Introduction to the Design and Analysis of Randomized Experiments
Class 2: Estimators and Tests, Randomization-Based Statistical Inference

Jake Bowers

July 30, 2024

Overview and Review

Today

0. Quiz and Questions
1. Some randomized designs: simple (coin flipping), complete (urn-drawing), factorial, blocked or stratified, clustered.
2. Statistical inference for causal effects in randomized experiments via the Fisher and Neyman approaches (Paul R Rosenbaum, 2010, Chap 2), (Gerber and Green, 2012, Chap 1-3):
   - Testing
   - Estimation

Tomorrow: More on statistical power.

Note: You can download (and contribute to) course materials at https://github.com/bowers-illinois-edu/short_course_experiments

Hay recursos en español aqui: https://egap.github.io/theory_and_practice_of_field_experiments_spanish/

Lingering Questions?

Questions arising?

Quiz

- What are key features of a randomized experiment? (What would we need to see in a research design to call it a randomized experiment?)

Quiz

- What are key features of a randomized experiment? (What would we need to see in a research design to call it a randomized experiment?)
- Why would social scientists want to use a randomized experiment? (How might a randomized experiment relate to one or more theoies?)

Quiz

- What are key features of a randomized experiment? (What would we need to see in a research design to call it a randomized experiment?)
- Why would social scientists want to use a randomized experiment? (How might a randomized experiment relate to one or more theoies?)
- Why would policy makers want to use a randomized experiment?

Quiz

- What are key features of a randomized experiment? (What would we need to see in a research design to call it a randomized experiment?)
- Why would social scientists want to use a randomized experiment? (How might a randomized experiment relate to one or more theoies?)
- Why would policy makers want to use a randomized experiment?
- Someone says a confusing statement, "I love experiments in my studies because I know that the treatment group income is exactly the same as the control group income. Plus I randomize so I know that my groups are representative." (What is wrong with this statement? How to help this person say it better? Como debemos hablar de covariables y lo que garantiza un experimento aleatorio y lo que no garantiza.)

Quiz

- What are key features of a randomized experiment? (What would we need to see in a research design to call it a randomized experiment?)
- Why would social scientists want to use a randomized experiment? (How might a randomized experiment relate to one or more theoies?)
- Why would policy makers want to use a randomized experiment?
- Someone says a confusing statement, "I love experiments in my studies because I know that the treatment group income is exactly the same as the control group income. Plus I randomize so I know that my groups are representative." (What is wrong with this statement? How to help this person say it better? Como debemos hablar de covariables y lo que garantiza un experimento aleatorio y lo que no garantiza.)
- Why would we like randomized experiments to have large N? (People often say "large samples" but often my experiments are not samples, they are the whole population, or a just a set of people with no known sampling plan.)

Quiz

- What are key features of a randomized experiment? (What would we need to see in a research design to call it a randomized experiment?)
- Why would social scientists want to use a randomized experiment? (How might a randomized experiment relate to one or more theoies?)
- Why would policy makers want to use a randomized experiment?
- Someone says a confusing statement, "I love experiments in my studies because I know that the treatment group income is exactly the same as the control group income. Plus I randomize so I know that my groups are representative." (What is wrong with this statement? How to help this person say it better? Como debemos hablar de covariables y lo que garantiza un experimento aleatorio y lo que no garantiza.)
- Why would we like randomized experiments to have large N? (People often say "large samples" but often my experiments are not samples, they are the whole population, or a just a set of people with no known sampling plan.)
- When I write, "Each person has two potential outcomes, $Y_i(T_i = 1), Y_i(T_i = 0)$" what am I assuming?

Some Randomized Designs

Simple randomization (coin-flipping)

- For each unit, flip a coin to see if it will be treated. Then you measure outcomes at the same level as the coin.
- The coins don't have to be fair (50-50), but you have to know the probability of treatment assignment.
- You can't guarantee a specific number of treated units and control units.
- Example: If you have 6 units and you flip a fair coin for each, you have about a 3% chance of assigning **all** units to treatment or assigning **all** units to control.

Example code for simple randomization I

```r
# set the random number seed to make this replicable
set.seed(12345)

# set a sample size
N <- 200

# Generate the simple random assignment
# (Notice that in an experiment we have a single
# trial and thus size=1)
# Our object with N people total is called simple.ra
simple.ra <- rbinom(n = N, size = 1, prob = .5)

# 112 people ended up in the treatment group
sum(simple.ra)
```

```
## [1] 112
```

Example code for simple randomization II

```r
# you can also use the randomizr package
library(randomizr)

# for replicability
set.seed(23456)
# Simple random assignment uses the simple_ra function
# Our object with N people total is called treatment
treatment <- simple_ra(
  N = N, # total sample size
  prob = .5 # probability of receiving treatment
)
sum(treatment)
```

```
## [1] 96
```

Complete randomization (drawing from an urn)

- A fixed number m out of N units are assigned to treatment.
- The probability a unit is assigned to treatment is m/N.
- This is like having an urn or bowl or bag with N balls, of which m are marked as treatment and $N - m$ are marked as control. Public lotteries use this method.

Example code for complete randomization I

```r
# set sample size N
N <- 200
# set number of treated units m
m <- 100

# create a vector of m 1's and N-m 0's
complete.ra <- c(rep(1, m), rep(0, N - m))

# And then scramble it randomly using sample()
# The default is sampling without replacement

set.seed(12345) # for replicability
complete.ra <- sample(complete.ra)

sum(complete.ra)
```

```
## [1] 100
```

Example code for complete randomization II

```r
# you can also use the randomizr package
library(randomizr)

# for replicability
set.seed(23456)

# Complete random assignment:
treatment <- complete_ra(
  N = 200, # total sample size
  m = 100
) # number to assign to treatment

sum(treatment)
```

```
## [1] 100
```

```r
# note what happens if you don't specify m!
```

Should always give you m treated.

Block (or stratified) randomization I

- We create blocks of units and randomize separately within each block. We are doing mini-experiments in each block.
  - Example: block = district, units = communities. We randomize treatment at the community level **within district** and also measure outcomes at the community level.
- Blocks that represent a substantively meaningful subgroup can help you to learn about how effects might differ by subgroup.
  - By controlling number of subjects per subgroup, you ensure that you have enough subjects in each group.
  - Especially useful when you have a rare group — by chance you might get very few of them in treatment or control even under random assignment (or you might have some imbalance).

Block (or stratified) randomization II

- Blocks that are homogeneous on a given outcome increase precision of estimation for that outcome as compared with the experiment without blocks. (We will address this more in the power analysis section).
- Blocks that are homogeneous on key covariates help you avoid unlucky randomizations and/or provide statistical power for testing hypotheses about differences in effects between levels of that covariate.

Example code for block or stratified randomization

```r
# for replicability
set.seed(23456)

dat <- data.frame(block = rep(c(1, 2), each = 10))

# Complete random assignment:
dat$treatment <- block_ra(blocks = dat$block, m = 3)

with(dat, table(block, treatment, useNA = "ifany"))
```

```
##       treatment
## block 0 1
##     1 7 3
##     2 7 3
```

Cluster randomization I

- A cluster is a **group of units**. In a cluster-randomized study, all units in the cluster are assigned to the same treatment status.
- Use cluster randomization if the intervention has to work at the cluster level.
  - For example, if the intervention is about school playgrounds, then the school is the unit of assignment even if student health may be an outcome measured at level of individual students.
- Having fewer clusters hurts your ability to detect treatment effects and make cause misleading p-values and confidence intervals (or even estimates). *How much* depends on the intra-cluster correlation (ICC or ρ).

Cluster randomization II

- Higher ρ is worse:
  - When $\rho = 0$ then the village doesn't matter for the behavior of the individuals.
  - When $\rho = 1$ then every person in the village would give exactly the same answer. Having another person from this village doesn't give you additional information since his outcome is identical to the person you already had.
- For the same number of units, having **more clusters** with fewer units per cluster can help.
- Trade off spillover and power.
- If you would not like an experiment with 10 units, then you should not be happy with an experiment with 10 clusters of 100 units. The effective sample size of this cluster randomized experiment is between 10 and $10 \times 100 = 1000$, but closer to 10 the higher the ρ.

Example code for clustered randomization

```r
# for replicability
set.seed(23456)

dat <- data.frame(cluster = rep(seq(1, 10), each = 3))

# Complete random assignment:
dat$treatment <- cluster_ra(clusters = dat$cluster, m = 3)

## 3 people in each cluster, all assigned to the same treatment
with(dat, table(cluster, treatment, useNA = "ifany"))
```

```
##         treatment
## cluster 0 1
##       1 3 0
##       2 3 0
##       3 3 0
##       4 3 0
##       5 0 3
##       6 0 3
##       7 3 0
##       8 0 3
```

You can combine blocks and clusters

- You can have clusters within blocks.
  - Example: block = district, cluster = communities, units = individuals. You are measuring outcomes at the individual level.
  - Example: block = province, cluster = district, units = communities. You are measuring outcomes at the community level.
- You can't have blocks within clusters.
- For block and cluster randomization, you can use `block_ra` and `cluster_ra` in the `randomizr` package in R.
- For more complicated designs, you might find `DeclareDesign` helpful. (https://declaredesign.org)

Example: Random Access

- Randomly select a treatment group through a lottery or equivalent mechanism, which randomizes **access** to the program.
- Useful when you do not have enough resources to treat everyone.
- Sometimes, some units (peoples, communities) must have access to a program.
  - For example: a partner organization doesn't want to risk a vulnerable community NOT getting a program (want a guarantee that they will be always be treated).
  - You can exclude those units from the experiment, and do random assignment among the remaining units that have a probability of assignment strictly between (and not including) 0 and 1.

Example: Delayed access (Phase-in or wait list)

- Randomize *timing* of access to the program.
- Often you do not have the capacity to implement the treatment in a lot of places at once and/or you cannot completely exclude people from the treatment.
- When an intervention can be or must be rolled out in stages, you can randomize **the order** in which units are treated.
- Your control group are the as-yet untreated units.
- Be careful: the probability of assignment to treatment will vary over time because units that are assigned to treatment in earlier stages are not eligible to be assigned to treatment in later stages.

Factorial or crossed-assignment

- Factorial design enables testing of more than one treatment.
- You can analyze one treatment at a time.
- Or combinations thereof.
- Example:

	$X_1 = 0$	$X_1 = 1$
$X_2 = 1$	A	C
$X_2 = 0$	B	D

We might focus on an estimand like
$\mathbb{E}[Y(X_1 = 1, X_2 = 1)] - \mathbb{E}[Y(X_1 = 0, X_2 = 0)]$.

Example code for factorial randomization

See https://egap.org/resource/10-things-to-know-about-randomization/

```
dat <- data.frame(id = 1:20)

dat$treatment_1 <- complete_ra(N = 20, m = 10)
dat$treatment_2 <- block_ra(blocks = dat$treatment_1)
with(dat, table(treatment_1, treatment_2))
```

```
##             treatment_2
## treatment_1 0 1
##           0 5 5
##           1 5 5
```

Example: Encouragement

- Randomize **encouragement** to take the treatment, such as an invitation or subsidy to participate in a program.
- Useful when you cannot force a subject to participate.
- Estimands:
  - the ATE of the encouragement for your experimental sample.
  - the ATE of participation (not the encouragement) for the units who would participate when encouraged and wouldn't participate when not encouraged (compliers).
- Instrumental variables analysis for the complier ATE, with the assignment as the instrument. Note the exclusion restriction.

Example is like any of the other examples, but we measure **compliance** with or **dose** of the treatment

Best practices in randomization: replicability

- EGAP Methods Guide on Randomization (https://egap.org/resource/10-things-to-know-about-randomization/)
- Set a seed and save your code and random assignment column
- Verify
- Sometimes increased transparency $>$ replicability

Balance Checking

- Check overall balance with an omnibus d^2-test using `balanceTest` in the `RItools` package (Hansen and Bowers (2008)) (large sample randomization inference):

```
set.seed(12345)
des <- create_design(N = 30)
sim_dat <- draw_data(des)
## Simulated data (just the first 6 lines)
head(sim_dat)
```

```
##   ID id x1 x2           x3           x4          x5          x6
## 1 01  1  2  3 -2.070671332  0.331702047  0.03336694  1.3769551 -
0.2442357
## 2 02  2  2  9 -0.509982788  1.739999719  0.72424390 -1.3849193  1.
## 3 03  3  2  4  0.005877185 -0.008925433 -1.49356610  0.8191503 -
2.1476497
## 4 04  4  2  7 -1.549423415 -0.326216850  0.38888233  0.4987881 -
0.7412983
## 5 05  5  0  2  0.871203754  0.148543198  0.51151537 -0.4737256  1.
## 6 06  6  0  4  0.255750581 -0.379867894 -0.64038261  2.5937732  0.
##           x8          x9         x10         x11         x12          x
```

What to do to ensure tighter than randomized balance? Block

What if we are disturbed by the imbalance on x1 even if we know that the design overall is consistent with what we'd expect from a well randomized design? If we observe x1 before randomizing we can create strata and then randomize within those strata.

```
table(sim_dat$x1)
```

```
##
##  0  2  4  6
## 18  9  2  1
```

```
sim_dat <- sim_dat %>% mutate(x1_cat = case_when(
  x1 == 0 ~ 1,
  x1 > 0 ~ 2
))

with(sim_dat, table(x1, x1_cat))
```

```
##     x1_cat
## x1    1  2
##    0 18  0
```

The Block-randomized design has less variability in x_1

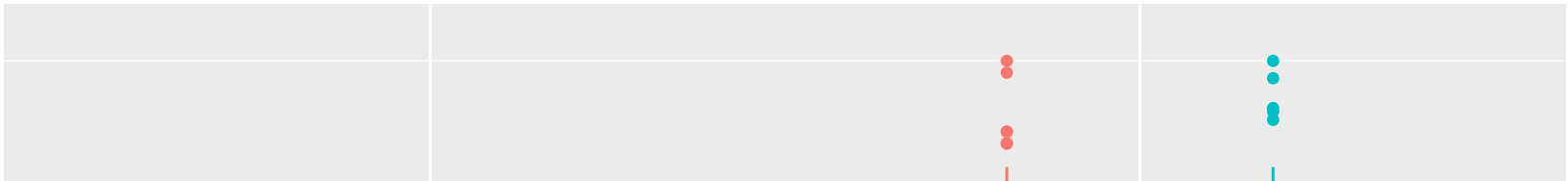
```
set.seed(12345)
des_blocked <- create_blocked_design(N = 30)
sim_dat_blocked <- draw_data(des_blocked)
with(sim_dat_blocked, table(x1_cat, Z))
```

```
##        Z
## x1_cat 0 1
##      0 9 8
##      1 7 6
```

```
simulated_des_blocked <- simulate_design(des_blocked, sims = 1000)
simulated_des <- simulate_design(des, sims = 1000)

all_sims <- bind_rows(simulated_des_blocked, simulated_des, .id = "bl

g_des <- ggplot(all_sims, aes(x = inquiry, y = estimate, groups = des
  geom_boxplot()
g_des
```

