Randomization Inference (RI), t-test Confidence Intervals (CIs), and Power Analysis (PA)

Ethan Moody

October 2023

Contents

1 What happens when pilgrims attend the Hajj pilgrimage to Mecca?

1.1 State a null hypothesis

State the sharp-null hypothesis that you will be testing.

```
# Answer provided in narrative form below
```

Answer: The sharp null hypothesis holds that the treatment effect is zero for all subjects (i.e., the potential outcomes to treatment and control are equivalent for each and every individual). In the context of this test, the sharp null hypothesis that we're testing is that, *for every individual respondent, there is no change in beliefs about members of other countries as a result of successfully attending the Hajj* (that is, every respondent's potential outcome to control — their views toward members of other countries after *not* attending the Hajj — is the same as their potential outcome to treatment — their views toward members of other countries after attending the Hajj).

1.2 Group by average

Using `data.table`, group the data by `success` and report whether views toward others are generally more positive among lottery winners or lottery non-winners. This answer should be of the form `d[ , .(mean_views = ...), keyby = ...]` where you have filled in the `...` with the appropriate functions and variables.

```
# the result should be a data.table with two columns and two rows

# Compute mean `views` across `success` groups (control and treatment)
hajj_group_mean <- d[, .(mean_views = mean(views)), keyby = success]

# from the `hajj_group_mean` produce a single, numeric vector that is the ate.
# check that it is numeric using `class(hajj_ate)`

# Compute ATE based on difference in `success` group means of `views`
hajj_ate        <- hajj_group_mean[success == 1, mean_views] -
                   hajj_group_mean[success == 0, mean_views]

# Display class for `hajj_ate` to check for numeric result
class(hajj_ate)
```

```
## [1] "numeric"
```

Answer: The `hajj_group_mean` (i.e., mean `views` across each `success` group) is displayed in the table below:

success	mean_views
0	1.868304
1	2.343137

Based on this data, the `hajj_ate` is equal to 0.4748337. Both this value and the group means above indicate that *views toward others are generally more positive among lottery winners (Hajj attendees) than lottery non-winners.*

```
## do your work to conduct the randomization inference here.
## as a reminder, RI will randomly permute / assign the treatment variable
## and recompute the test-statistic (i.e. the mean difference) under each permutation
## this should be a numeric vector that has a length equal to the number
```

```r
## of RI permutations you ran

# Create randomization inference function for modular and reusable code
ri <- function(permutations = 10000) {

  # Create vector to store test-statistic (ATE) under each permutation
  hajj_ate_vector <- NA

  # success_group1  <- NA   # For checking treatment group proportions (TGP)
  # success_group0  <- NA   # For checking control group proportions (CGP)

  # Simulate 10000 random permutations of treatment assignment
  for(perm in 1:permutations) {

    # Add computed test-statistics to vector after treatment randomization
    hajj_ate_vector[perm] <- d[, .(mean_views = mean(views)),
                               keyby = .(sample(success))][, diff(mean_views)]

    # success_group1[perm]  <- d[, sum(success == 1)]  # For TGP
    # success_group0[perm]  <- d[, sum(success == 0)]  # For CGP
  }

  # Return vector of test-statistics
  return(hajj_ate_vector)

  # return(list(hajj_ate_vector, success_group1, success_group0))  # For TGP/CGP
}

# Run randomization inference function with 10000 permutations
hajj_ri_distribution <- ri(10000)
```

1.3 Randomization inference: At least as large

C. How many of the simulated random assignments generate an estimated ATE that is at least as large as the actual estimate of the ATE? Conduct your work in the code chunk below, saving the results into `hajj_count_larger`, but also support your coding with a narrative description. In that narrative description (and throughout), use R's "inline code chunks" to write your answer consistent with each time your run your code.

```r
# length 1 numeric vector from comparison of `hajj_ate` and `hajj_ri_distribution`

# Determine number of randomizations with an ATE >= est.ATE
hajj_count_larger <- as.numeric(sum(hajj_ri_distribution >= hajj_ate))

# Display result
hajj_count_larger
```

```
## [1] 12
```

Answer: Only 12 of the 10000 permutations/random assignments generated an estimated ATE that is at least as large as the actual estimated ATE of 0.4748337.

