1 | 20 | 0.41 |
| 1999 | 1 | 20 | 0.75 |
| 2000 | 1 | 20 | 0.50 |
| 2001 | 2 | 21 | 0.56 |
| 2002 | 2 | 21 | 0.22 |
| 2003 | 2 | 21 | 0.16 |
| 2004 | 2 | 21 | 0.58 |
| ... | ... | ... | ... |
| 3997 | 2 | 40 | 0.26 |
| 3998 | 2 | 40 | 0.14 |
| 3999 | 2 | 40 | 0.99 |
| 4000 | 2 | 40 | 0.05 |

Each participant score is determined by:

- The condition they are in (either 1 or 2): Condition effect
- The cluster they belong to (ranging from 1 to 40): Cluster effect
- Their own individual performance: Participant effect

```
##Sample Size
N_cluster <- 40
N_per_cluster <- 100
N_Total <- N_per_cluster * N_cluster

#Parameters
effectSD <- 0.45
var_cluster <- 1
var_error <- 2
totalSD <- sqrt(var_cluster + var_error)
#population ICC
#var_cluster/(var_cluster + var_error)

#Containers
pvalues <- NULL
```

48

Example 10: Regression with Nested Data

Intraclass correlation coefficient (ICC)
The ICC represents the proportion of total variation that is due to the differences between clusters.

| ID | Condition_ID | Cluster_ID | DV |
|------|--------------|------------|------|
| 1 | 1 | 1 | 0.34 |
| 2 | 1 | 1 | 0.37 |
| 3 | 1 | 1 | 0.60 |
| 4 | 1 | 1 | 0.42 |
| ... | ... | ... | ... |
| 1997 | 1 | 20 | 0.52 |
| 1998 | 1 | 20 | 0.41 |
| 1999 | 1 | 20 | 0.75 |
| 2000 | 1 | 20 | 0.50 |
| 2001 | 2 | 21 | 0.56 |
| 2002 | 2 | 21 | 0.22 |
| 2003 | 2 | 21 | 0.16 |
| 2004 | 2 | 21 | 0.58 |
| ... | ... | ... | ... |
| 3997 | 2 | 40 | 0.26 |
| 3998 | 2 | 40 | 0.14 |
| 3999 | 2 | 40 | 0.99 |
| 4000 | 2 | 40 | 0.05 |

```
#Sample Size
N_cluster <- 40
N_per_cluster <- 100
N_Total <- N_per_cluster * N_cluster

#Parameters
effectSD <- 0.45
var_cluster <- 1
var_error <- 2
totalSD <- sqrt(var_cluster + var_error)
#population ICC
#var_cluster/(var_cluster + var_error)

#Containers
pvalues <- NULL
```

```
for(i in 1:1000){#Increase for more accurate results (but a MUCH longer running time).

#Condition Effect
condition_ID <- as.factor(rep(c(1, 2), each = N_Total/2))
condition_effect <- rep(c(0, (effectSD * totalSD)), each = N_Total/2)

#Cluster Effect
cluster_ID <- as.factor(rep(1:N_cluster, each = N_per_cluster))
cluster_effect_values <- rnorm(n = N_cluster, mean = 0, sd = sqrt(var_cluster))
cluster_effect <- rep(cluster_effect_values, each = N_per_cluster)

#Participant Effect
individual_effect <- rnorm(n = N_Total, mean = 0, sd = sqrt(var_error))

#Dependent Variable
y <- condition_effect + cluster_effect + individual_effect

#Analysis
MyData <- data.frame(condition_ID,
                     cluster_ID,
                     y)

fit.rdm <- lmer(y ~ condition_ID + (1|cluster_ID), data = MyData)
pvalues <- c(pvalues, summary(fit.rdm)$coefficients[2,5])
}

#Power Condition Effect
sum(pvalues<=0.05)/length(pvalues)
```

49

RECAP

- What is data simulation
- What is power analysis
- What is power analysis using simulation
- Important concepts and R functions for the workshop
- Examples
 1. Independent Sample t-test
 2. One-Way ANOVA
 3. Correlation
 4. Paired Sample t-test
 5. Repeated-measure ANOVA (3 Levels)
 6. Factorial ANOVA (Two Between-Subject Variables)
 7. Mixed-design ANOVA (One Between-Subject and One Within-Subject Variable).
 8. Simple Regression
 9. Multiple Regression with Interaction
 10. Regression with Nested Data Using a Mixed-Effect Model

50

R packages

Gpower or the R package pwr

Independent Sample t-test
One-Way ANOVA
Correlation
Paired Sample t-test

R package Superpower (Lakens & Caldwell, 2022)

Repeated-measure ANOVA
Factorial ANOVA
Mixed-design ANOVA

PANGAEA: Power ANalysis for GEneral Anova designs (Jake Westfall)

Regression with Nested Data Using a Mixed-Effect Model

51

Other useful functions

- The **sample()** function is used to randomly sample elements from a given vector or set
- The **set.seed()** function is used to initialize the random number generator with a specific seed value.
 - You will always obtain the same values.

```
'''{r}
# Sample 3 random elements from a vector
vector <- c(1, 2, 3, 4, 5)
sampled <- sample(vector, 3)
sampled
'''
```

```
[1] 2 3 5
```

```
'''{r}
# Sample 3 random elements from a vector
vector <- c(1, 2, 3, 4, 5)
sampled <- sample(vector, 3, replace = TRUE)
sampled
'''
```

```
[1] 2 5 5
```

```
'''{r}
set.seed(123)
random_numbers <- rnorm(5)
print(random_numbers)
'''
```

```
[1] -0.56047565 -0.23017749 1.55870831 0.07050839 0.12928774
```

52

How to increase the power of your study?

- Increase sample size.
- Other ways:
 - Reduce measurement error (better measures, more raters)
 - Control for covariates associated with the outcome
 - Select extreme groups
- <https://janhove.github.io/analysis/2017/10/24/increasing-power-precision>
- <https://janhove.github.io/design/2017/07/14/OtherRoadsToPower>
- <https://janhove.github.io/design/2014/11/18/neglected-covariates>