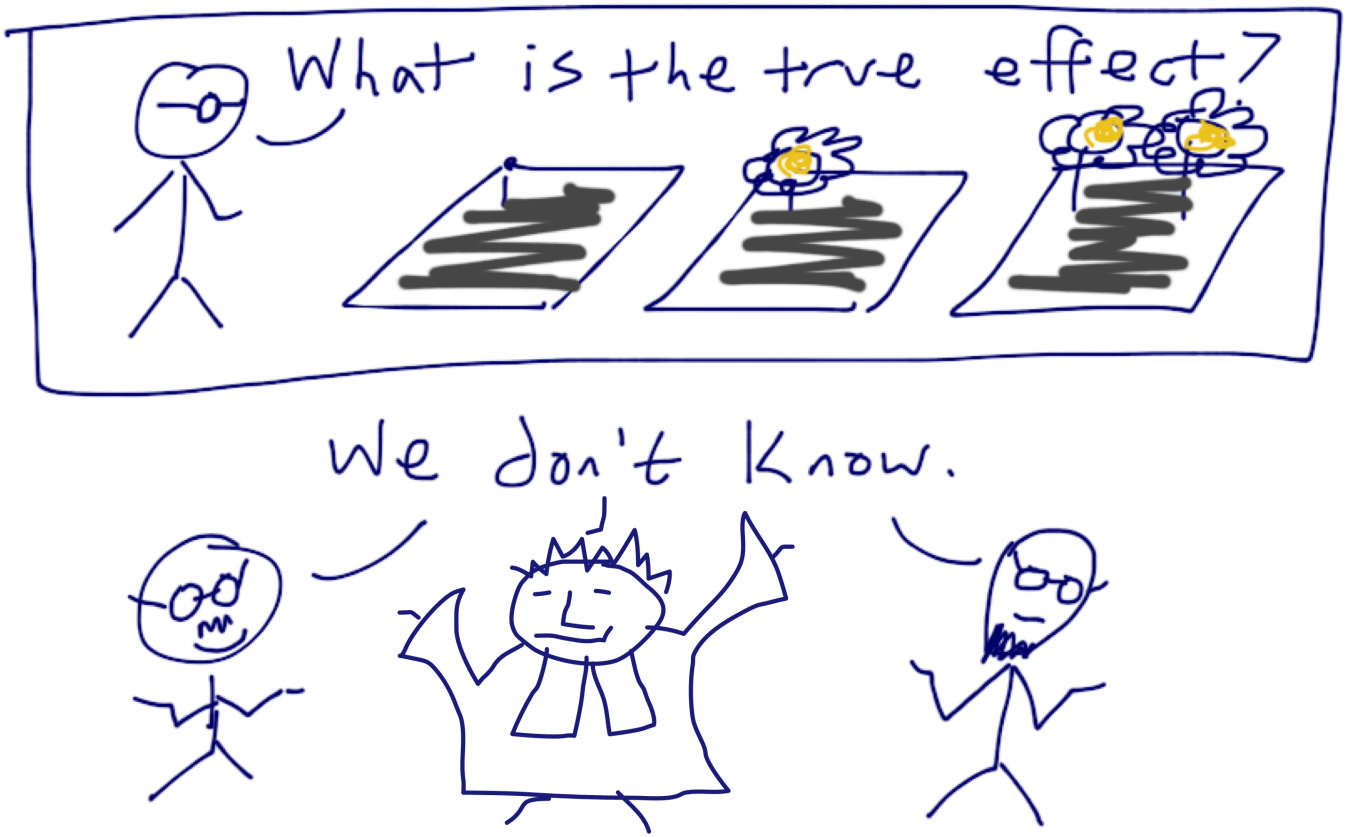


Summary about balance

- Random assignment gives us, in expectation, **overall balance** on the many covariates. It does not guarantee that all covariate to treatment relationships will be zero. In fact, in a small experiment, the magnitudes of imbalance may be high even if the randomization occurred perfectly.
- You will see t-tests of covariates one by one. Just by chance, you might get statistically significant differences on a variable. If you check balance on 100 variables, you will reject the null of no relationship in 5 of them even if there truly is no relationship. This is not a good practice. At least adjust the p-values using something like `p.adjust`.
- A small p-value from an omnibus balance test (like `balanceTest` or `independence_test`) is like a **screener** for errors in the administration of the experiment or maybe in the randomization code. It doesn't mean that the experiment was broken. The code for randomization is probably working fine, but the administration might have caused a problem. Or you might just be unlucky.
- To prevent unlucky designs trying stratifying or blocking **before** randomizing to create a block or strata-randomized design.

Testing hypotheses partially observed potential outcomes

Statistical Approaches To Causal Inference: Potential Outcomes



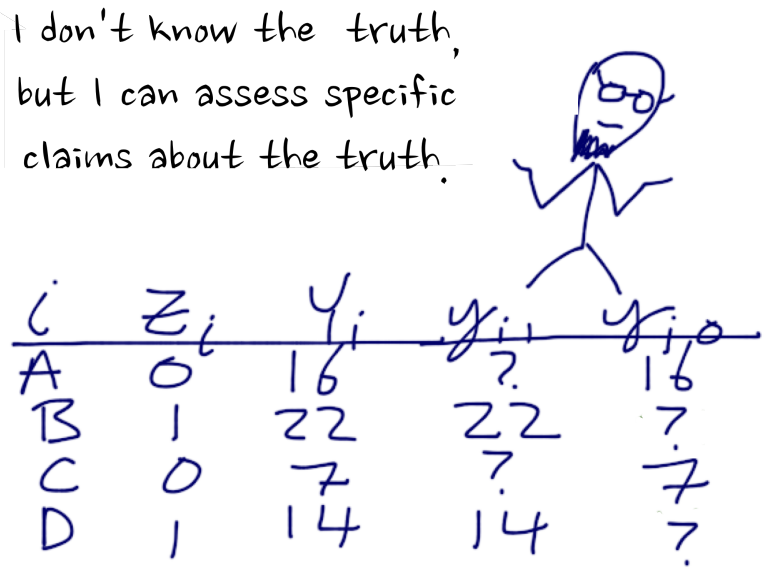
Imagine we would observe so many bushels of corn, Y, if plot i were randomly assigned to new fertilizer, $Y_i(Z_i = 1)$ (where $Z_i = 1$ means "assigned to new fertilizer" and $Z_i = 0$ means "assigned status quo fertilizer") and another amount of corn, $Y_i(Z_i = 0)$, if the same plot were assigned the status quo fertilizer condition. These Y are are *potential* or *partially observed* outcomes.

Statistical Approaches To To Causal Inference: Notation

- *Treatment* $T_i = 1$ for treatment and $T_i = 0$ for control for units i
- *One Causal Effect* for unit i is τ_i, $\tau_i = f(Y_i(1), Y_i(0))$. For example,$\tau_i = Y_i(1) - Y_i(0)$
- *Fundamental Problem of (Counterfactual) Causality* We only see one potential outcome $Y_i = T_i Y_i(1) + (1 - T_i)Y_i(0)$ manifest in our observed outcome, Y_i. Treatment reveals one potential outcome to us in a simple randomized experiment.

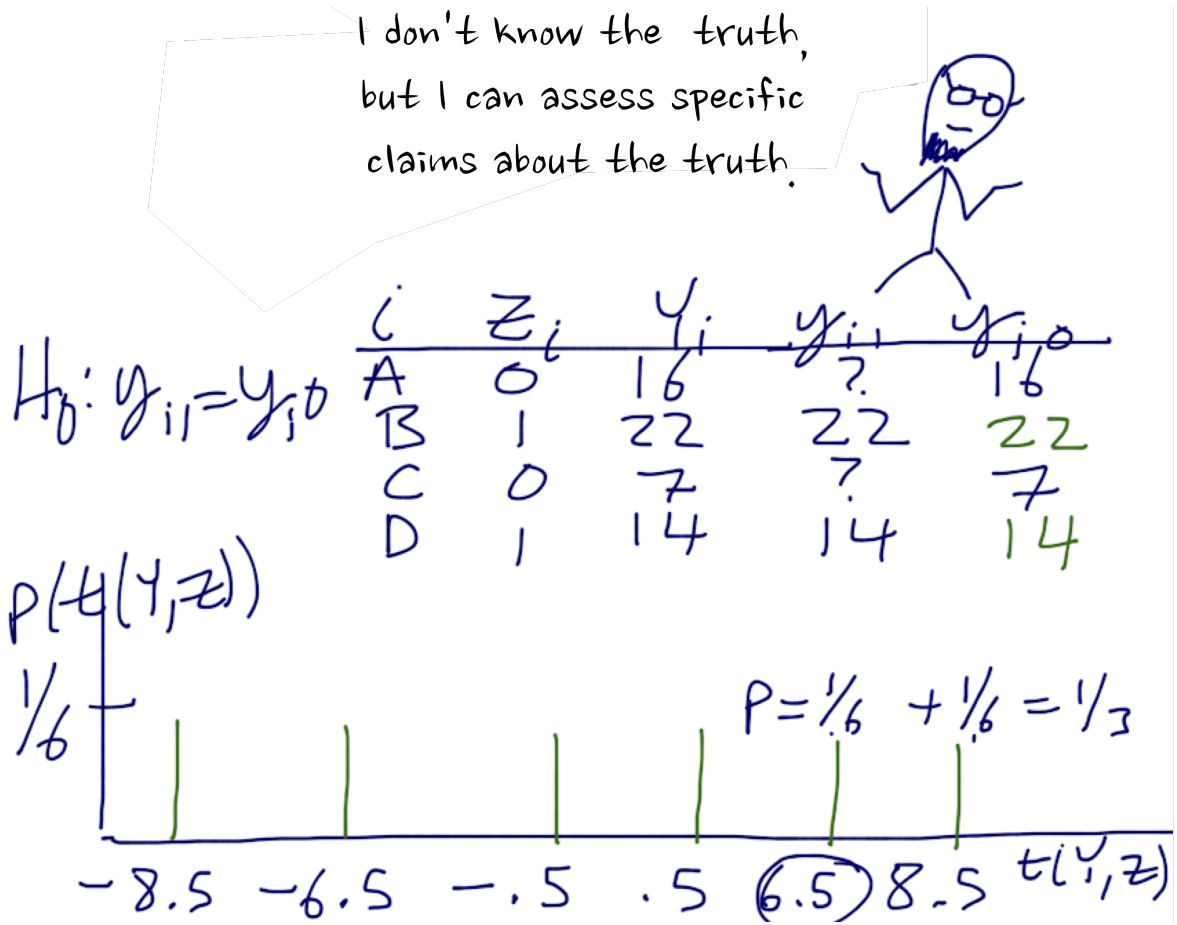
Design Based Approach 1: Compare Models of Potential Outcomes to Data

1. Posit a model of unobserved causal effects $\tau_i = f(Y_i(1), Y_i(0))$. For example $H_0 : Y_i(1) = Y_i(0) + \tau_i$ and $\tau_i = 0$ is the **sharp null hypothesis of no effects.**
2. Measure consistency of the data with this model given the research design and choice of test statistic (summarizing the treatment-to-outcome relationship).



Design Based Approach 1: Compare Models of Potential Outcomes to Data

1. Make a guess (or model of) about τ_i.
2. Measure consistency of data with this model given the design and test statistic with a p-value.



Design Based Approach 1: Compare Models of Potential Outcomes to Data

		fully observed		part observed	
	i	Z_i	Y_i	$y_{i,Z_i=1}$	$y_{i,Z_i=0}$
units	A	0	16	?	16
	B	1	22	22	?22
	C	0	7	?	7
	D	1	14	14	?14
mean diff			6.5		

We see $\bar{Y}|Z=1 = \frac{22+14}{2} = 18$

$\bar{Y}|Z=0 = \frac{16+7}{2} = 11.5$

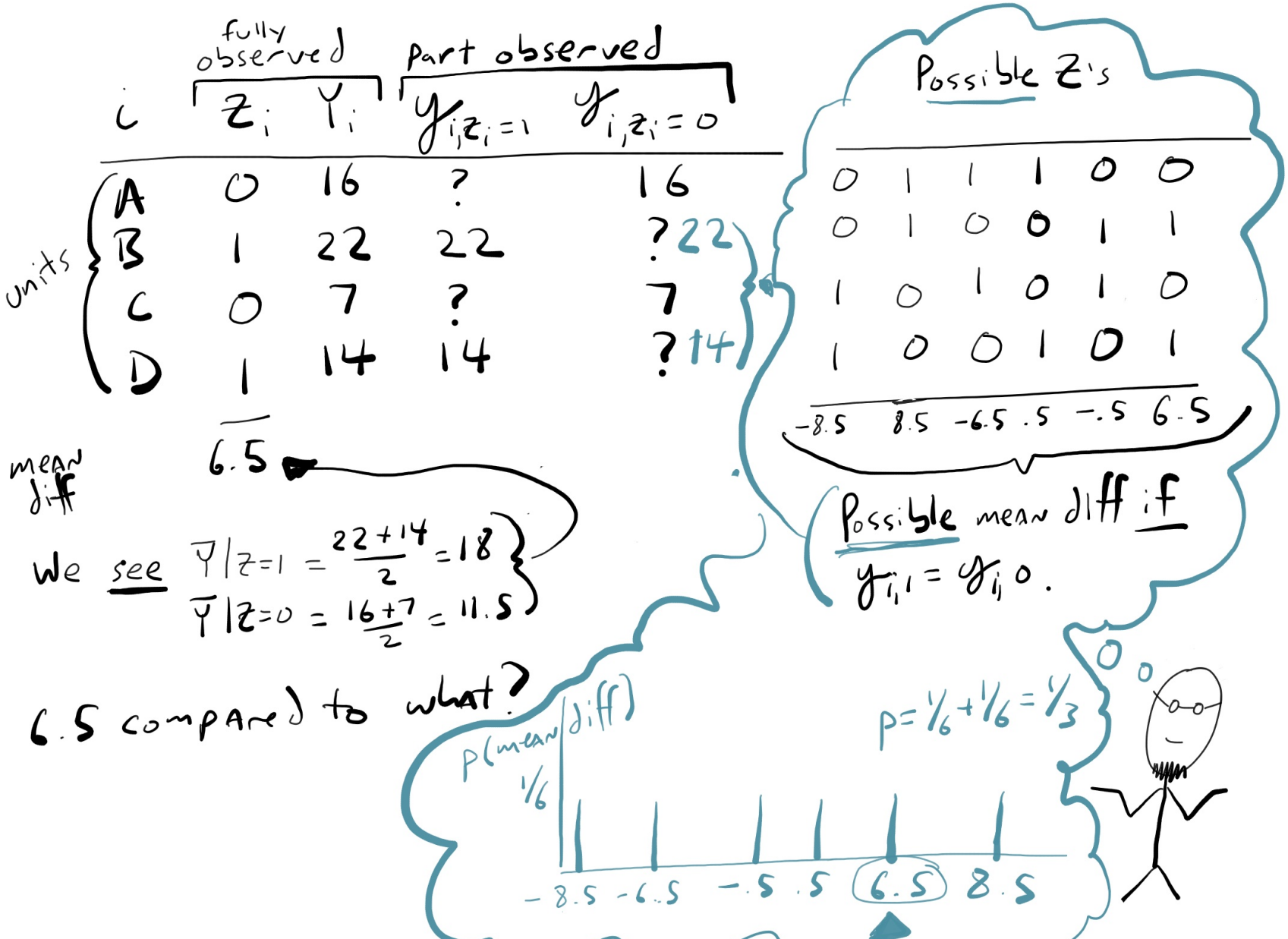
6.5 compared to what?

Possible Z's

0	1	1	1	0	0
0	1	0	0	1	1
1	0	1	0	1	0
1	0	0	1	0	1
-8.5	8.5	-6.5	.5	-.5	6.5

Possible mean diff if $y_{i,1} = y_{i,0}$.

Design Based Approach 1: Compare Models of Potential Outcomes to Data



The role of hypothesis tests in causal inference

- The hypothesis testing approach to causal inference doesn't provide a best guess but instead tells you *how much evidence or information the research design provides about a causal claim*.
- The estimation approach provides a best guess but doesn't tell you how much you know about that guess.
  - For example, a best guess with $N = 10$ seems to tell us less about the effect than $N = 1000$.
  - For example, a best guess when 95% of $Y = 1$ and 5% of $Y = 0$ seems to tell us less than when outcomes are evenly split between 0 and 1.
- We nearly always report both since both help us make decisions: "Our best guess of the treatment effect was 5, and we could reject the idea that the effect was 0 (p=.01)." (Is short hand for "I have a big enough sample or enough information (say, not a very rare outcome) that our hypothesized model of no effects would be surprising from the perspective of the observed data — using a test statistic that is a mean difference.")

Ingredients of a hypothesis test

- A **hypothesis** is a statement about a relationship among potential outcomes.
- A **test statistic** summarizes the relationship between treatment and observed outcomes.
- The **design** allows us to link the hypothesis and the test statistic: calculate a test statistic that describes a relationship between potential outcomes.
- The **design** also tells us how to generate a *distribution* of possible test statistics implied by the hypothesis.
- A p-**value** describes the relationship between our observed test statistic and the distribution of possible hypothesized test statistics.

A hypothesis is a statement about or model of a relationship between potential outcomes

Example simulated data with known individual treatment effects (ITE) and potential outcomes

Treatment	$Y_i(1)$	ITE	$Y_i(1)$	Y	$Y > 100$
0	0	10	10	0	0
1	0	30	30	30	0
0	0	200	200	0	0
0	1	90	91	1	0
1	1	10	11	11	0
1	3	20	23	23	0
1	4	30	34	34	0
1	5	40	45	45	0
0	190	90	280	190	1
0	200	20	220	200	1

For example, the sharp, or strong, null hypothesis of no effects is $H_0 : Y_i(1) = Y_i(0)$

Test statistics summarize treatment-to-outcome relationships

```r
## The mean difference test statistic
meanTT <- function(ys, z) {
  mean(ys[z == 1]) - mean(ys[z == 0])
}
## The difference of mean ranks test statistic
meanrankTT <- function(ys, z) {
  ranky <- rank(ys)
  mean(ranky[z == 1]) - mean(ranky[z == 0])
}

observedMeanTT <- meanTT(ys = Y, z = T)
observedMeanRankTT <- meanrankTT(ys = Y, z = T)
observedMeanTT
```

```
## [1] -49.6
```

```r
observedMeanRankTT
```

```
## [1] 1
```

The design links test statistic and hypothesis

The outcome we observe for each person i, Y_i, is either what we would have observed in treatment ($Y_i(1)$) **or** what we would have observed in control ($Y_i(0)$).

$$Y_i = T_i Y_i(1) + (1 - T_i) Y_i(0)$$

So, if $Y_i(1) = Y_i(0)$ then algebra tells us that $Y_i = Y_i(0)$.