1.4 Randomization inference: one-sided p-value

If there are `hajj_count_larger` (12) randomizations that are larger than `hajj_ate` (0.4748337), what is the *one-tailed* p-value? Both write the code in the following chunk, and include a narrative description of the result following your code.

```
# length 1 numeric vector

# Determine one-tailed p-value corresponding to `hajj_count_larger` result
hajj_one_tailed_p_value <- as.numeric(
  hajj_count_larger / length(hajj_ri_distribution))

# Display result
hajj_one_tailed_p_value
```

```
## [1] 0.0012
```

Answer: The one-tailed p-value is 0.0012, which suggests that only a very small proportion of randomizations (i.e., only 12 of the 10000 permutations/random assignments) are at least as large or larger than the actual estimated ATE of 0.4748337. This result indicates that the original difference in means we calculated for `views` across `success` groups is *meaningful* and not likely to arise just by chance alone (or by an "unlucky" randomization) under the assumption of the sharp null hypothesis.

1.5 Randomization inference: two-sided p-value

Now, conduct a similar test, but for a two-sided p-value. You can either use two tests, one for larger than and another for smaller than; or, you can use an absolute value (`abs`). Both write the code in the following chunk, and include a narrative description of the result following your code.

```
# length 1 numeric vector

# Determine number of randomizations with an ATE <= -1 * est. ATE
hajj_count_smaller <- as.numeric(sum(hajj_ri_distribution <= (-1 * hajj_ate)))

# Determine number of randomizations with an ATE at least as extreme as est. ATE
hajj_count_extreme <- as.numeric(sum(hajj_count_larger, hajj_count_smaller))

# Determine two-tailed p-value corresponding to `hajj_count_extreme` result
hajj_two_tailed_p_value <- as.numeric(
  hajj_count_extreme / length(hajj_ri_distribution))

# Display result
hajj_two_tailed_p_value
```

```
## [1] 0.0033
```

Answer: The two-tailed p-value is 0.0033, which — like the one-tailed p-value — suggests that only a very small proportion of randomizations (i.e., only 33 of the 10000 permutations/random assignments) are at least as extreme as the actual estimated ATE of 0.4748337. This result also indicates that the original difference in means we calculated for `views` across `success` groups is *meaningful* and not likely to arise just by chance alone (or by an "unlucky" randomization) under the assumption of the sharp null hypothesis.

2 Sports Cards

2.1 t-test and confidence interval

Using a `t.test`, compute a 95% confidence interval for the difference between the treatment mean and the control mean. After you conduct your test, write a narrative statement, *using inline code evaluation* that describes what your tests find, and how you interpret these results. (You should be able to look into `str(t_test_cards)` to find the pieces that you want to pull to include in your written results.)

```
# this should be the t.test object. Extract pieces from this object
# in-text below the code chunk.

# Run t-test to evaluate difference in means between treatment and control
t_test_cards <- t.test(bid ~ uniform_price_auction, data = d)

# Display components of t-test result
# str(t_test_cards)

# Extract p-value from t-test result
# t_test_cards$p.value

# Extract confidence interval from t-test result
t_test_ci <- t_test_cards$conf.int[1:2]

# Display confidence interval
print(t_test_ci)
```

```
## [1]  3.557141 20.854624
```

Narrative Analysis: The 34 bidders who received the treatment auction format had a *significantly lower* average bid (M_t = 16.6176471) than the 34 bidders who received the control treatment auction format (M_c = 28.8235294), based on the results of the t-test (t(61.9828667) = 2.8211439, p = 0.0064208). The 95% confidence interval for the difference between the control mean and treatment mean is [3.5571406, 20.8546241]. Note that this confidence interval spans strictly *positive* numbers, because the t-test is subtracting the treatment mean (a lower value) from the control mean (a higher value), rather than the other way around.

2.2 Interpretation of confidence interval

In your own words, what does this confidence interval mean? This can be simple language, but it has to be statistically appropriate language.

Answer: The 95% confidence interval for the difference between the control mean and treatment mean is [3.5571406, 20.8546241], which indicates that this interval has a 95% chance of including the actual estimated average treatment effect (ATE). Put another way, if we were to conduct numerous iterations (perhaps thousands!) of this experiment, after which we ran a t-test and computed a confidence interval for the difference in means (just like we did above), we would expect 95% of the confidence intervals we computed to contain this actual estimated ATE. Note that the ATE converted to a t-statistic within our `t_test_cards` object is actually being represented as the control mean minus the treatment mean — that is, the t-test is comparing the mean bid from the lowest-labeled `uniform_price_auction` group (the control group, with label 0) to the mean bid from the highest-labeled `uniform_price_auction` group (the treatment group, with label 1). This is the reverse of how we would typically think of the ATE — treatment group mean minus control group mean — but it yields the same value in *absolute terms* either way.

