Problem Set 2

Alex, Micah, Scott, and David

12/01/2021

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.

```
# Under the sharp null Y_0 and Y_1 are the same
d_sharp <- d[ ,  .(Y_0 = views, Y_1 = views, tau = views - views, success)]
rbind(head(d_sharp,5), tail(d_sharp,5))
```

```
##       Y_0   Y_1   tau success
##     <int> <int> <int>   <int>
##  1:     2     2     0       0
##  2:     1     1     0       0
##  3:     0     0     0       0
##  4:     5     5     0       0
##  5:     3     3     0       0
##  6:     9     9     0       1
##  7:     0     0     0       1
##  8:     2     2     0       1
##  9:     2     2     0       1
## 10:     4     4     0       1
```

Answer: ... Under the sharp null hypothesis $Y_i(1) = Y_i(0)$, meaning there's no treatment effect ($\tau_i = 0$), for all subjects (see example table above). In this specific case it would mean that the views toward members of other countries won't change whether the respondent successfully attended the Hajj or not.

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
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)`
hajj_ate        <- hajj_group_mean[ , diff(mean_views)]
hajj_ate
```

```
## [1] 0.4748337
```

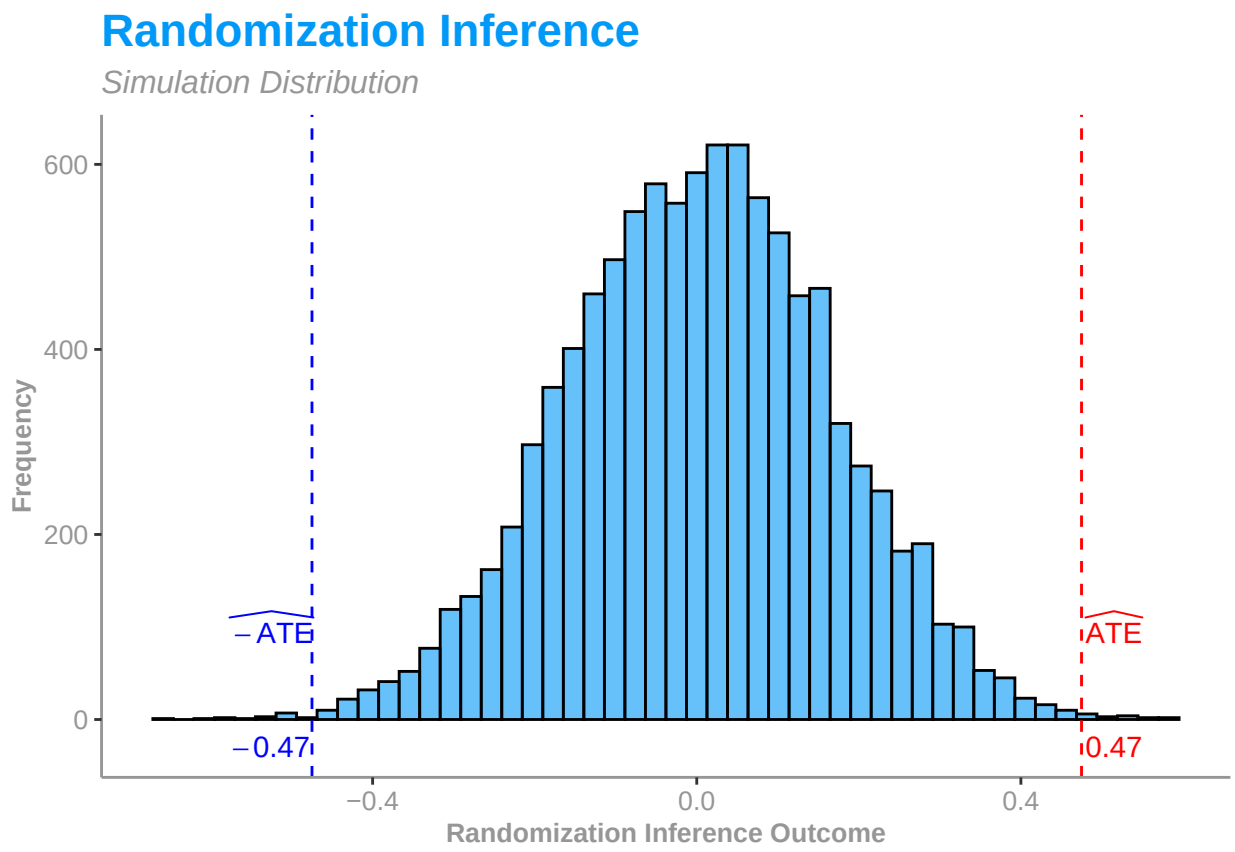
```
class(hajj_ate)
```

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

Answer: ... By taking a look at the data we find that Hajj attenders' (winners) views toward others seem more positive than among non-winners (by about 0.47 scale points).