What we *actually observe* is what we *would have observed in the control condition*.

The design guides creation of a **distribution** of hypothetical test statistics

We need to know how to repeat our experiment:

```
repeatExperiment <- function(N) {
  complete_ra(N)
}
```

Then we repeat it, calculating the implied test statistic by the hypothesis and design each time:

```
set.seed(123456)
possibleMeanDiffsH0 <- replicate(
  10000,
  meanTT(ys = Y, z = repeatExperiment(N = 10))
)
set.seed(123456)
possibleMeanRankDiffsH0 <- replicate(
  10000,
  meanrankTT(ys = Y, z = repeatExperiment(N = 10))
)
```

Plot the randomization distributions under the null

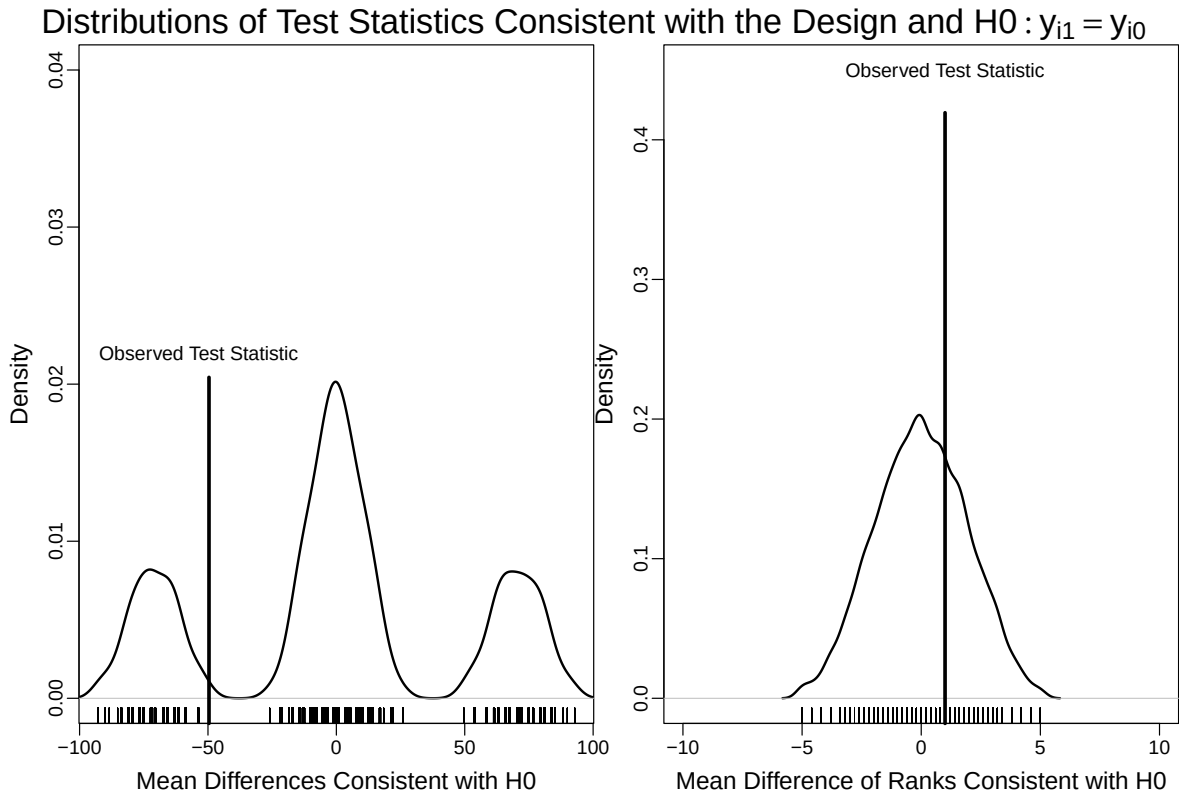


Figure 1: An example of using the design of the experiment to test a hypothesis with two different test statistics.

p-values summarize the plots

How should we interpret these p-values? (Notice that they are one-tailed.)

```
pMeanTT <- mean(possibleMeanDiffsH0 >= observedMeanTT)
pMeanRankTT <- mean(possibleMeanRankDiffsH0 >= observedMeanRankTT)
pMeanTT
```

```
## [1] 0.7785
```

```
pMeanRankTT
```

```
## [1] 0.3198
```

How to do this in R: COIN

```
## using the coin package
library(coin)
set.seed(12345)
pMean2 <- coin::pvalue(oneway_test(Y ~ factor(T),
  data = dat,
  distribution = approximate(nresample = 1000), alternative = "less"
))
dat$rankY <- rank(dat$Y)
pMeanRank2 <- coin::pvalue(oneway_test(rankY ~ factor(T),
  data = dat,
  distribution = approximate(nresample = 1000), alternative = "less"
))
pMean2
```

```
## [1] 0.783
## 99 percent confidence interval:
##  0.7476049 0.8156543
```

```
pMeanRank2
```

```
## [1] 0.323
## 99 percent confidence interval:
```

Next topics

- Testing weak null hypotheses, $H_0 : \bar{y}_1 = \bar{y}_0$.
- Rejecting null hypotheses (and making false positive and/or false negative errors).
- Maintaining correct false positive error rates when testing more than one hypothesis.
- Power of hypothesis tests (Module on Statistical Power and Design Diagnosands tomorrow).

Testing weak hypotheses about aggregates of potential outcomes like the ATE

Testing the weak null of no average effects

- The weak null hypothesis is a claim about aggregates, and it is nearly always stated in terms of averages: $H_0 : \bar{y}_1 = \bar{y}_0$
- The test statistic for this hypothesis is nearly always the simple difference of means (i.e., `meanTT()` above).

```
lm1 <- lm(Y ~ T, data = dat)
lm1P <- summary(lm1)$coef["T", "Pr(>|t|)"]
ttestP1 <- t.test(Y ~ T, data = dat)$p.value
library(estimatr)
ttestP2 <- difference_in_means(Y ~ T, data = dat)
c(lm1P = lm1P, ttestP1 = ttestP1, tttestP2 = ttestP2$p.value)
```

```
##       lm1P    ttestP1 tttestP2.T
##  0.3320959  0.3587401  0.3587401
```

- Why is the OLS p-value different? What assumptions do we use to calculate it? How do those assumptions relate to the experimental design?

Testing the weak null of no average effects

Both variation and location of Y changes with treatment in this simulation, plus we have some extreme points — like two groups that become much more similar after the intervention.

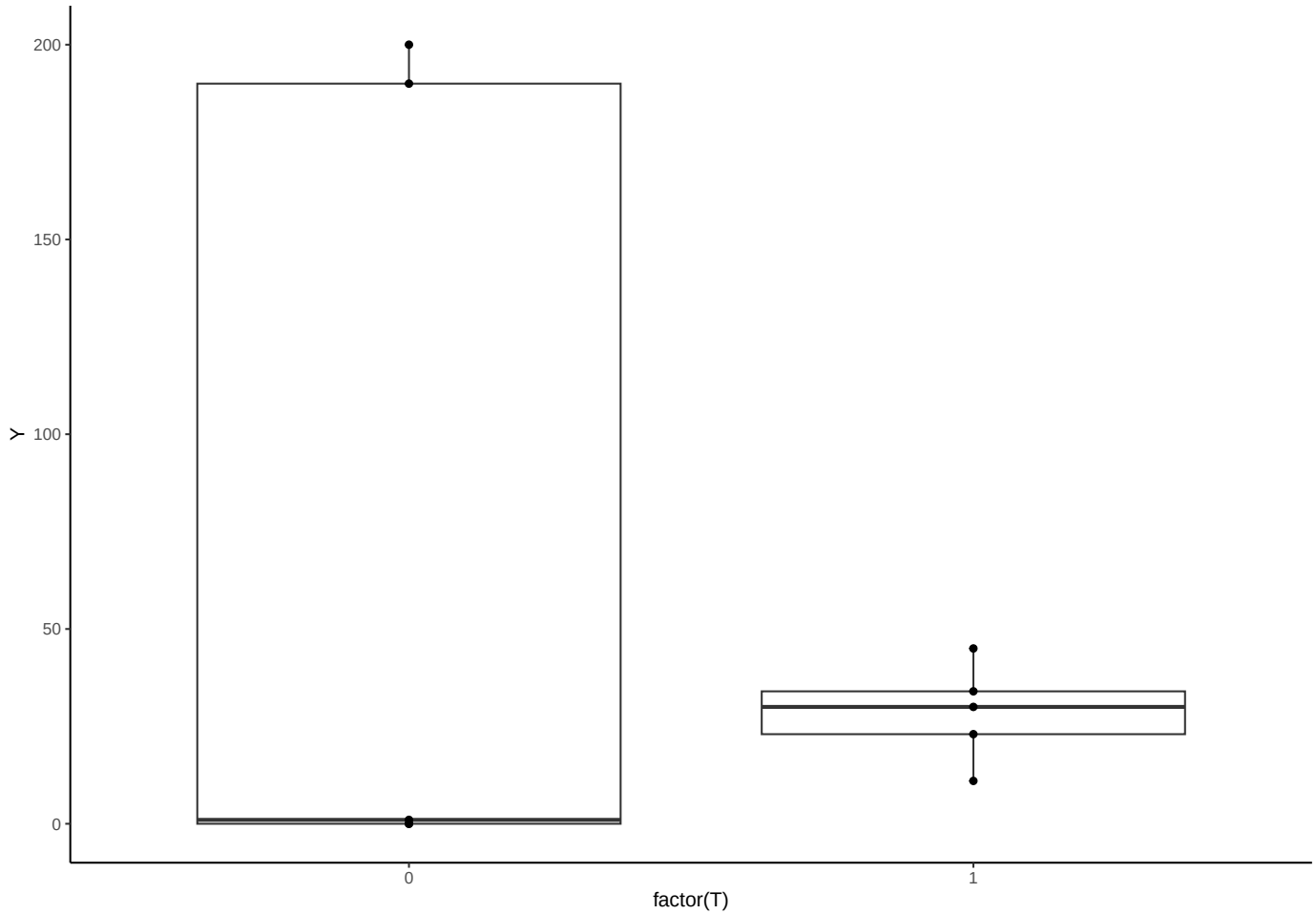


Figure 2: Boxplot of observed outcomes by treatment status

Testing the weak null of no average effects

Reporting a two-sided p-value here:

```
## By hand:
varEstATE <- function(Y, T) {
  var(Y[T == 1]) / sum(T) + var(Y[T == 0]) / sum(1 - T)
}
seEstATE <- sqrt(varEstATE(dat$Y, dat$T))
obsTStat <- observedMeanTT / seEstATE
c(
  observedTestStat = observedMeanTT,
  stderror = seEstATE,
  tstat = obsTStat,
  pval = 2 * min(
    pt(obsTStat, df = 8, lower.tail = TRUE),
    pt(obsTStat, df = 8, lower.tail = FALSE)
  )
)
```

```
## observedTestStat          stderror             tstat              pva
##      -49.6000000        48.0447708        -1.0323704         0.33209
```

Rejecting hypotheses and making errors

- "In typical use, the level of the test [α] is a promise about the test's performance and the size is a fact about its performance..." (Rosenbaum 2010, Glossary)
- α is the probability of rejecting the null hypothesis when the null hypothesis is true.
- How should we interpret p=0.78? What about p=0.32 (our tests of the sharp null)?
- What does it mean to "reject" $H_0 : Y_i(1) = Y_i(0)$ at $\alpha = .05$?

False positive rates in hypothesis testing I

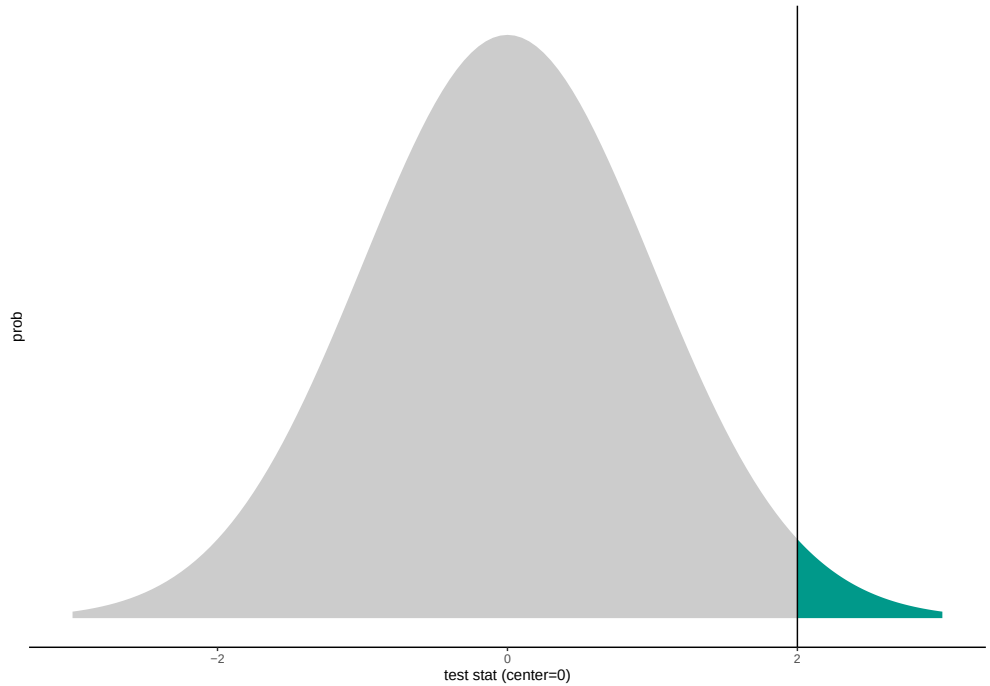


Figure 3: One-sided p-value from a Normally distributed test statistic.

Notice:

- The curve is centered at the hypothesized value.
- The curve represents the world of the hypothesis.

False positive rates in hypothesis testing II

- The p-value is how rare it would be to see the observed test statistic (or a value farther away from the hypothesized value) in the world of the null.
- In the picture, the observed value of the test statistic is consistent with the hypothesized distribution, but just not super consistent.
- Even if $p < .05$ (or $p < .001$) the observed test statistic must reflect some value on the hypothesized distribution. **This means that you can always make an error when you reject a null hypothesis.**

False positive and false negative errors

- If we say, "The experimental result is significantly different from the hypothesized value of zero ($p = .001$)! We reject that hypothesis!" **when the truth is zero** we are making a **false positive error** (claiming to detect something positively when there is no signal, only noise).
- If we say, "We cannot distinguish this result from zero ($p = .3$). We cannot reject the hypothesis of zero." **when the truth is not zero** we are making a **false negative error** (claiming inability to detect something when there is a signal, but it is overwhelmed by noise.)

A single test of a single hypothesis

- A single test of a single hypothesis should encourage false positive errors rarely. For example, if we set $\alpha = .05$, then we are saying that we are comfortable with our testing procedure making false positive errors in **no more than 5% of tests of a given treatment assignment in a given experiment**.
- Also, a **single test of a single hypothesis** should detect a signal when it exists — it should be have high **statistical power**. In other words, it should not fail to detect a signal when it exists (i.e. should have low false negative error rates).

Decisions imply errors

You don't have to reject or not-reject. But it is difficult to *act* without deciding.

- If errors are necessary, how can we diagnose them? How do we learn whether our hypothesis-testing procedure might generate too many false positive errors?
- Diagnose by simulation!

Diagnosing false positive rates by simulation

- Across repetitions of the design:
  - Create a true null hypothesis.
  - Test the true null.
  - The p-value should be large if the test is operating correctly.
- The proportion of small p-values should be no larger than α if the test is operating correctly.