2.3 Randomization inference, and confidence interval?

Conduct a randomization inference process, with `n_ri_loops = 1000`, using an estimator that you write by hand (i.e. in the same way as earlier questions). On the sharp-null distribution that this process creates,

compute the 2.5% quantile and the 97.5% quantile using the function `quantile` with the appropriate vector passed to the `probs` argument. This is the randomization-based uncertainty that is generated by your design. After you conduct your test, write a narrative statement of your test results.

```
## first, do you work for the randomization inference

# Set number of iterations for randomization inference
n_ri_loops <- 1000

# Create randomization inference function for modular and reusable code
ri <- function(permutations = n_ri_loops) {

  # Create vector to store test-statistic (ATE) under each permutation
  cards_ate_vector <- NA

  # Simulate `n_ri_loops` random permutations of treatment assignment
  for(perm in 1:permutations) {

    # Add computed test-statistics to vector after treatment randomization
    cards_ate_vector[perm] <- d[, .(mean_bids = mean(bid)),
                  keyby = .(sample(uniform_price_auction))][, diff(mean_bids)]

  }

  # Return vector of test-statistics
  return(cards_ate_vector)

}

# Compute mean `bids` across `uniform_price_auction` groups
bids_group_mean <- d[, .(mean_bids = mean(bid)), keyby = uniform_price_auction]

# Compute ATE
cards_ate <- bids_group_mean[uniform_price_auction == 1, mean_bids] -
             bids_group_mean[uniform_price_auction == 0, mean_bids]

# Run randomization inference function with `n_ri_loops` permutations
cards_ri_distribution <- ri(n_ri_loops) # numeric vector of length equal
                                        # to your number of RI permutations

# Compute the 2.5% and 97.5% quantiles
cards_ri_quantiles <- quantile(cards_ri_distribution, probs = c(0.025, 0.975))
# there's a built-in to pull these.

# Compute the p-value associated with obtaining a test-statistic at least as
# extreme as the 2.5% and 97.5% quantile values
cards_ri_p_value <- as.numeric(
  (sum(cards_ri_distribution <= cards_ate) +
   sum(cards_ri_distribution >= (-1 * cards_ate))) /
   length(cards_ri_distribution)
  )
```

Narrative: With 1000 permutations, our randomization inference test reveals the following: (1) like the t-test, we see that the estimated `cards_ate` (-12.2058824) — that is, the difference in mean bids between bidders in the treatment and control groups — is *statistically significant* with a p-value of 0.004, (2) the

lowest 2.5% of ATE outcomes from the distribution of randomized/permuted ATEs (under the assumption of the sharp null hypothesis) is defined as all outcomes at or below -8.4441176, and (3) the highest 2.5% of ATE outcomes from this same distribution is defined as all outcomes at or above 8.0911765. The first of these findings (#1) suggests that there is a very slim probability of seeing the actual estimated ATE based on chance alone (i.e., this ATE would fall within the left-tail of the randomization distribution assuming the sharp null hypothesis). The second and third of these findings (#2/#3) confirms that we're seeing a statistically meaningful ATE in the original data, since the value of the estimated ATE (-12.2058824) falls *below* the 2.5% quantile value (-8.4441176) of the randomization distribution. It also indicates that the 95% confidence interval generated by the randomization inference test is [-8.4441176, 8.0911765], which means that 95% of the time, we could expect the estimates generated from the sharp null distribution to fall within this range. Additionally, it's interesting to note that the size/width of the 95% confidence interval for our randomization inference test (i.e., the arithmetic difference between the interval lower bound at the 2.5% quantile and upper bound at the 97.5% quantile) is virtually the same as the size/width of the confidence interval for our t-test. For our randomization inference test, this width is equal to 16.5352941; for our t-test, this width is equal to 17.2974834. The *locations* of these intervals are different, however, because our randomization inference interval is based on the assumption of the *sharp null hypothesis* and is therefore centered (virtually perfectly) around 0. We could transform our randomization inference confidence interval to look like our t-test confidence interval by just subtracting the `cards_ate` from the interval lower bound and upper bound (*subtracting*, in this case, again because the t-test confidence interval was generated based on comparing the control group mean bid to the treatment group mean bid instead of the other way around). This finding confirms that, when interpreting confidence intervals, "the *location of the interval varies from one experiment to the next*, while the true ATE remains constant", as Green & Gerber observe in *Field Experiments* (p. 67).

2.4 Compare regression and randomization inference

Do you learn anything different if you regress the outcome on a binary treatment variable? To answer this question, regress `bid` on a binary variable equal to 0 for the control auction and 1 for the treatment auction and then calculate the 95% confidence interval using *classical standard errors* (in a moment you will calculate with *robust standard errors*). There are two ways to do this – you can code them by hand; or use a built-in, `confint`. After you conduct your test, write a narrative statement of your test results.

```
# this should be a model object, class = 'lm'.

# Create linear model for `bid` regressed on group assignment variable
mod <- lm(bid ~ uniform_price_auction, data = d)

# Compute 95% confidence interval for the model with classical SEs
mod_ci <- confint(mod, level = 0.95)

# Display confidence interval
print(mod_ci)
```

```
##                            2.5 %    97.5 %
## (Intercept)             22.71534 34.931716
## uniform_price_auction -20.84416 -3.567603
```

Narrative: Our regression model yields very similar results to our t-test and slightly different results than our randomization inference test. The 95% confidence interval generated by our regression model is [-20.844162, -3.5676027], which is virtually the same interval as what we saw for our t-test, but with negative numbers instead of positive numbers due to the order in which the t-test compares group means (i.e., control to treatment). The resulting regression-generated interval has a 95% chance of bracketing the actual estimated ATE, which corroborates our earlier t-test. Meanwhile, this regression-generated confidence interval looks different than the randomization-generated confidence interval (even though both have the same size/width) because it's not based on the sharp null hypothesis. We could transform it to look more like the randomization

inference-generated interval by adding the `cards_ate` value (-12.2058824) to the lower bound and upper bound of the interval, which would center it around 0.

2.5 Regression with robust confidence interval

Calculate the 95% confidence interval using robust standard errors, using the `sandwich` package. There is a function in `lmtest` called `coefci` that can help with this. It is also possible to do this work by hand. After you conduct your test, write a narrative statement of your test results.

```
# this should be a numeric vector of length 2

# Generate 95% confidence interval using robust SEs
cards_robust_ci <- c(
  coefci(mod, vcov = vcovHC(mod))[2],
  coefci(mod, vcov = vcovHC(mod))[4]
)

# Display class for `cards_robust_ci` to check for numeric result
# class(cards_robust_ci)

# Display confidence interval with robust SEs
cards_robust_ci
```

```
## [1] -20.974068  -3.437696
```

Narrative: Since robust standard errors (RSEs) are "robust" against outliers and against the assumption of homoscedasticity in the data (i.e., constant/consistent variance in errors across all values of the independent variable), I would generally expect to see *wider*/more conservative 95% confidence intervals with RSEs applied than with OLS/classical SEs applied. Interestingly, applying RSEs to our regression model yields only a slightly wider 95% confidence interval of [-20.9740683, -3.4376964]. This could mean that the underlying data may meet (or nearly meet) the assumption of homoscedasticity, or that there are few outliers influencing the original confidence interval. Either way, the slightly wider interval here makes sense with RSEs applied.

2.6 Compare and contrast results

Characterize what you learn from each of these different methods – are the results contingent on the method of analysis that you choose?

Answer: As expected, all three statistical tests/methods lead to a similar conclusion about the difference in mean bids across control and treatment groups: there is a slim probability this difference in mean bids (the estimated ATE) across auction formats could arise simply due to chance, making it a statistically significant outcome. This main result/conclusion isn't contingent on the method of analysis chosen, even though p-values may differ slightly between tests/methods. However, our choice of method *can* influence the "location" (i.e., lower bound and upper bound points) of a test-generated 95% confidence interval. For example, the confidence interval generated by our randomization inference test is centered around 0 since we're making an assessment of the likelihood of seeing the estimated ATE against the assumption of the sharp null hypothesis; meanwhile, the confidence intervals generated by our t-test and regression model are not centered the same way since we're not invoking the sharp null hypothesis. This means that the language we use to describe the conclusions drawn from these confidence intervals may vary depending on the method of analysis chosen. Yet, we also know that we can transform each of these intervals or re-center them around a consistent value by adding or subtracting the ATE from the interval lower/upper bounds. This indicates that all three tests tell a similar story about the data and the estimated ATE, but do so from slightly different angles.

3 Power Analysis

3.1 Describe your testing procedure

Describe a t-test based testing procedure that you might conduct for this experiment. What is your null hypothesis, and what would it take for you to reject this null hypothesis? (This second statement could either be in terms of p-values, or critical values.)