Diagnosing false positive rates by simulation

Example with a binary outcome. Does the test work as it should? What do the p-values look like when there is no effect?

```
collectPValues <- function(y, z, thedistribution = exact()) {
  ## Make Y and T have no relationship by re-randomizing T
  newz <- repeatExperiment(length(y))
  ## The four tests
  thelm <- lm(y ~ newz, data = dat)
  ttestP2 <- difference_in_means(y ~ newz, data = dat)
  owP <- coin::pvalue(oneway_test(y ~ factor(newz),
    distribution = thedistribution
  ))
  ranky <- rank(y)
  owRankP <- coin::pvalue(oneway_test(ranky ~ factor(newz),
    distribution = thedistribution
  ))
  ## Return the p-values
  return(c(
    lmp = summary(thelm)$coef["newz", "Pr(>|t|)"],
    neyp = ttestP2$p.value[[1]],
    rtp = owP,
```

Diagnosing false positive rates by simulation

- After setting the simulation to have no effect, a test of the null hypothesis of no effects should produce a **big** p-value.
- If the test is working well, we should see mostly big p-values and very few small p-values.
- A few of the p-values for the four different tests (we did 5000 simulations, just showing 5)

```
##                [,1]      [,2] [,3]      [,4] [,5]
## lmp       0.1411133 0.1411133    1 0.1411133    1
## neyp      0.1778078 0.1778078    1 0.1778078    1
## rtp       0.4444444 0.4444444    1 0.4444444    1
## rtpRank   0.4444444 0.4444444    1 0.4444444    1
```

Diagnosing false positive rates by simulation

In fact, if there is no effect, and if we decided to reject the null hypothesis of no effects with $\alpha = .25$, we would want **no more than 25% of our p-values in this simulation to be less than p=.25**. What do we see here? Which tests appear to have false positive rates that are too high?

```
## Calculate the proportion of p-values less than .25 for each row of
apply(pDist, 1, function(x) {
  mean(x < .25)
})
```

```
##     lmp    neyp     rtp rtpRank
##   0.445   0.445   0.000   0.000
```

Diagnosing false positive rates by simulation

Compare tests by plotting the proportion of p-values less than any given number. The "randomization inference" tests control the false positive rate (these are the tests of using direct permutation, repeating the experiment).

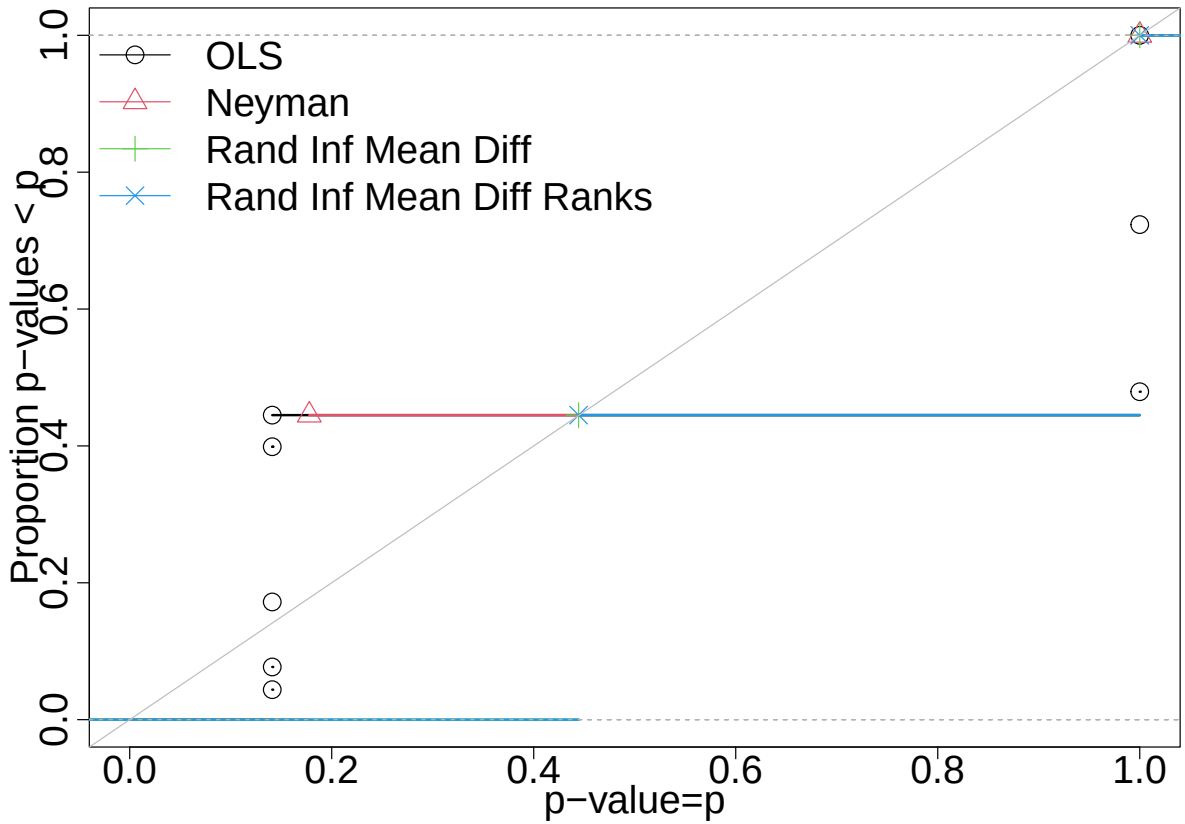


Figure 4: P-value distributions when there are no effects for four tests with n=10. A test that controls its false positive rate should have points on or below the diagonal line.

False positive rate with $N = 60$ and binary outcome

In this design only the direct randomization inference-based tests control the false positive rate.

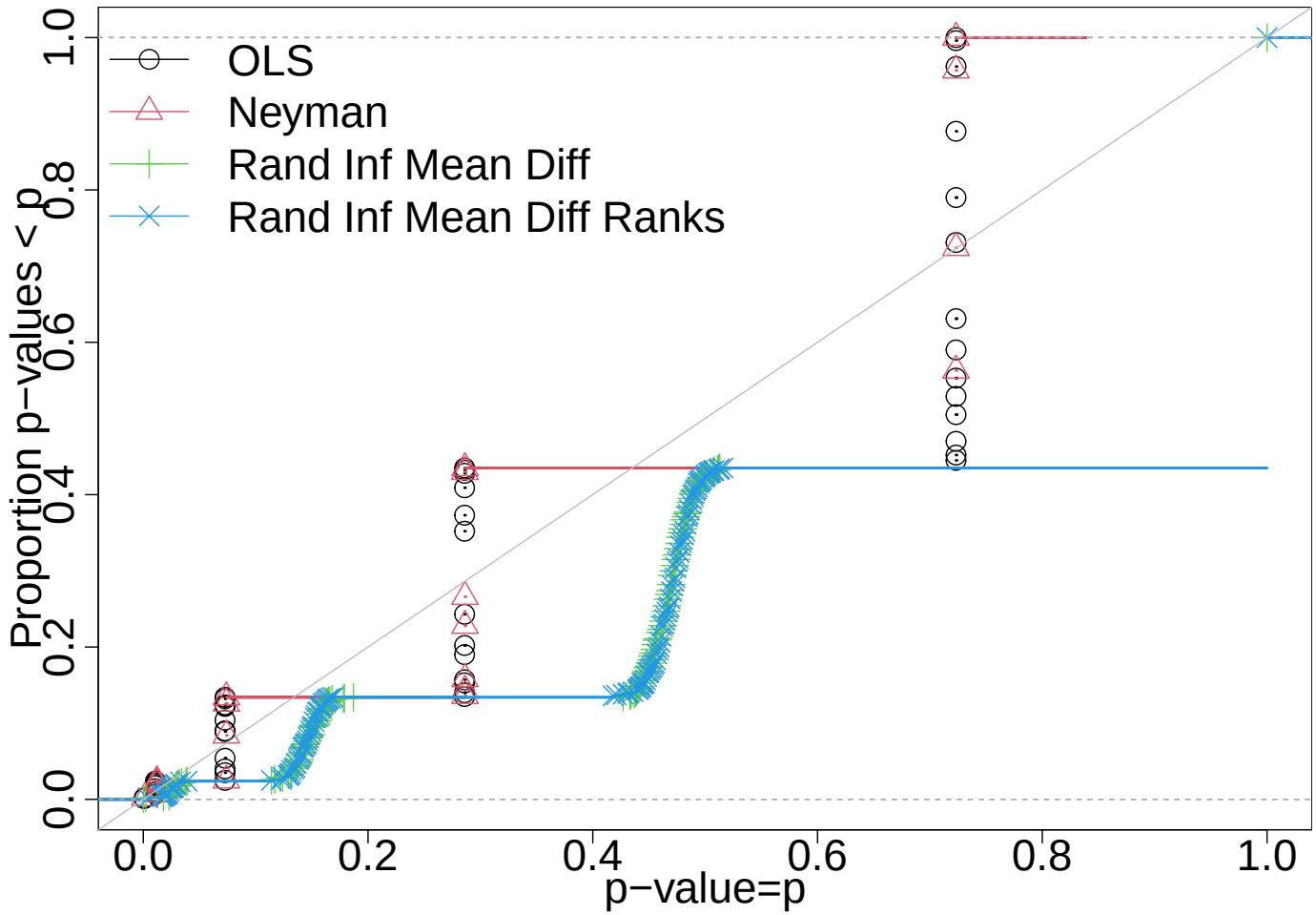


Figure 5: P-value distributions when there are no effects for four tests with n=60 and a binary outcome. A test that controls its false positive rate should have points on or below the diagonal line.

False positive rate with $N = 60$ and continuous outcome

Here, all of the tests do a good job of controlling the false positive rate.

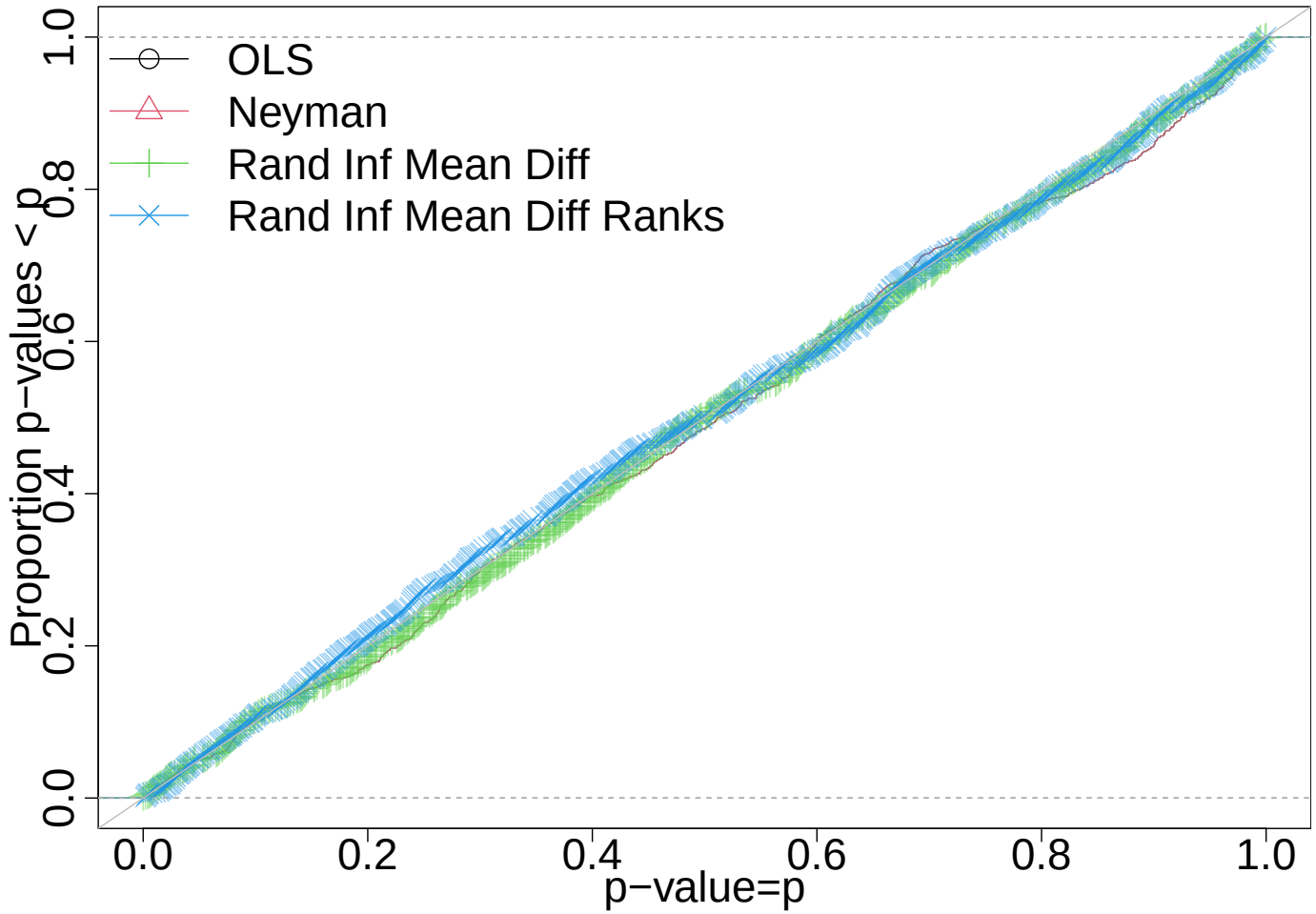


Figure 6: P-value distributions when there are no effects for four tests with n=60 and a continuous outcome. A test that controls its false positive rate should have points on or below the diagonal line.

Summary

- A good test:
  1. casts doubt on the truth rarely, and
  2. easily distinguishes signal from noise (casts doubt on falsehoods often).
- We can learn whether our testing procedure controls false positive rates given our design using simulation.
- When false positive rates are not controlled, what might be going wrong? (Often has to do with asymptotics.)

Advanced Topics in Hypothesis Testing

Some advanced topics connected to hypothesis testing

- Even if a given testing procedure controls the false positive rate for a single test, it may not control the rate for a group of multiple tests. See 10 Things you need to know about multiple comparisons for a guide to the approaches to controlling such rejection-rates in multiple tests.
- A $100(1-\alpha)\%$ confidence interval can be defined as the range of hypotheses where all of the p-values are greater than or equal to α. This is called inverting the hypothesis test (P R Rosenbaum (2010)). That is, a confidence interval is a collection of hypothesis tests.

What else to know about hypothesis tests I

- A point estimate based on hypothesis testing is called a Hodges-Lehmann point estimate (Paul R. Rosenbaum (1993),Hodges and Lehmann (1963)).
- A set of hypothesis tests can be combined into one single hypothesis test (Hansen and Bowers (2008),Caughey, Dafoe, and Seawright (2017)).
- In equivalence testing, one can hypothesize that two test-statistics are equivalent (i.e., the treatment group is the same as the control group) rather than only about one test-statistic (the difference between the two groups is zero) (Hartman and Hidalgo (2018)).
- Since a hypothesis test is a model of potential outcomes, one can use hypothesis testing to learn about complex models, such as models of spillover and propagation of treatment effects across networks (Bowers, M. M. Fredrickson, and Panagopoulos (2013), Bowers, M. Fredrickson, and Aronow (2016), Bowers, Desmarais, et al. (2018))

Exercise: Hypothesis Tests and Test Statistics

1. If an intervention was very effective at increasing the variability of an outcome but did not change the mean, would the p-value reported by R or Stata if we used `lm_robust()` or `difference_of_means()` or `reg` or `t.test` be large or small?
2. If an intervention caused the mean in the control group to be moderately reduced but increased a few outcomes a lot (like a 10 times effect), would the p-value from R `lm_robust()` or `difference_of_means()` be large or small?

Testing many hypotheses

When might we test many hypotheses?

- Does the effect of an experimental treatment differ between different groups? Could differences in treatment effect arise because of some background characteristics of experimental subjects?
- Which, among several, strategies for communication were most effective on a single outcome?
- Which, among several outcomes, were influenced by a single experimental intervention?

False positive rates in multiple hypothesis testing

Say our probability of making a false positive error is .05 in a single test. What happens if we ask: (1) *which of these 10 outcomes has a statistically significant relationship with the two arms of treatment*? or (2) *which of these 10 treatment arms had a statistically significant relationship with the single outcome*?

- Prob of false positive error should be less than or equal to .05 in 1 test.
- Prob of one false positive error should be less than or equal to $1 - ((1 - .05) \times (1 - .05)) = .0975$ in 2 tests.
- Prob of at least one false positive error with $\alpha = .05$ in 10 tests should be $\leq 1 - (1 - .05)^{10} = .40$.

Discoveries with multiple tests

Number of errors committed when testing m null hypotheses (Benjamini and Hochberg, 1995, 's Table 1). Cells are numbers of tests. R is # of "discoveries" and V is # of false discoveries, U is # of correct non-rejections, and S is # of correct rejections.