Answer: For this experiment and under a t-test based testing procedure, I would posit the following as our null hypothesis, sticking with the classical convention of using a "two-tailed" test (specifying no preconceived direction of effect): *there is no difference in mean bids between the bidders who receive the treatment auction format and the bidders who receive the control auction format.* We would then need to (A) run the experiment and collect data on card bids between participants in each group (control and treatment), (B) subtract the average (mean) of all bids from participants in the treatment group from the average (mean) of all bids from participants in the control group to get a "difference in means" estimate for the ATE, and then (C) run a t-test to determine if this difference in means is likely (probable) to occur under the assumption of the null hypothesis. Also sticking with convention, I would recommend using an alpha value (or significance level) of 0.05 for the t-test. The t-test would convert our observed difference in means to a t-statistic based upon the assumptions underlying a t-distribution, and then it would generate a p-value that signifies the probability of obtaining a value *as extreme or more extreme* as that t-statistic. If the p-value generated by the test turned out to be <= 0.05 (our chosen significance level), we would conclude two things: (1) the difference in mean bids we observed in our data falls within *either the lower 2.5% quantile or upper 2.5% quantile* of the t-distribution, which means it's an outcome that's improbable to observe based on chance alone under the assumption of the null hypothesis, and (2) we should reject the null hypothesis and instead posit that there *is* a statistically significant difference in mean bids between the bidders who receive the treatment auction format and the bidders who receive the control auction format (this constitutes an "alternative" hypothesis).

3.2 Suppose you only had ten subjects, what would you learn

Suppose that you are only able to recruit 10 people to be a part of your experiment – 5 in treatment and another 5 in control. Simulate "re-conducting" the sports card experiment once by sampling from the data you previously collected, and conducting the test that you've written down in part 1 above. *Given the results of this 10 person simulation, would your test reject the null hypothesis?*

```
# this should be a test object

# Create random sample + t-test function for modular and reusable code
rs_t_test <- function(sample_size) {

  # Randomly sample `sample_size/2` control group bids (balanced groups)
  c_sample_dt <- d[uniform_price_auction == 0,
                   .SD[sample(.N, sample_size / 2, replace = TRUE)], ]

  # Randomly sample `sample_size/2` treatment group bids (balanced groups)
  t_sample_dt <- d[uniform_price_auction == 1,
                   .SD[sample(.N, sample_size / 2, replace = TRUE)], ]

  # Combine sampled bids into a single data.table
  sampled_dt <- rbindlist(list(c_sample_dt, t_sample_dt))

  # Run t-test on sampled bids to evaluate control/treatment difference in means
  t_test_sampled <- t.test(bid ~ uniform_price_auction, data = sampled_dt)

  # Return t-test outcome
  return(t_test_sampled)
```

```
}

# Run random sample + t-test function for 10 participants
t_test_ten_people <- rs_t_test(10)

# Display result
t_test_ten_people
```

```
##
##  Welch Two Sample t-test
##
## data:  bid by uniform_price_auction
## t = 0.86712, df = 7.4, p-value = 0.4131
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -9.844831 21.444831
## sample estimates:
## mean in group 0 mean in group 1
##            19.0            13.2
```

Answer: Based on the results of this 10 person simulation *at the time I ran the code*, our t-test would *not* reject the null hypothesis. This test indicates that there is no statistically significant difference between the average bid of the 5 bidders who received the treatment auction format ($M_t = 13.2$) and the average bid of the 5 bidders who received the control auction format ($M_c = 19$), as characterized by its test statistic value and a p-value that is greater than our chosen significance level of 0.05 ($t\,(7.399965) = 0.8671216$, $p = 0.4131076$).

3.3 With only ten subjects, what is your power?

Repeat this process – sampling 10 people from your existing data and conducting the appropriate test – one-thousand times. Each time that you conduct this sample and test, pull the p-value from your t-test and store it in an object for later use. *Consider whether your sampling process should sample with or without replacement.*

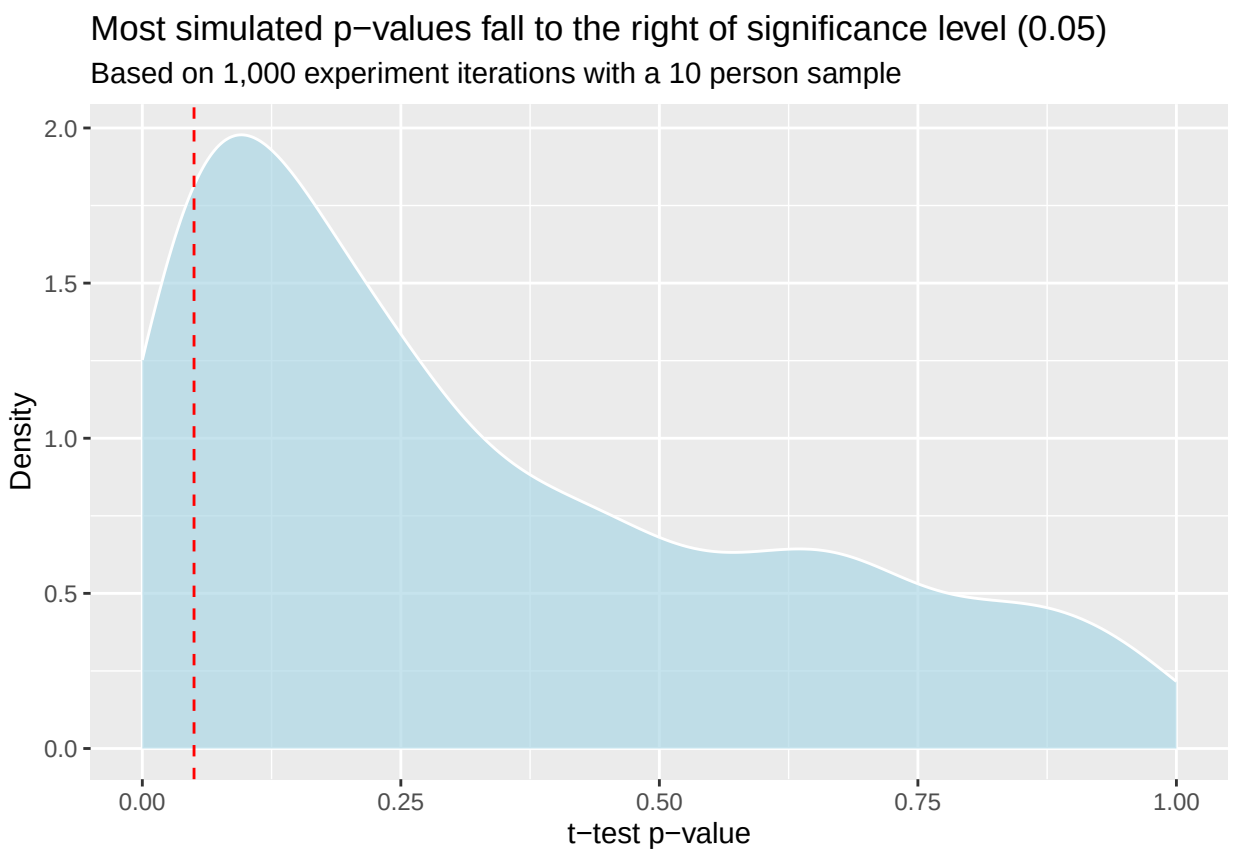
```
# fill this in with the p-values from your power analysis

# Instantiate empty numeric vector of length 1000 to store p-values
t_test_p_values <- numeric(1000)

## you can either write a for loop, use an apply method, or use replicate
## (which is an easy-of-use wrapper to an apply method)

# Run random sample + t-test function 1000 times, storing p-values along the way
for (i in 1:1000) {
  t_test_trial_result <- rs_t_test(10)
  t_test_p_values[i]  <- t_test_trial_result$p.value
}
```

Answer: The vector of p-values from this simulation is notated as `t_test_p_values`. Interestingly, the number of p-values below our chosen significance level of 0.05 is 136, which means only 13.6% of all simulated t-tests generate a statistically significant result.

3.4 Visual analysis

Use `ggplot` and either `geom_hist()` or `geom_density()` to produce a distribution of your p-values, and describe what you see. *What impression does this leave you with about the power of your test?*

```
# Create density plot to show distribution of p-values from simulated t-tests
ggplot(
  data = NULL,
  aes(
    x = t_test_p_values)) +
  geom_density(
    fill = "lightblue",
    color = "white",
    alpha = 0.7) +
  xlim(
    min_value = 0,
    max_value = 1) +
  geom_vline(
    xintercept = 0.05,
    color = "red",
    linetype = "dashed") +
  labs(
    title = "Most simulated p-values fall to the right of significance level (0.05)",
    subtitle = "Based on 1,000 experiment iterations with a 10 person sample",
    x = "t-test p-value",
    y = "Density"
)
```



Answer: Based on the definition of power provided earlier (i.e., "Given a test procedure and data, what proportion of the tests I *could conduct* would reject the null hypothesis?") and based on the distribution of p-values in the chart above, my impression of the power of this 10 person test is that it's quite *low*. In fact, we can calculate it as 0.136. Only a small proportion of tests within our 1000 test simulation result in statistically significant p-values of <= 0.05 that would reject the null hypothesis. Therefore, this test doesn't seem to have a lot of power.