	Declared Non-Significant	Declared Significant	Total
True null hypotheses ($H_{true} = 0$)	U	V	m_0
Not true null hyps ($H_{true} \neq 0$)	T	S	m - m_0
Total	m-R	R	m

Two main error rates to control when testing many hypotheses I

1. **Family wise error rate (FWER)** is $P(V > 0)$ (Probability of any false positive error).
   - We'd like to control this if we plan to make a decision about the results of our multiple tests. The research project is mostly confirmatory.
   - See, for example, the projects of the OES http://oes.gsa.gov: federal agencies will make decisions about programs depending on whether they detect results or not.
2. **False Discovery Rate (FDR)** is $E(V/R|R > 0)$ (Average proportion of false positive errors given some rejections).
   - We'd like to control this if we are using *this* experiment to plan *the next* experiment. We are willing to accept a higher probability of error in the interests of giving us more possibilities for discovery.
   - For example, one could imagine an organization, a government, an NGO, could decide to conduct *a series* of experiments as a part of a *learning agenda*: no single experiment determines decision making, more room for exploration.

Two main error rates to control when testing many hypotheses II

We will focus on FWER but recommend thinking about FDR and learning agendas as a very useful way to go.

Questions with multiple outcomes

- What is the effect of one treatment on multiple outcomes?
- On which outcomes (out of many) did the treatment have an effect?
- The second question, in particular, can lead to the kind of uncontrolled family wise error rate problems that we referred to above.

Multiple hypothesis testing: Multiple Outcomes

Imagine we had five outcomes and one treatment (showing potential and observed outcomes here):

```
##    ID T Y1_T_1 Y2_T_0 Y2_T_1 Y3_T_0 Y3_T_1 Y4_T_0 Y4_T_1 Y5_T_0
## 1 001 0   0.19  0.366  0.366  0.546  0.546 -0.626 -0.626 -0.125
## 2 002 0  -0.43  0.931  0.931 -2.233 -2.233  1.309  1.309  1.078
## 3 003 0   0.91 -1.907 -1.907  0.288  0.288 -0.133 -0.133 -1.261
## 4 004 0   1.79  0.052  0.052  0.544  0.544 -1.608 -1.608 -0.452
## 5 005 1   1.00 -0.848 -0.848 -1.192 -1.192 -1.308 -1.308 -1.027
## 6 006 0   1.11 -0.368 -0.368 -0.018 -0.018 -0.045 -0.045  0.068
```

```
##    ID Y5_T_1    Y1     Y2     Y3     Y4     Y5
## 1 001 -0.125  0.19  0.366  0.546 -0.626 -0.125
## 2 002  1.078 -0.43  0.931 -2.233  1.309  1.078
## 3 003 -1.261  0.91 -1.907  0.288 -0.133 -1.261
## 4 004 -0.452  1.79  0.052  0.544 -1.608 -0.452
## 5 005 -1.027  1.00 -0.848 -1.192 -1.308 -1.027
## 6 006  0.068  1.11 -0.368 -0.018 -0.045  0.068
```

Can we detect an effect on outcome Y1?

Can we detect an effect on outcome Y1? (i.e., does the hypothesis test produce a small enough p-value?)

```
coin::pvalue(oneway_test(Y1 ~ factor(T), data = dat1))
```

```
## [1] 0.88
```

```
## Notice that the t-test p-value is also a chi-squared test
## p-value.
coin::pvalue(independence_test(Y1 ~ factor(T),
  data = dat1,
  teststat = "quadratic"
))
```

```
## [1] 0.88
```

On which of the five outcomes can we detect an effect?

On which of the five outcomes can we detect an effect? (i.e., does any of the five hypothesis tests produce a small enough p-value?)

```
p1 <- coin::pvalue(oneway_test(Y1 ~ factor(T), data = dat1))
p2 <- coin::pvalue(oneway_test(Y2 ~ factor(T), data = dat1))
p3 <- coin::pvalue(oneway_test(Y3 ~ factor(T), data = dat1))
p4 <- coin::pvalue(oneway_test(Y4 ~ factor(T), data = dat1))
p5 <- coin::pvalue(oneway_test(Y5 ~ factor(T), data = dat1))
theps <- c(p1 = p1, p2 = p2, p3 = p3, p4 = p4, p5 = p5)
sort(theps)
```

```
##   p5   p4   p3   p2   p1
## 0.27 0.30 0.43 0.59 0.88
```

Can we detect an effect for *any* of the five outcomes?

Can we detect an effect for *any* of the five outcomes? (i.e., does the hypothesis test for *all* five outcomes at once produce a small enough p-value?)

```
coin::pvalue(independence_test(Y1 + Y2 + Y3 + Y4 + Y5 ~ factor(T),
  data = dat1, teststat = "quadratic"
))
```

```
## [1] 0.67
```

Which approach is likely to mislead us with too many "statistically significant" results (5 tests or 1 omnibus test)?

Comparing approaches I

Let's do a simulation to learn about these testing approaches.

- We will (1) set the true causal effects to be 0, (2) repeatedly re-assign treatment, and (3) each time, do each of those three tests.
- Since the true effect is 0, we expect *most* of the p-values to be large. (In fact, we'd like no more than 5% of the p-values to be greater than $p = .05$ if we are using the $\alpha = .05$ accept-reject criterion).

```
des1_sim <- simulate_design(des1_plus, sims = 1000)
res1 <- des1_sim %>%
  group_by(estimator) %>%
  summarize(fwer = mean(p.value < .05), .groups = "drop")
```

Comparing approaches II

Table 3: Family wise error rates

estimator	fwer
t-test Y1	0.05
t-test all	0.24
t-test all holm adj	0.05
t-test omnibus	0.04

- The approach using 5 tests produces a $p < .05$ much too often — recall that there are no causal effects at all for any of these outcomes.
- A test of a single outcome (here `Y1`) has $p < .05$ no more than 5% of the simulations.
- The omnibus test also shows a well-controlled error rate.
- Using a multiple testing correction (here we use the "Holm" correction) also correctly controls the false positive rate.

The Holm correction

FYI, here is how to use the Holm correction (Notice what happens to the p-values):

```
theps
```

```
##   p1   p2   p3   p4   p5
## 0.88 0.59 0.43 0.30 0.27
```

```
p.adjust(theps, method = "holm")
```

```
## p1 p2 p3 p4 p5
##  1  1  1  1  1
```

```
## To show what happens with "significant" p-values
theps_new <- sort(c(theps, newlowp = .01))
p.adjust(theps_new, method = "holm")
```

```
## newlowp      p5      p4      p3      p2      p1
##    0.06    1.00    1.00    1.00    1.00    1.00
```

Multiple hypothesis testing: Multiple treatment arms I

- The same kind of problem can happen when the question is about the differential effects of a multi-armed treatment.
- With 5 arms, "the effect of arm 1" could mean many different things: "Is the average potential outcome under arm 1 bigger than arm 2?", "Are the potential outcomes of arm 1 bigger than the average potential outcomes of all of the other arms?"
- If we just focus on pairwise comparisons across arms, we could have $((5 \times 5) - 5)/2 = 10$ unique tests!

Multiple hypothesis testing: Multiple treatment arms I

Here are some potential and observed outcomes and T with multiple values.

```
##    ID T  Y_T_2  Y_T_3  Y_T_4  Y_T_5      Y
## 1 001 3  0.366  0.546 -0.626 -0.125  0.546
## 2 002 3  0.931 -2.233  1.309  1.078 -2.233
## 3 003 4 -1.907  0.288 -0.133 -1.261 -0.133
## 4 004 5  0.052  0.544 -1.608 -0.452 -0.452
## 5 005 2 -0.848 -1.192 -1.308 -1.027 -0.848
## 6 006 3 -0.368 -0.018 -0.045  0.068 -0.018
```

Multiple hypothesis testing: Multiple treatment arms I

Here are the 10 pairwise tests with and without adjustment for multiple testing. Notice how one "significant" result ($p = .01$) changes with adjustment.

```
##    Comparison      Stat p.value p.adjust
## 1   1 - 2 = 0     1.435   0.231   1.0000
## 2   1 - 3 = 0    0.8931  0.3447   1.0000
## 3   1 - 4 = 0     6.404 0.01139   0.1139
## 4   1 - 5 = 0    0.8216  0.3647   1.0000
## 5   2 - 3 = 0   0.05882  0.8084   1.0000
## 6   2 - 4 = 0     2.641  0.1041   0.7287
## 7   2 - 5 = 0    0.0437  0.8344   1.0000
## 8   3 - 4 = 0     3.232 0.07222   0.6500
## 9   3 - 5 = 0 0.0003464  0.9852   1.0000
## 10  4 - 5 = 0     2.899 0.08861   0.7089
```

Approaches to testing hypotheses with multiple arms

We illustrate four different approaches:

1. do all of the pairwise tests and choose the best one (a bad idea);
2. do all the pairwise tests and choose the best one after adjusting the p-values for multiple testing (a fine idea but one with very low statistical power);
3. test the hypothesis of no relationship between *any arm* (an omnibus test) and the outcome (a fine idea);
4. choose one arm to focus on in advance (a fine idea).

Table 4: Approaches to testing in multi-arm experiments.

estimator	fwer
Choose best pairwise test	0.234
Choose best pairwise test after adjustment	0.015
Overall test	0.030
t-test T1 vs all	0.016

Summary

- Multiple testing problems can arise from multiple outcomes or multiple treatments (or multiple moderators/interaction terms).
- Procedures for making hypothesis tests and confidence intervals can involve error. Ordinary practice controls the error rates in a single test (or single confidence interval). But multiple tests require extra work to ensure that error rates are controlled.
- The loss of power arising from adjustment approaches encourages us to consider what *questions we want to ask of the data*. For example, if we want to know if the treatment had *any effect*, then a joint test or omnibus test of multiple outcomes will increase our statistical power without requiring adjustment.

Estimation

Based Approach 2: Estimate Averages of Potential Outcomes

1. Notice that the observed Y_i are a sample from the (small, finite) population of unobserved potential outcomes $(y_{i,1}, y_{i,0})$.
2. Decide to focus on the average, $\bar{\tau}$, because sample averages, $\hat{\bar{\tau}}$ are unbiased and consistent estimators of population averages.
3. Estimate $\bar{\tau}$ with the observed difference in means as $\hat{\bar{\tau}}$.

I don't know the truth, but I can provide a good guess of the average causal effect.

i	Z_i	Y_i	$y_{i,1}$	$y_{i,0}$
A	0	16	?	16
B	1	22	22	?
C	0	7	?	7
D	1	14	14	?
			$\bar{y}_{i1}$	$\bar{y}_{i0}$

$$\widehat{ATE} = \bar{Y}_i|Z_i=1 - \bar{Y}_i|Z_i=0$$

$$= \frac{22+14}{} \quad \frac{16+7}{}$$

Design Based Approach 2: Estimate Averages of Potential Outcomes

I don't know the truth, but I can provide a good guess of the average causal effect.

i	Z_i	Y_i	y_{i1}	y_{i0}
A	0	16	?	16
B	1	22	22	?
C	0	7	?	7
D	1	14	14	?
			$\bar{y}_{i1}$	$\bar{y}_{i0}$

$$ATE = \bar{Y}_i|Z_i=1 - \bar{Y}_i|Z_i=0$$

$$= \frac{22+14}{2} - \frac{16+7}{2} = 6.5$$

Design Based Approach 2: Estimate Averages of Potential Outcomes

Here using Neyman's standard errors (same as HC2 SEs) and Central Limit Theorem based p-values and 95% confidence intervals:

```
est1 <- difference_in_means(Y ~ T, data = dat)
est1
```

```
## Design:  Standard
##   Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper    DF
## T    -49.6      48.04  -1.032   0.3587   -181.6    82.36 4.113
```

Key points about estimation I

- A causal effect, τ_i, is a comparison of unobserved potential outcomes for each unit i: examples $\tau_i = Y_i(T_i = 1) - Y_i(T_i = 0)$ or $\tau_i = \frac{Y_i(T_i=1)}{Y_i(T_i=0)}$.
- To learn about τ_i, we can treat τ_i as an **estimand** or target quantity to be estimated (discussed here) or as a target quantity to be hypothesized about (session on hypothesis testing).
- Many focus on the **average treatment effect (ATE)**, $\bar{\tau} = \sum_{i=1}^{n} \tau_i$, in part, because it allows for easy **estimation**.

Key points about estimation II

- They key to estimation for causal inference is to choose an estimand that helps you learn about your theoretical or policy question. So, one could use the ATE but other common estimands include the ITT, LATE/CACE, ATT, or ATE for some subgroup (or even a different of causal effects between groups).
- An **estimator** is a recipe for calculating a guess about the value of an estimand. For example, the difference of observed means for m treated units is one estimator of $\bar{\tau}$: $\hat{\bar{\tau}} = \frac{\sum_{i=1}^{n}(T_i Y_i)}{m} - \frac{\sum_{i=1}^{n}((1-T_i)Y_i)}{(n-m)}$.

Key points about estimation III

- The **standard error** of an estimator in a randomized experiment summarizes how the estimates would vary if the experiment were repeated.
- We use the **standard error** to produce **confidence intervals** and **p-values**: so that we can begin with an estimator and end at a hypothesis test.
- Different randomizations will produce different values of the same estimator targeting the same estimand. A **standard error** summarizes this variability in an estimator.
- A $100(1-\alpha)\%$ **confidence interval** is a collection of hypotheses that cannot be rejected at the α level. We tend to report confidence intervals containing hypotheses about values of our estimand and use our estimator as a test statistic.

Key points about estimation IV

- Estimators should:
  - avoid systematic error in their guessing of the estimand (be unbiased);
  - vary little in their guesses from experiment to experiment (be precise or efficient); and
  - perhaps ideally converge to the estimand as they use more and more information (be consistent).

Key points about estimation V

- **Analyze as you randomize** in the context of estimation means that (1) our standard errors should measure variability from randomization and (2) our estimators should target estimands defined in terms of potential outcomes.
- We do not **control for** background covariates when we analyze data from randomized experiments. But covariates can make our estimation more **precise**. This is called **covariance adjustment** (or covariate adjustment). **Covariance adjustment** in randomized experiments differs from controlling for in observational studies.

Review: Causal effects

Review: Causal inference refers to a comparison of unobserved, fixed, potential outcomes.

For example:

- the potential, or possible, outcome for unit i when assigned to treatment, $T_i = 1$ is $Y_i(T_i = 1)$.
- the potential, or possible, outcome for unit i when assigned to control, $T_i = 0$ is $Y_i(T_i = 0)$.

Treatment assignment, T_i, has a causal effect on unit i, that we call τ_i, if $Y_i(T_i = 1) - Y_i(T_i = 0) \neq 0$ or $Y_i(T_i = 1) \neq Y_i(T_i = 0)$.

How can we learn about causal effects from observed data?

1. Recall: we can **test hypotheses** about the pair of potential outcomes $\{Y_i(T_i = 1), Y_i(T_i = 0)\}$.
2. We can **define estimands** in terms of $\{Y_i(T_i = 1), Y_i(T_i = 0)\}$ or τ_i, **develop estimators** for those estimands, and then calculate values and standard errors for those estimators.

A common estimand and estimator: The average treatment effect and the difference of means

Say we are interested in the ATE, or $\bar{\tau} = \sum_{i=1}^{n} \tau_i$. What is a good estimator?

Two candidates:

1. The difference of means: $\hat{\bar{\tau}} = \frac{\sum_{i=1}^{n}(T_i Y_i)}{m} - \frac{\sum_{i=1}^{n}((1-T_i)Y_i)}{n-m}$.
2. A difference of means after top-coding the highest Y_i observation (a kind of "winsorized" mean to prevent extreme values from exerting too much influence over our estimator — to increase *precision*).

How would we know which estimator is best for our particular research design?

Let's simulate!

Simulation Step 1: create some data with a known ATE

Notice that we need to *know* the potential outcomes and the treatment assignment in order to learn whether our proposed estimator does a good job.

| Z| y0| y1| |–:|—:|—:| | 0| 0| 10| | 0| 0| 30| | 0| 0| 200| | 0| 1| 91| | 1| 1| 11| | 1| 3| 23| | 0| 4| 34| | 0| 5| 45| | 1| 190| 280| | 1| 200| 220|

```
## The true ATE is 54
```

In reality, we would observe only one of the potential outcomes.

Note that each unit has its own treatment effect.