3.5 Interpret your results, given your power

Suppose that you and David were to actually run this experiment and design – sample 10 people, conduct a t-test, and draw a conclusion. **And** suppose that when you get the data back, **lo and behold** it happens to reject the null hypothesis. Given the power that your design possesses, does the result seem reliable? Or, does it seem like it might be a false-positive result?

```
# Answer provided in narrative form below
```

Answer: If this scenario *were* to happen, color me surprised! Given the power that our design possesses with just a 10 person sample, this result wouldn't seem reliable to me. Our test's power indicates that we could expect this outcome to arise a mere 13.6% of the time, which doesn't feel like great odds. If we were to suspend reality for a moment and say that, in this scenario, I hadn't already worked through question 2 on this problem set and calculated an estimate for the ATE from the FULL cards data and established its statistical significance using a t-test/randomization inference test/regression model (!!!), then I *would* consider this outcome to represent a false-positive result.

3.6 Conduct a power analysis

Apply the decision rule that you wrote down in part 1 above to each of the simulations you have conducted. What proportion of your simulations have rejected your null hypothesis? This is the power that this design and testing procedure generates. After you write and execute your code, include a narrative sentence or two about what you see.

```
# Compute power of 1000 simulation test based on number of p-values <= 0.05
t_test_rejects <- sum(t_test_p_values <= 0.05) / 1000

# Display power
t_test_rejects
```

```
## [1] 0.136
```

Answer: As stated above, the power of this design is 0.136. This means that only 13.6% of our simulations would reject the null hypothesis. This level of power is quite small and makes it difficult to actually justify this design (i.e., using only a 10 person sample) because it leaves a lot of room for error. More specifically, it makes it likely that we'll commit a Type II error and fail to reject the null hypothesis when we actually *should* reject it.

3.7 Moar power!

Does buying more sample increase the power of your test? Apply the algorithm you have just written onto different sizes of data. Namely, conduct the exact same process that you have for 10 people, but now conduct the process for every 10% of recruitment size of the original data: Conduct a power analysis with a 10%, 20%, 30%, ... 200% sample of the original data. (You could be more granular if you like, perhaps running this task for every 1% of the data).