First make fake data

The table in the previous slide was generated in R with:

```r
# We have ten units
N <- 10
#  y0 is potential outcome to control
y0 <- c(0, 0, 0, 1, 1, 3, 4, 5, 190, 200)
# Each unit has its own treatment effect
tau <- c(10, 30, 200, 90, 10, 20, 30, 40, 90, 20)
# y1 is potential outcome to treatment
y1 <- y0 + tau
# Two blocks, a and b
block <- c("a", "a", "a", "a", "a", "a", "b", "b", "b", "b")
# Z is treatment assignment (Z instead of T in the code)
Z <- c(0, 0, 0, 0, 1, 1, 0, 0, 1, 1)
# Y is observed outcomes
Y <- Z * y1 + (1 - Z) * y0
# The data
dat <- data.frame(Z = Z, y0 = y0, y1 = y1, tau = tau, b = block, Y =
set.seed(12345)
```

Using DeclareDesign

DeclareDesign represents research designs in a few steps shown below:

```
# take just the potential outcomes under treatment and control from c
# fake data
small_dat <- dat[, c("y0", "y1")]

# DeclareDesign first asks you to declare your population
pop <- declare_population(small_dat)

# 5 units assigned to treatment; default is simple random assignment
# probability 0.5
trt_assign <- declare_assignment(Z = complete_ra(N = nrow(small_dat),

# observed Y is y1 if Z=1 and y0 if Z=0
pot_out <- declare_potential_outcomes(Y ~ Z * y1 + (1 - Z) * y0)

# specify outcome and assignment variables
reveal <- declare_reveal(Y, Z)

# the basic research design object includes these four objects
base_design <- pop + trt_assign + pot_out + reveal
```

Using DeclareDesign: make fake data

DeclareDesign renames `y0` and `y1` by default to `Y_Z_0` and `Y_Z_1`:

```
## A simulation is one random assignment of treatment
sim_dat1 <- draw_data(base_design)

## Simulated data (just the first 6 lines)
head(sim_dat1)
```

```
##    y0  y1 Z Y_Z_0 Y_Z_1  Y
## 1   0  10 1     0    10 10
## 2   0  30 1     0    30 30
## 3   0 200 0     0   200  0
## 4   1  91 1     1    91 91
## 5   1  11 0     1    11  1
## 6   3  23 1     3    23 23
```

Using DeclareDesign: define estimand and estimators

No output here. Just define functions and estimators and one estimand.

```
## The estimand
estimandATE <- declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0))

## The first estimator is difference-in-means
diff_means <- declare_estimator(Y ~ Z,
  inquiry = estimandATE,
  .method = lm_robust, se_type = "classical", label = "Diff-Means/OLS
)
```

Using DeclareDesign: define estimand and estimators

```r
## The second estimator is top-coded difference-in-means
diff_means_topcoded_fn <- function(data) {
  data$rankY <- rank(data$Y)
  ## Code the maximum value of Y as the second to maximum value of Y
  data$newY <- with(
    data,
    ifelse(rankY == max(rankY), Y[rankY == (max(rankY) - 1)], Y)
  )
  obj <- lm_robust(newY ~ Z, data = data, se_type = "classical")
  res <- tidy(obj) %>% filter(term == "Z")
  return(res)
}
diff_means_topcoded <- declare_estimator(
  handler = label_estimator(diff_means_topcoded_fn),
  inquiry = estimandATE, label = "Top-coded Diff Means"
)
```

Using DeclareDesign: define estimand and estimators

Here we show how the DD estimators work using our simulated data.

```
## Demonstrate that the estimand works:
estimandATE(sim_dat1)
```

```
##   inquiry estimand
## 1     ATE       54
```

```
## Demonstrate that the estimators estimate
## Estimator 1 (difference in means)
diff_means(sim_dat1)[-c(1, 2, 10, 11)]
```

```
##   estimate std.error statistic p.value conf.low conf.high df
## 1    -39.2     49.41   -0.7934  0.4505   -153.1     74.74  8
```

```
## Estimator 2 (top-coded difference in means)
diff_means_topcoded(sim_dat1)[-c(1, 2, 10, 11)]
```

```
##   estimate std.error statistic p.value conf.low conf.high df
## 1    -37.2     48.21   -0.7716  0.4625   -148.4     73.98  8
```

Then simulate with one randomization

Recall the true ATE:

```
trueATE <- with(sim_dat1, mean(y1 - y0))
with(sim_dat1, mean(Y_Z_1 - Y_Z_0))
```

```
## [1] 54
```

In one experiment (one simulation of the data) here are the simple estimates:

```
## Two ways to calculate the difference of means estimator
est_diff_means_1 <- with(sim_dat1, mean(Y[Z == 1]) - mean(Y[Z == 0]))
est_diff_means_2 <- coef(lm_robust(Y ~ Z,
  data = sim_dat1,
  se = "classical"
))[["Z"]]
c(est_diff_means_1, est_diff_means_2)
```

```
## [1] -39.2 -39.2
```

Then simulate with one randomization

In one experiment (one simulation of the data) here are the estimates after top-coding:

```
## Two ways to calculate the topcoded difference of means estimator
sim_dat1$rankY <- rank(sim_dat1$Y)
sim_dat1$Y_tc <- with(sim_dat1, ifelse(rankY == max(rankY),
  Y[rankY == (max(rankY) - 1)], Y
))
est_topcoded_1 <- with(sim_dat1, mean(Y_tc[Z == 1]) - mean(Y_tc[Z ==
est_topcoded_2 <- coef(lm_robust(Y_tc ~ Z,
  data = sim_dat1,
  se = "classical"
))[["Z"]]
c(est_topcoded_1, est_topcoded_2)
```

```
## [1] -37.2 -37.2
```

Then simulate a different randomization and estimate the ATE with the same estimators

Now calculate your estimate with the same estimators using a **different** randomization. Notice that the answers differ. The estimators are estimating the *same estimand* but now they have a different randomization to work with.

```
# do another random assignment of the treatment in DeclareDesign
# this produces a new simulated dataset with a different random assig
sim_dat2 <- draw_data(base_design)
# the first estimator (difference in means)
coef(lm_robust(Y ~ Z, data = sim_dat2, se = "classical"))[["Z"]]
```

```
## [1] 76.8
```

```
# the second estimator (top-coded difference in means)
sim_dat2$rankY <- rank(sim_dat2$Y)
sim_dat2$Y_tc <- with(sim_dat2, ifelse(rankY == max(rankY),
  Y[rankY == (max(rankY) - 1)], Y
))
coef(lm_robust(Y_tc ~ Z, data = sim_dat2, se = "classical"))[["Z"]]
```

```
## [1] 36.25
```

How do our estimators behave in general for this design?

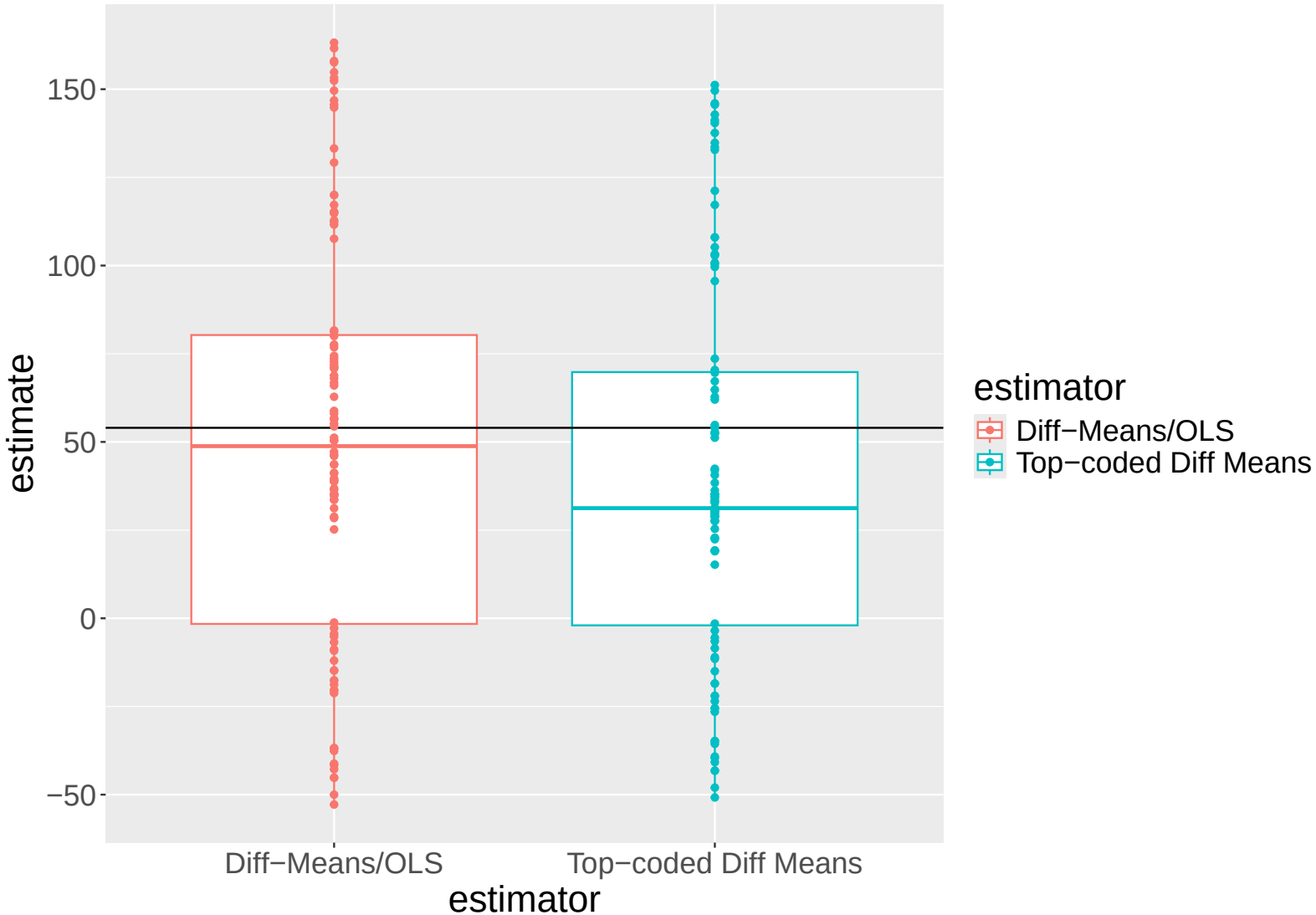
Our estimates vary across randomizations. Do our two estimators vary in the same ways?

```
## Combine into one DeclareDesign design object
## This has the base design, estimand, then our two estimators
design_plus_ests <- base_design + estimandATE + diff_means +
  diff_means_topcoded
## Run 100 simulations (reassignments of treatment) and
## apply the two estimators (diff_means and diff_means_topcoded)
diagnosis1 <- diagnose_design(design_plus_ests,
  bootstrap_sims = 0, sims = 100
)
sims1 <- get_simulations(diagnosis1)
head(sims1[, -c(1:6)])
```

```
##   estimate std.error statistic p.value conf.low conf.high df outco
## 1    -14.8     58.65   -0.2524  0.8071  -150.04    120.44  8
## 2    -18.5     47.70   -0.3878  0.7115  -135.22     98.22  6    ne
## 3     36.4     54.01    0.6740  0.5193   -88.14    160.94  8
## 4     30.4     50.19    0.6057  0.5615   -85.35    146.15  8    ne
## 5     50.4     62.79    0.8027  0.4454   -94.39    195.19  8
## 6     34.4     51.88    0.6631  0.5259   -85.23    154.03  8    ne
```

How do our estimators behave in general for this design?

Our estimates vary across randomizations. Do our two estimators vary in the same ways? How should we interpret this plot?



Which estimator is closer to the truth?

One way to choose among estimators is to choose the one that is **close to the truth** whenever we use it — regardless of the specific randomization.

An "unbiased" estimator is one for which **average of the estimates across repeated designs** is the same as the truth (or $E_R(\hat{\bar{\tau}}) = \bar{\tau}$). An unbiased estimator has "no systematic error" but doesn't guarantee closeness to the truth.

Another measure of closeness is **root mean squared error** (RMSE) which records squared distances between the truth and the individual estimates.

Which estimator is better? (One is closer to the truth on average (RMSE) and is more precise. The other has no systematic error — is unbiased.)

Estimator	Estimator.1	Bias	RMSE	SD Estimate	Power
Diff-Means/OLS	Diff-Means/OLS	-2.51	58.07	58.31	0.13
Top-coded Diff Means	Top-coded Diff Means	-13.18	55.73	54.43	0.14

Unbiased and biased estimators

Summary:

- We have a *choice* of both estimands and estimators
- A good estimator performs well regardless of the particular randomization of a given design. And *performs well* can mean "unbiased" and/or "low mse" (or "consistent" — which means increasingly close to the truth as the sample size increases).
- We can learn about how a given estimator performs in a given study using simulation.

Block randomization

Block-randomized experiments are a collection of mini-experiments

What is the ** ATE** estimand in a block-randomized experiment?

If we think of the unit-level ATE as: $(1/N)\sum_{i=1}^{N} y_{i,1} - y_{i,0}$ then we could re-express this equivalently using the ATE in block j is ATE_j as follows:

$$ATE = \frac{1}{J}\sum_{j=1}^{J}\sum_{i=1}^{N_j}\frac{y_{i,1} - y_{i,0}}{N_j} = \sum_{j=1}^{J}\frac{N_j}{N}ATE_j \tag{1}$$

And it would be natural to *estimate* this quantity by plugging in what we can calculate:

$$\widehat{ATE} = \sum_{j=1}^{J}\frac{N_j}{N}\widehat{ATE}_j \tag{2}$$

Block-randomized experiments are a collection of mini-experiments

And we could *define* the standard error of the estimator by also just averaging the within-block standard errors (if our blocks are large enough):

$$SE(\widehat{ATE}) = \sqrt{\sum_{j=1}^{J} (\frac{N_j}{N})^2 SE^2(\widehat{ATE}_j)}$$

Estimating the ATE in block-randomized experiments

One approach to estimation simply replaces ATE_j with $\widehat{ATE}$ above:

```
with(dat, table(b, Z))
```

```
##     Z
## b    0 1
##   a 4 2
##   b 2 2
```

We have 6 units in block a, 2 of which are assigned to treatment, and 4 units in block b, 2 of which are assignment to treatment.

Estimating the ATE in block-randomized experiments

One approach to estimation simply replaces ATE_j with $\widehat{ATE}$ above:

```
datb <- dat %>%
  group_by(b) %>%
  summarize(
    nb = n(), pb = mean(Z), estateb = mean(Y[Z == 1]) - mean(Y[Z == 0
    ateb = mean(y1 - y0), .groups = "drop"
  )
datb
```

```
## # A tibble: 2 x 5
##   b        nb    pb estateb  ateb
##   <chr> <int> <dbl>   <dbl> <dbl>
## 1 a         6 0.333    16.8    60
## 2 b         4 0.5     246.     45
```

```
## True ate by block:
with(dat, mean(y1 - y0))
```

```
## [1] 54
```

```
## This is another way to calculate the true ate
with(datb, sum(ateb * (nb / sum(nb))))
```

Estimating the ATE in block-randomized experiments

One approach is to estimate the overall ATE using block-size weights:

```
## Showing that difference_in_means uses the blocksize weight.
e1 <- difference_in_means(Y ~ Z, blocks = b, data = dat)
e2 <- with(datb, sum(estateb * (nb / sum(nb))))
c(coef(e1)[["Z"]], e2)
```

```
## [1] 108.2 108.2
```

Estimating the ATE in block-randomized experiments

Notice that this is **not** the same as either of the following:

```
## Ignoring blocks
e3 <- lm(Y ~ Z, data = dat)
coef(e3)[["Z"]]
```

```
## [1] 131.8
```

```
## With block fixed effects
e4 <- lm(Y ~ Z + block, data = dat)
coef(e4)[["Z"]]
```

```
## [1] 114.8
```

How do they differ? (The first ignores the blocks. The second uses a different set of weights that are created by use of "fixed effects" or "indicator" or "dummy" variables.)

Which estimator should we use?