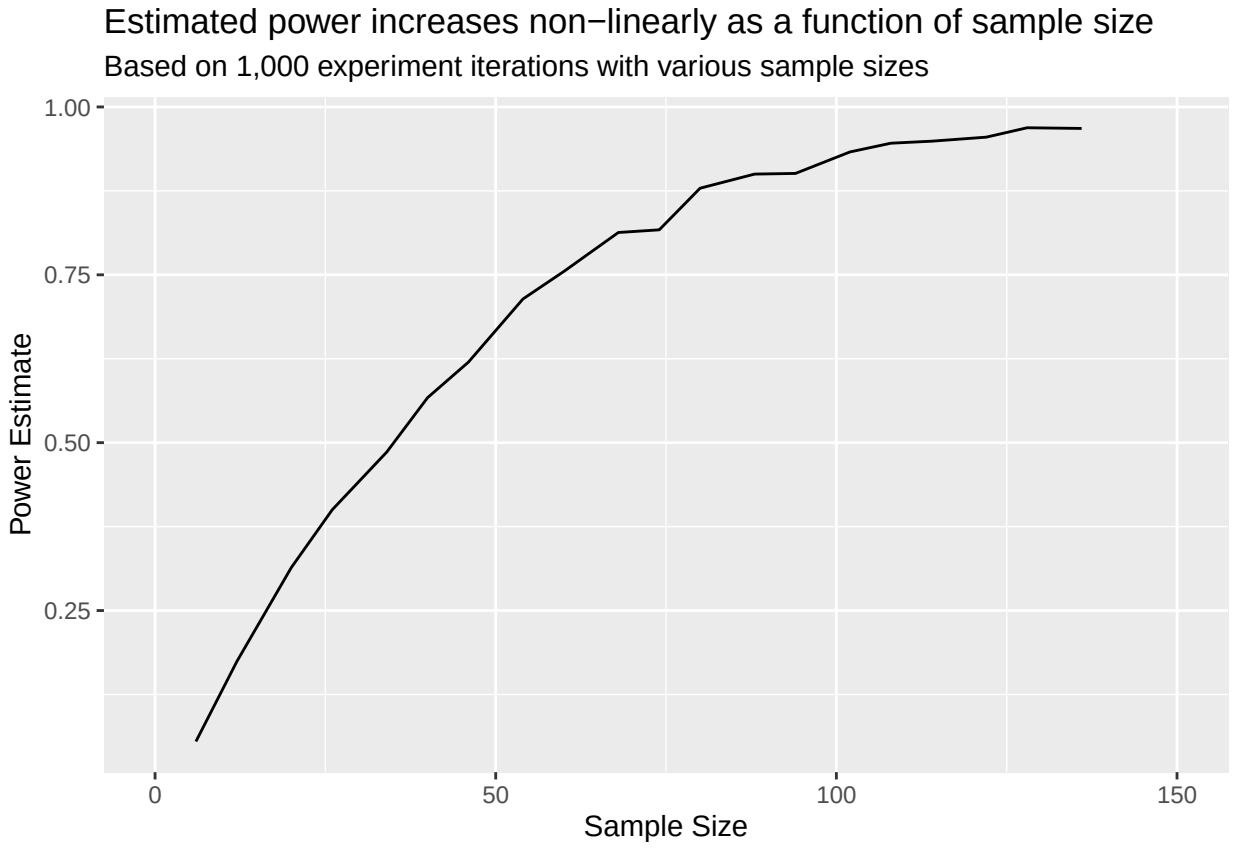
```
# Create vector of uneven sample sizes ranging from 10% - 200% of original data
sampling_trials_uneven <- floor(seq(0.10, 2, by = 0.10) * nrow(d))

# Create function to round value down to nearest whole number and make even
```

```r
floor_to_even <- function(n) {

  # Store `result` value as rounded down number that's divisible by 2
  result <- floor(n)
  if (result %% 2 != 0) {
    result <- result - 1
  }

  # Return rounded even value
  return(result)
}

# Instantiate new vector to contain rounded even sampling trials (sizes)
sampling_trials_even <- numeric(0)

# Transform uneven sampling trials to rounded even values and append to vector
for (i in sampling_trials_uneven) {
  sampling_trials_even <- c(sampling_trials_even, floor_to_even(i))
}

# Create power evaluator function to assess sample size impacts on power
percentages_to_sample <- function(samples) {

  # Instantiate new vector to hold power estimates/proportion of t-test rejects
  t_test_rejects <- numeric(0)

  # Loop through sampling trials (e.g., 10%, 20%, ... , 200% of original data)
  for (j in sampling_trials_even) {

    # Instantiate vector of length 1000 to hold p-values
    t_test_p_values <- numeric(1000)

    # Run random sample + t-test function 1000 times, storing each p-value
    for (k in 1:1000) {
      t_test_trial_result <- rs_t_test(j)
      t_test_p_values[k]  <- t_test_trial_result$p.value
      }

    # Compute power of 1000 simulation test based on number of p-values <= 0.05
    t_test_rejects <- c(t_test_rejects, sum(t_test_p_values <= 0.05) / 1000)

  }

  # Return vector of power estimates (proportion of t-test rejects)
  return(t_test_rejects)

}

# Run power evaluator function for a vector of 20 different sampling schemes,
# ranging from 10% to 200% of original data, with step size 10%
power_estimates <- percentages_to_sample(sampling_trials_even)

# Display estimates of power across all sampling trials
```

```
power_estimates
```

```
##  [1] 0.055 0.174 0.314 0.400 0.486 0.567 0.620 0.714 0.755 0.813 0.817 0.879
## [13] 0.900 0.901 0.933 0.946 0.949 0.955 0.969 0.968
```

```
# Create line plot to show relationship between power and sample size
ggplot(
  data = NULL,
  aes(
    x = sampling_trials_even,
    y = power_estimates)
  ) +
  geom_line(
    color = "black",
  ) +
  xlim(
    min_value = 0,
    max_value = 150) +
  labs(
    title = "Estimated power increases non-linearly as a function of sample size",
    subtitle = "Based on 1,000 experiment iterations with various sample sizes",
    x = "Sample Size",
    y = "Power Estimate"
)
```



Answer: Yes, buying more sample *does* increase the power of our test — though not linearly, as both the vector of power estimates (`power_estimates`) and line chart showing power vs. sample size indicate. The relationship we're seeing between power and sample size across the 20 different simulations of the experiment

mirrors what Green & Gerber outline in a formal statistical formula for power on p. 93 of *Field Experiments*: our power estimates generally increase at a rate consistent with the square-root of the sample size. This means that increasing sample size is indeed one way to boost power, but it has diminishing returns (e.g., moving from a sample size of 10 to 20 provides much more impact to power than moving from a sample size of 50 to 100). Therefore, it should be considered and evaluated alongside other methods for increasing power, like reducing variance or increasing effect size/treatment dosage.