We now have three estimators each with a different estimate (imagining they all target the same estimand):

```
c(coef(e1)[["Z"]], coef(e3)[["Z"]], coef(e4)[["Z"]])
```

```
## [1] 108.2 131.8 114.8
```

Which estimator should we use for this design? We can set up a DeclareDesign simulation to figure this out.

```
## declare a new base design that includes the block indicator b
base_design_blocks <-
  # declare the population
  declare_population(dat[, c("b", "y0", "y1")]) +
  # tell DD that b indicates block and to assign 2 treated units in e
  declare_assignment(
    Z = block_ra(m = 2, blocks = b),
    Z_cond_prob = obtain_condition_probabilities(Z, declaration = dec
  ) +
  # relationship of potential outcomes to observed outcome
  declare_potential_outcomes(Y ~ Z * y1 + (1 - Z) * y0) +
  # observed outcome and treatment assignment
  declare_reveal(Y, Z)
```

Which estimator should we use?

```r
# the estimand is the average treatment effect
estimandATEb <- declare_inquiry(ATE = mean(Y_Z_1 - Y_Z_0))

# three different estimators
est1 <- declare_estimator(Y ~ Z,
  inquiry = estimandATEb, .method = lm_robust,
  label = "Ignores Blocks"
)
est2 <- declare_estimator(Y ~ Z,
  inquiry = estimandATEb, .method = difference_in_means, blocks = b,
  label = "DiM: Block-Size Weights"
)
est3 <- declare_estimator(Y ~ Z,
  inquiry = estimandATEb, .method = lm_robust,
  weights = (Z / Z_cond_prob) + ((1 - Z) / (Z_cond_prob)),
  label = "LM: Block Size Weights"
)
```

Which estimator should we use?

```
# two more estimators
est4 <- declare_estimator(Y ~ Z,
  inquiry = estimandATEb,
  .method = lm_robust, fixed_effects = ~b, label = "Precision Weights
)
est5 <- declare_estimator(Y ~ Z + b,
  inquiry = estimandATEb,
  .method = lm_robust, label = "Precision Weights (LSDV)"
)

## new design object has the base design, the estimand, and five esti
design_blocks <- base_design_blocks + estimandATEb +
  est1 + est2 + est3 + est4 + est5
```

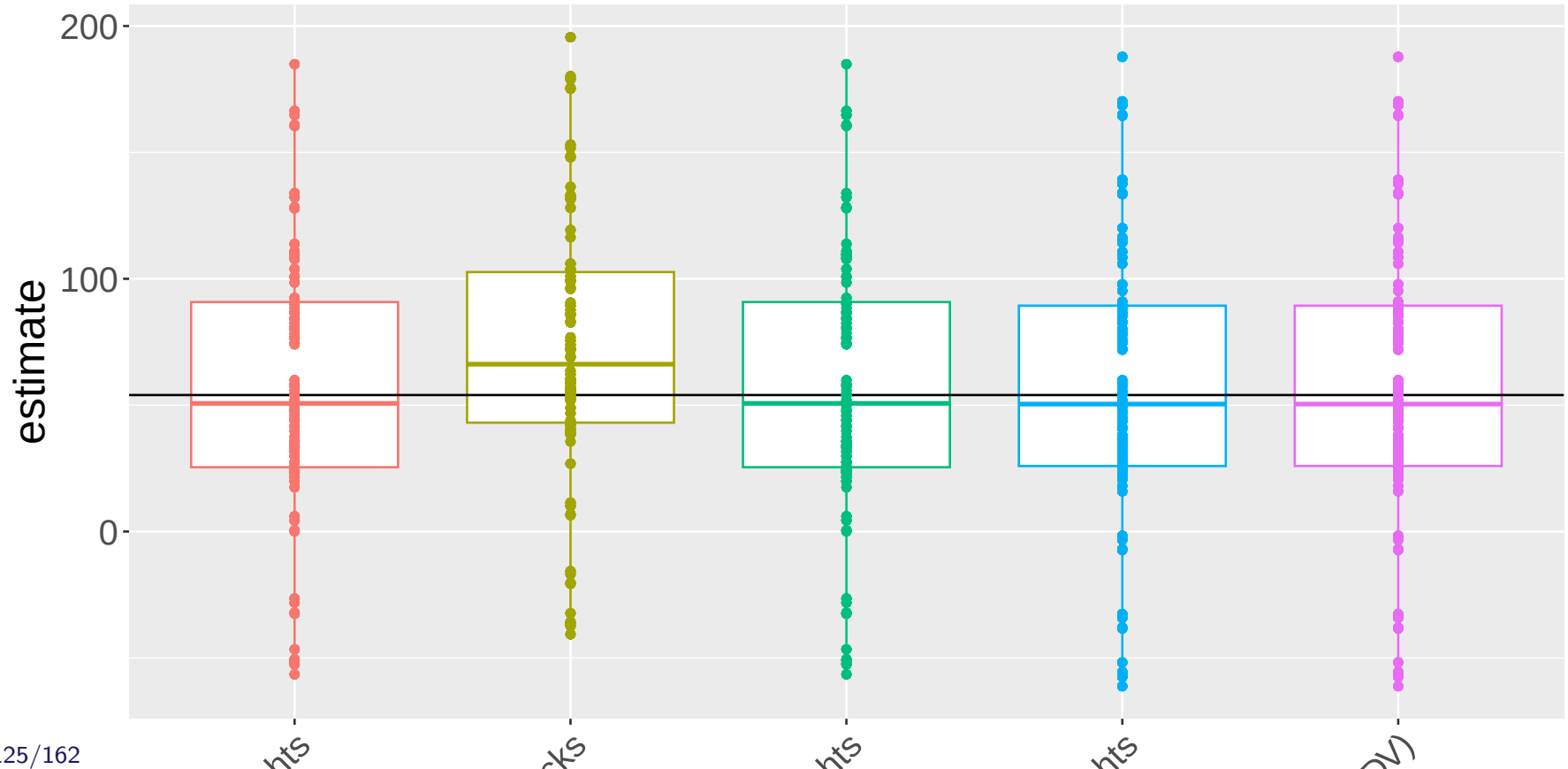
Then we will run 10,000 simulations (reassign treatment 10,000 times) and summarize the estimates produced by each of these five estimators.

Which estimator should we use?

```
##
## Attaching package: 'future'

## The following object is masked from 'package:survival':
##
##     cluster
```

How should we interpret this plot?



Which estimator is closer to the truth?

Which estimator works better on this design and these data?

Design	Inquiry	Estimator	Outcome	Term	N Sims	Estimator.1	Bias	RMSE	SD Estimate	Power	Coverage
design_blocks	ATE	DiM: Block-Size Weights	Y	Z	1000	DiM: Block-Size Weights	2.94	53.07	53.02	0.25	0.76
design_blocks	ATE	Ignores Blocks	Y	Z	1000	Ignores Blocks	18.06	56.52	53.58	0.11	0.98
design_blocks	ATE	LM: Block Size Weights	Y	Z	1000	LM: Block Size Weights	2.94	53.07	53.02	0.09	0.94
design_blocks	ATE	Precision Weights	Y	Z	1000	Precision Weights	2.70	55.54	55.50	0.12	0.93
design_blocks	ATE	Precision Weights (LSDV)	Y	Z	1000	Precision Weights (LSDV)	2.70	55.54	55.50	0.12	0.93

Cluster randomization

In cluster-randomized experiments, units are randomized as a group (cluster) to treatment I

- **Example 1:** an intervention is randomized across neighborhoods, so **all** households in a neighborhood will be assigned to the same treatment condition, but different neighborhoods will be assigned different treatment conditions.
- **Example 2:** an intervention is randomized across people and each person is measured four times after treatment, so our data contain four rows per person.
- **Not An Example 1:** Neighborhoods are chosen for the study. Within each neighborhood about half of the people are assigned to treatment and half to control. (What kind of study is this? It is not a cluster-randomized study.)
- **Not an Example 2:** an intervention is randomized to some neighborhoods and not to others, the outcomes include measurements of neighborhood-level trust in government and total land area in the neighborhood devoted to gardens. (Sometimes a cluster randomized experiment can be turned into a simple randomized experiment. Or may contain more than one possible approach to analysis and interpretation.)

How might the distribution of test statistics and estimators differ from an experiment where individual units (not clusters) are randomized?

Estimating the ATE in cluster-randomized experiments

Bias problems in cluster-randomized experiments:

- When clusters are the same size, the usual difference-in-means estimator is unbiased.
- But be careful when clusters have different numbers of units or you have very few clusters because then treatment effects may be correlated with cluster size.
- When cluster size is related to potential outcomes, the usual difference-in-means estimator is biased.
  https://declaredesign.org/blog/bias-cluster-randomized-trials.html

Estimating the SE for the ATE in cluster-randomized experiments I

- **Misleading statistical inferences:** The default SE will generally underestimate precision in such designs and thus produce tests with false positive rates that are too high (or equivalently confidence intervals coverage rates that are too low).
- The "cluster robust standard errors" implemented in common software work well **when the number of clusters is large** (like more than 50 in some simulation studies).
- The default cluster-appropriate standard errors in `lm_robust` (the `CR2` SEs) work better than the common approach in Stata (as of this writing).
- The wild bootstrap helps control error rates but gives up statistical power much more than perhaps necessary in a cluster randomized study where direct randomization inference is possible.
- When in doubt, one can produce p-values by direct simulation (direct randomization inference) to see if they agree with one of the cluster robust approaches.

Estimating the SE for the ATE in cluster-randomized experiments II

Overall, it is worth simulating to study the performance of your estimators, tests, and confidence intervals if you have any worries or doubts.

An example of estimation

Imagine we had data from 10 clusters with either 100 people (for 2 clusters) or 10 people per cluster (for 8 clusters). The total size of the data is 280.

```
## # A tibble: 6 x 6
## # Groups:   clus_id [2]
##   clus_id indiv Y_Z_0 Y_Z_1     Z     Y
##   <chr>   <chr> <dbl> <dbl> <int> <dbl>
## 1 01      010    4.51  4.61     0  4.51
## 2 01      035    4.63  4.73     0  4.63
## 3 01      068    4.76  4.86     0  4.76
## 4 03      205    3.13  4.13     1  4.13
## 5 03      206    2.41  3.41     1  3.41
## 6 03      208    2.95  3.95     1  3.95
```

An example of estimation

Which estimator should we use? Which test should we use? On what basis should we choose among these approaches?

```
lmc1 <- lm_robust(Y ~ Z, data = dat1)
lmc2 <- lm_robust(Y ~ Z, clusters = clus_id, data = dat1)
lmc3 <- lm_robust(Y ~ Z + cl_sizeF, clusters = clus_id, data = dat1)
tidy(lmc1)[2, ]
```

```
##   term estimate std.error statistic p.value conf.low conf.high  df
## 2    Z   0.3024    0.1207     2.504 0.01284  0.06471    0.5401 278
```

```
tidy(lmc2)[2, ]
```

```
##   term estimate std.error statistic p.value conf.low conf.high
## 2    Z   0.3024     1.079    0.2804   0.796   -2.969     3.574 3.2
```

```
tidy(lmc3)[2, ]
```

```
##   term estimate std.error statistic p.value conf.low conf.high
## 2    Z   0.3024     0.306    0.9882  0.4386   -1.194     1.799 1.7
```

Use simulation to assess estimators and tests

If you look at the code for the slides you will see that we simulate the design 5000 times, each time calculating an estimate and confidence interval for different estimators of the ATE.

What should we learn from this table? (Coverage? `sd_estimate` versus `mean_se`).

```
## 1 coefficient  not defined because the design matrix is rank defic
##
## 1 coefficient  not defined because the design matrix is rank defic
##
## 1 coefficient  not defined because the design matrix is rank defic

## 1 coefficient  not defined because the design matrix is rank defic
##
## 1 coefficient  not defined because the design matrix is rank defic
##
## 1 coefficient  not defined because the design matrix is rank defic
##
## 1 coefficient  not defined because the design matrix is rank defic
##
## 1 coefficient  not defined because the design matrix is rank defic
##
```

Use simulation to assess estimators and tests

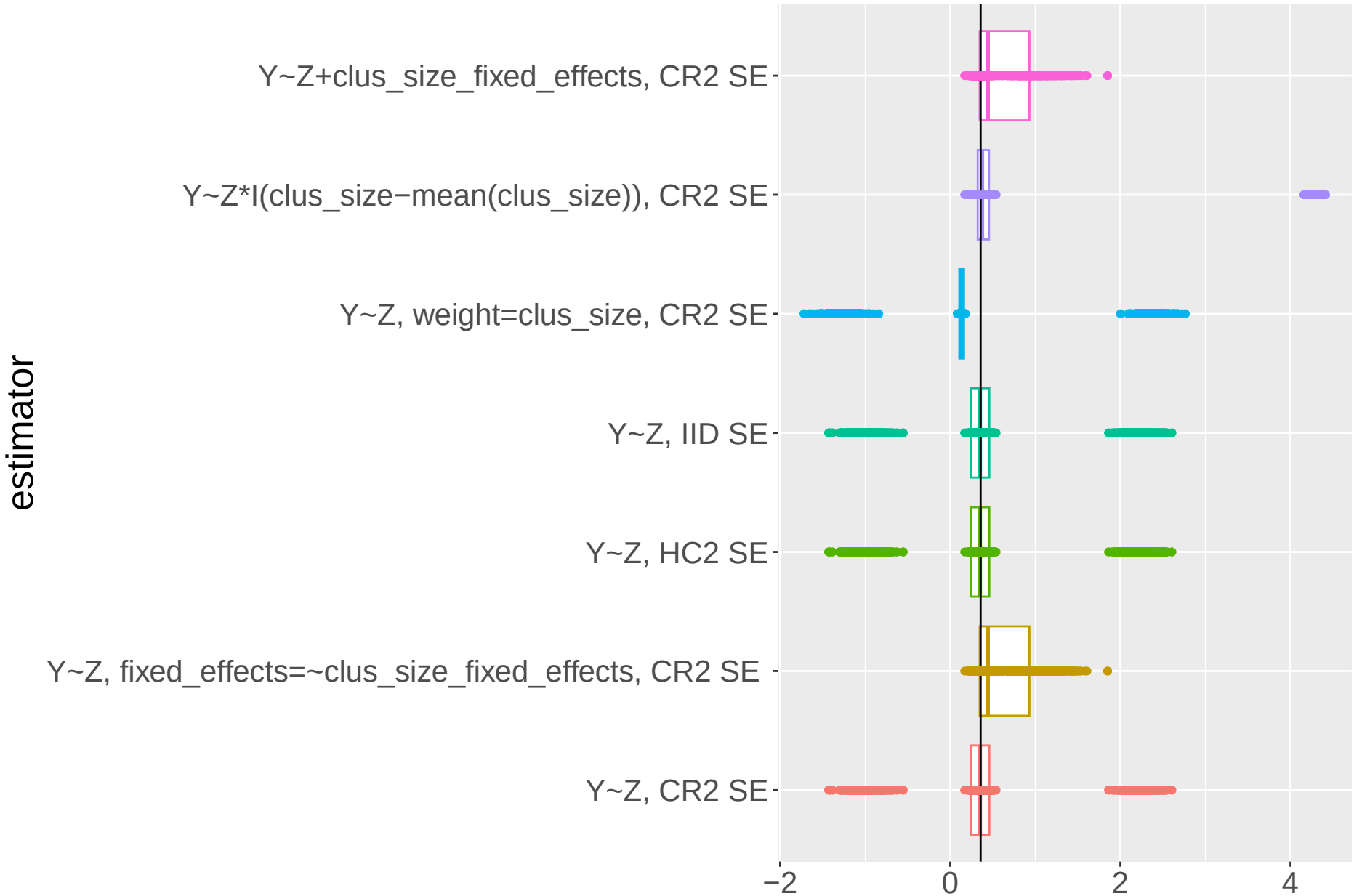
What should we learn from this table? (Bias? Closeness to truth?)

Table 8: Estimator and Test Performance in 5000 simulations of the cluster randomized design for different estimators and confidence intervals

estimator	bias	rmse
Y~Z*I(clus_size-mean(clus_size)), CR2	0.762	1.735
Y~Z, cl_size fe, CR2	0.280	0.450
Y~Z, CR2	0.087	1.089
Y~Z, HC2	0.087	1.089
Y~Z, IID	0.087	1.089
Y~Z*I(cl_size-mean(cl_size)), CR2	0.280	0.450
Y~Z+cl_sizeF, CR2	-0.062	1.237

Use simulation to assess estimators and tests

How should we interpret this plot?



Summary of estimation and testing in cluster-randomized trials

- Cluster randomized trials pose special problems for standard approaches to estimation and testing.
- If randomization is at the cluster level, then uncertainty arises from the cluster level randomization.
- If we have enough clusters, then one of the "cluster robust" standard errors can help us produce confidence intervals with correct coverage. **Cluster robust standard errors require many clusters**.
- If cluster size (or characteristic) is related to effect size, then we can have bias (and we need to adjust somehow).

Binary outcomes

Binary outcomes: Set up our data for simulation in DeclareDesign

```
# population size
N <- 20
# declare the population
thepop_bin <- declare_population(
  N = N, x1 = draw_binary(prob = .5, N = N),
  x2 = rnorm(N)
)
# declare the potential outcomes
thepo_bin <- declare_potential_outcomes(Y ~ rbinom(
  n = N, size = 1,
  prob = 0.5 + 0.05 * Z + x1 * .05
))
# two possible targets: difference in means or difference in log-odds
thetarget_ate <- declare_estimand(ate = mean(Y_Z_1 - Y_Z_0))
```

```
## Warning in declare_estimand(ate = mean(Y_Z_1 - Y_Z_0)): 'declare_e
## Use 'declare_inquiry' instead.
## See help("Deprecated")
```

```
thetarget_logodds <- declare_estimand(
```

Binary outcomes: Set up our data for simulation in DeclareDesign

```
# declare how treatment is assigned
# m units are assigned to levels of treatment Z
theassign_bin <- declare_assignment(Z = conduct_ra(N = N, m = floor(N
# declare what outcome values are revealed for possible values of Z
thereveal_bin <- declare_reveal(Y, Z)
# pull this all together: population, potential outcomes, assignment,
## outcome values connected to Z
des_bin <- thepop_bin + thepo_bin + theassign_bin + thereveal_bin
# then make one draw (randomize treatment once)
set.seed(12345)
dat2 <- draw_data(des_bin)
```

Binary outcomes: Estimands I

How would we interpret the following true quantities or estimands? (`Y_Z_1`, `Y_Z_0` are potential outcomes, `Y` is observed, `x1`, `x2` are covariates, `Z` is treatment assignment. Here N=20.

```
## Look at the first 6 observations only:
head(dat2[, -7])
```

```
##    ID x1      x2 Y_Z_0 Y_Z_1 Z
## 1 01  1 -0.1162     0     1 0
## 2 02  1  1.8173     0     1 1
## 3 03  1  0.3706     0     1 0
## 4 04  1  0.5202     1     1 0
## 5 05  0 -0.7505     1     0 1
## 6 06  0  0.8169     0     1 0
```

Binary outcomes: Estimands II

How would we interpret the following true quantities or estimands? (`Y_Z_1`, `Y_Z_0` are potential outcomes, `Y` is observed, `x1`, `x2` are covariates, `Z` is treatment assignment. Here N=20.

```
ate_bin <- with(dat2, mean(Y_Z_1 - Y_Z_0))
bary1 <- mean(dat2$Y_Z_1)
bary0 <- mean(dat2$Y_Z_0)
diff_log_odds_bin <- with(
  dat2,
  log(bary1 / (1 - bary1)) - log(bary0 / (1 - bary0))
)
c(
  bary1 = bary1, bary0 = bary0, true_ate = ate_bin,
  true_diff_log_odds = diff_log_odds_bin
)
```

```
##                bary1                bary0             true_ate true_diff
##                 0.55                 0.55                 0.00
```

Binary outcomes: Estimands III

Do you want to estimate the difference in log-odds?

$$\delta = \log \frac{\bar{y}_1}{1 - \bar{y}_1} - \log \frac{\bar{y}_0}{1 - \bar{y}_0} \tag{3}$$

Or the difference in proportions?

$$\bar{\tau} = \bar{y}_1 - \bar{y}_0 \tag{4}$$

Recall that $\bar{y}_1$ is the *proportion* of $y_1 = 1$ in the data.

Freedman (2008b) shows us that the logit coefficient estimator is a biased estimator of the difference in log-odds estimand. He also shows an unbiased estimator of that estimand.

We know that the difference of proportions in the sample should be an unbiased estimator of the difference of proportions.

An example of estimation I

How should we interpret the following estimates? (What does the difference of means estimator require in terms of assumptions? What does the logistic regression estimator require in terms of assumptions?)

```
lmbin1 <- lm_robust(Y ~ Z, data = dat2)
glmbin1 <- glm(Y ~ Z, data = dat2, family = binomial(link = "logit"))
library(broom)
tidy(lmbin1)[2, ]
```

```
##   term estimate std.error statistic p.value conf.low conf.high df
## 2    Z  -0.4048    0.2159    -1.875 0.07716  -0.8584   0.04884 18
```

```
tidy(glmbin1)[2, ]
```

```
## # A tibble: 1 x 5
##   term  estimate std.error statistic p.value
##   <chr>    <dbl>     <dbl>     <dbl>   <dbl>
## 1 Z        -1.90      1.22     -1.55   0.120
```

An example of estimation II

What about with covariates? Why use covariates?

```
lmbin2 <- lm_robust(Y ~ Z + x1, data = dat2)
glmbin2 <- glm(Y ~ Z + x1, data = dat2, family = binomial(link = "log

tidy(lmbin2)[2, ]
```

```
##   term estimate std.error statistic p.value conf.low conf.high df
## 2    Z  -0.4058    0.2179    -1.862 0.07996  -0.8656   0.05398 17
```

```
tidy(glmbin2)[2, ]
```

```
## # A tibble: 1 x 5
##   term  estimate std.error statistic p.value
##   <chr>    <dbl>     <dbl>     <dbl>   <dbl>
## 1 Z        -1.90      1.22     -1.55   0.120
```

An example of estimation III

Let's compare our estimates

```
c(
  dim = coef(lmbin1)[["Z"]],
  dim_x1 = coef(lmbin2)[["Z"]],
  glm = coef(glmbin1)[["Z"]],
  glm_x1 = coef(glmbin2)[["Z"]]
)
```

```
##      dim  dim_x1      glm  glm_x1
## -0.4048 -0.4058 -1.8971 -1.9025
```

An example of estimation: The Freedman plugin estimators I

No covariate:

```
freedman_plugin_estfn1 <- function(data) {
  glmbin <- glm(Y ~ Z, data = dat2, family = binomial(link = "logit")
  preddat <- data.frame(Z = rep(c(0, 1), nrow(dat2)))
  preddat$yhat <- predict(glmbin, newdata = preddat, type = "response
  bary1 <- mean(preddat$yhat[preddat$Z == 1])
  bary0 <- mean(preddat$yhat[preddat$Z == 0])
  diff_log_odds <- log(bary1 / (1 - bary1)) - log(bary0 / (1 - bary0)
  return(data.frame(estimate = diff_log_odds))
}
```

An example of estimation: The Freedman plugin estimators II

With covariate:

```r
freedman_plugin_estfn2 <- function(data) {
  N <- nrow(data)
  glmbin <- glm(Y ~ Z + x1, data = data, family = binomial(link = "lo
  preddat <- data.frame(Z = rep(c(0, 1), each = N))
  preddat$x1 <- rep(data$x1, 2)
  preddat$yhat <- predict(glmbin, newdata = preddat, type = "response
  bary1 <- mean(preddat$yhat[preddat$Z == 1])
  bary0 <- mean(preddat$yhat[preddat$Z == 0])
  diff_log_odds <- log(bary1 / (1 - bary1)) - log(bary0 / (1 - bary0)
  return(data.frame(estimate = diff_log_odds))
}
```

Let's compare our estimates from the six different estimators

```
##         dim      dim_x1         glm      glm_x1    freedman freeman_x1
##     -0.4048     -0.4058     -1.8971     -1.9025     -1.8971     -
1.9020
```

An example of using DeclareDesign to assess our estimators I

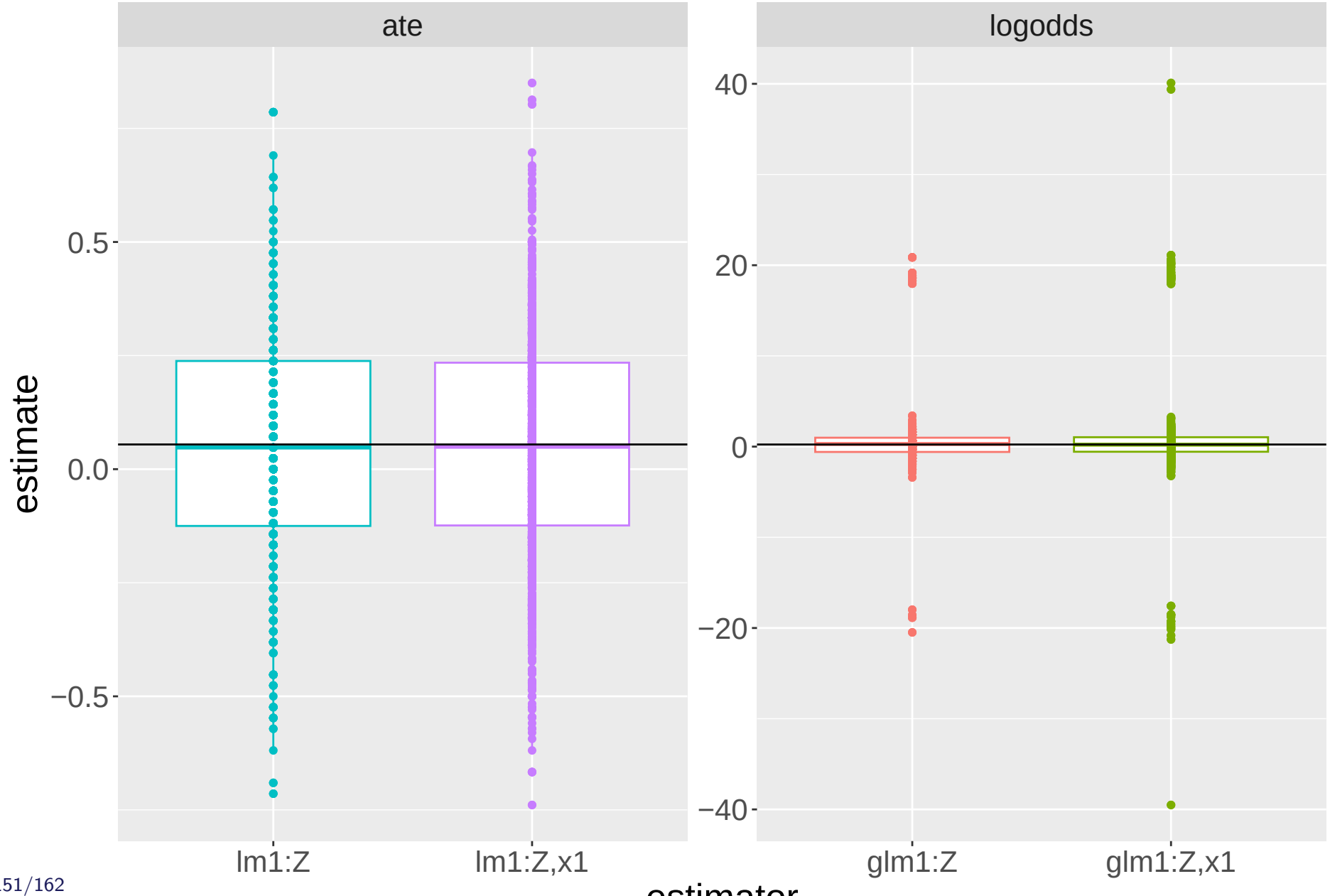
```
# declare 4 estimators for DD
# first estimator: linear regression with ATE as target
estb1 <- declare_estimator(Y ~ Z,
  .method = lm_robust, label = "lm1:Z",
  inquiry = thetarget_ate
)
# second estimator: linear regression with covariate, with ATE as tar
estb2 <- declare_estimator(Y ~ Z + x1,
  .method = lm_robust, label = "lm1:Z,x1",
  inquiry = thetarget_ate
)
# third estimator: logistic regression, with log odds as target
estb3 <- declare_estimator(Y ~ Z,
  .method = glm, family = binomial(link = "logit"),
  label = "glm1:Z", inquiry = thetarget_logodds
)
# fourth estimtor: logistic regression with covariate, with log odds
estb4 <- declare_estimator(Y ~ Z + x1,
  .method = glm, family = binomial(link = "logit"),
```

An example of using DeclareDesign to assess our estimators II

```
# Pull together: des_bin is population, potential outcomes, assignmen
# outcome values connected to Z.  We add the two targets and four est
des_bin_plus_est <- des_bin + thetarget_ate + thetarget_logodds +
  estb1 + estb2 + estb3 + estb4
```

Using simulation to assess our estimators

How should we interpret this plot? (Differences in scales make it difficult.)



Which estimator is closer to the truth?

Which estimator works better on this design and these data?

```
## `summarise()` has grouped output by 'estimator'. You can override
## `.groups` argument.
```

Table 9: Estimator and Test Performance in 5000 simulations of the different estimators and confidence intervals for a binary outcome and completely randomized design.

est	estimand	bias	rmse	power	coverage	sd_est	mean_se
glm1:Z	logodds	0.349	3.330	0.019	0.992	3.479	100.056
glm1:Z,x1	logodds	0.383	4.678	0.016	0.994	4.867	208.045
lm1:Z	ate	-0.009	0.183	0.066	0.967	0.240	0.242
lm1:Z,x1	ate	-0.008	0.194	0.078	0.967	0.253	0.248

Other topics in estimation

Covariance adjustment: Estimands

In general, simply "controlling for" produces a biased estimator of the ATE **or** ITT estimand. See for example Lin (2013) and Freedman (2008a). Lin (2013) shows how to reduce this bias and, importantly, that this bias tends to be small as the sample size increases.

Effects that differ by groups I

If our theory suggests that effects should differ by group, how can we assess evidence for or against such claims?

- We can **design** for an assessment of this theory by creating a block-randomized study — with blocked defined by the theoretically relevant groups.
- We can **plan** for such an assessment by (1) **pre-registering specific subgroup analyses** (whether or not we block on that group in the design phase) and (2) making sure to measure group membership during baseline data collection pre-treatment

Effects that differ by groups II

- If we have not planned ahead, subgroup-specific analyses can be useful as explorations but should not be understood as confirmatory: they can too easily create problems of testing too many hypotheses thus inflated false positive rates.
- We **should not use groups formed by treatment**. (This is either "mediation analysis" or "conditioning on post-treatment variables" and deserves its own module).

Final thoughts on basics of estimation

- Counterfactual causal estimands are unobserved functions of potential outcomes.
- Estimators are recipes or functions that use observed data to learn about an estimand.
- Good estimators produce estimates that are close to the true estimand
- (Connecting estimation with testing) Standard errors of estimators allow us to calculate confidence intervals and p-values. Certain estimators have larger or smaller (or more or less correct) standard errors.
- You can assess the utility of a chosen estimator for a chosen estimand by simulation.

Summary of the day

Summary

- Statistics helps us **infer** about partially observable counterfactual causal effects
- We can **hypothesize about those effects** and summarize how much evidence our design and data provide.
- We can **calculate best guesses** about aggregates of causal effects.
- We can test hypotheses about the aggregates, too.
- The statistical properties of our tests and estimators arise from the experimental design, not from a relationship between sample and population.

References I

 Benjamini, Yoav and Yosef Hochberg (1995). "Controlling the false discovery rate: a practical and powerful approach to multiple testing". In: *Journal of the Royal Statistical Society* 57.1, pp. 289–300. ISSN: 0035-9246. arXiv: 95/57289 [0035-9246]. URL: http://www.jstor.org/stable/2346101.

 Bowers, Jake, Bruce A Desmarais, et al. (2018). "Models, methods and network topology: Experimental design for the study of interference". In: *Social Networks* 54, pp. 196–208.

 Bowers, Jake, Mark Fredrickson, and Peter M Aronow (2016). "Research Note: A more powerful test statistic for reasoning about interference between units". In: *Political Analysis* 24.3, pp. 395–403.

 Bowers, Jake, Mark M Fredrickson, and Costas Panagopoulos (2013). "Reasoning about Interference Between Units: A General Framework". In: *Political Analysis* 21.1, pp. 97–124.

 Caughey, Devin, Allan Dafoe, and Jason Seawright (2017). "Nonparametric combination (NPC): A framework for testing elaborate theories". In: *The Journal of Politics* 79.2, pp. 688–701.

 Freedman, David A. (2008a). "On regression adjustments to experimental data". In: *Advances in Applied Mathematics* 40.2, pp. 180–193.

 — (2008b). "Randomization does not justify logistic regression". In: *Statistical Science* 23.2, pp. 237–249.

- Gerber, Alan S and Donald P Green (2012). *Field Experiments: Design, Analysis, and Interpretation*. New York, NY: W.W. Norton.
- Hansen, Ben B. and Jake Bowers (2008). "Covariate balance in simple, stratified and clustered comparative studies". In: *Statistical Science* 23.2, pp. 219–236.
- Hartman, Erin and F Daniel Hidalgo (2018). "An equivalence approach to balance and placebo tests". In: *American Journal of Political Science* 62.4, pp. 1000–1013.
- Hodges, J.L. and E.L. Lehmann (1963). "Estimates of location based on rank tests". In: *Ann. Math. Statist* 34, pp. 598–611.
- Lin, Winston (Mar. 2013). "Agnostic notes on regression adjustments to experimental data: Reexamining Freedman's critique". en. In: *The Annals of Applied Statistics* 7.1, pp. 295–318. ISSN: 1932-6157. (Visited on 10/07/2016).
- Rosenbaum, P R (2010). *Design of observational studies*. New York [etc.]: Springer.
- Rosenbaum, Paul R (2010). *Design of Observational Studies*. New York, NY: Springer.
- — (1993). "Hodges-Lehmann Point Estimates of Treatment Effect in Observational Studies". In: *Journal of the American Statistical Association* 88.424, pp. 1250–1253. ISSN: 01621459.