```
## 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
## of RI permutations you ran
ri_simulation <- function(simulations = 10000){
  vec_ate <- NA
  for(sim in 1:simulations) {
    vec_ate[sim] <- d[ , .(mean_views = mean(views)),
                    keyby = .(sample(success))][ , diff(mean_views)]
  }
  return(vec_ate)
}

hajj_ri_distribution <- ri_simulation()

sim_dist_hist <-  ggplot() +
  geom_histogram(
    aes(hajj_ri_distribution),
    fill = "#0099F8",
    color="black",
    bins = 50,
    alpha = 0.6) +
  geom_vline(xintercept = hajj_ate, linetype = "dashed", color = "red") +
  geom_vline(xintercept = -hajj_ate, linetype = "dashed", color = "blue") +
  annotate(geom='text', x=hajj_ate + 0.04, y=100,
           label= TeX("$\\widehat{ATE}", output='character'),
           parse=TRUE, color= "red", size = 4) +
  annotate(geom='text', x=-hajj_ate - 0.05, y=100,
           label= TeX("$\\widehat{-ATE}", output='character'),
           parse=TRUE, color= "blue", size = 4) +
  annotate(geom='text', x=hajj_ate + 0.04, y=-30,
           label= round(hajj_ate,2),
           parse=TRUE, color= "red", size = 4) +
  annotate(geom='text', x=-hajj_ate - 0.05, y=-30,
           label= round(-hajj_ate,2),
           parse=TRUE, color= "blue", size = 4) +
  labs(
    title = "Randomization Inference",
    subtitle = "Simulation Distribution",
    x = "Randomization Inference Outcome",
    y = "Frequency"
  ) +
  theme_classic() +
  theme(
    plot.title = element_text(color = "#0099F8",
                              size = 17,
                              face = "bold"),
    plot.subtitle = element_text(color="#969696",
                                 size = 12,
                                 face = "italic"),
```

```
    axis.title = element_text(color = "#969696",
                              size = 10,
                              face = "bold"),
    axis.text = element_text(color = "#969696", size = 10),
    axis.line = element_line(color = "#969696")
  )

sim_dist_hist
```



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.

```
# length 1 numeric vector from comparison of `hajj_ate` and `hajj_ri_distribution`
hajj_count_larger <- sum(hajj_ri_distribution > hajj_ate)
```

Answer: ... Out of the 10,000 simulated values assuming the sharp null hypothesis, only 14 of them are larger than the ATE computed from the actual data.

1.4 Randomization inference: one-sided p-value

If there are `hajj_count_larger` (14) 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
hajj_one_tailed_p_value <- hajj_count_larger / length(hajj_ri_distribution)
```

Answer: ... This would represent a one-tailed p-value $p = 0.0014$, which basically means that, under the sharp null-hypothesis, finding an outcome larger than the one we have would only happen $p = 0.14\%$ of the times.

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
hajj_two_tailed_p_value <- sum(
  abs(
    hajj_ri_distribution
    ) > hajj_ate
  ) / length(hajj_ri_distribution)
```

Answer: ... By computing a two-sided p-value we are basically trying to find values that are "more extreme" than the magnitude of the one we've found, no matter the direction. Which translates into the proportion of randomization inference loops on the left side of the blue line plus the ones on the right side of the red line on the graph above. In this case it means a two-tailed p-value $p = 0.0031$.

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.
t_test_cards <- t.test(bid ~ uniform_price_auction, data=d)
t_test_cards$conf.int
```

```
## [1]  3.557141 20.854624
## attr(,"conf.level")
## [1] 0.95
```

Narrative Analysis: ... The computed interval is (3.5571, 20.8546), and it assumes that it contains the true ATE at a 95% confidence level.

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: ... As we said the interval is supposed to contain the true ATE at a 95% confidence level. This doesn't mean that there's a 95% chance that the true value is between these two specific values, it only means that if we compute confidence intervals several times using the same statistical procedures, these intervals will contain the true population's value 95% of the times.

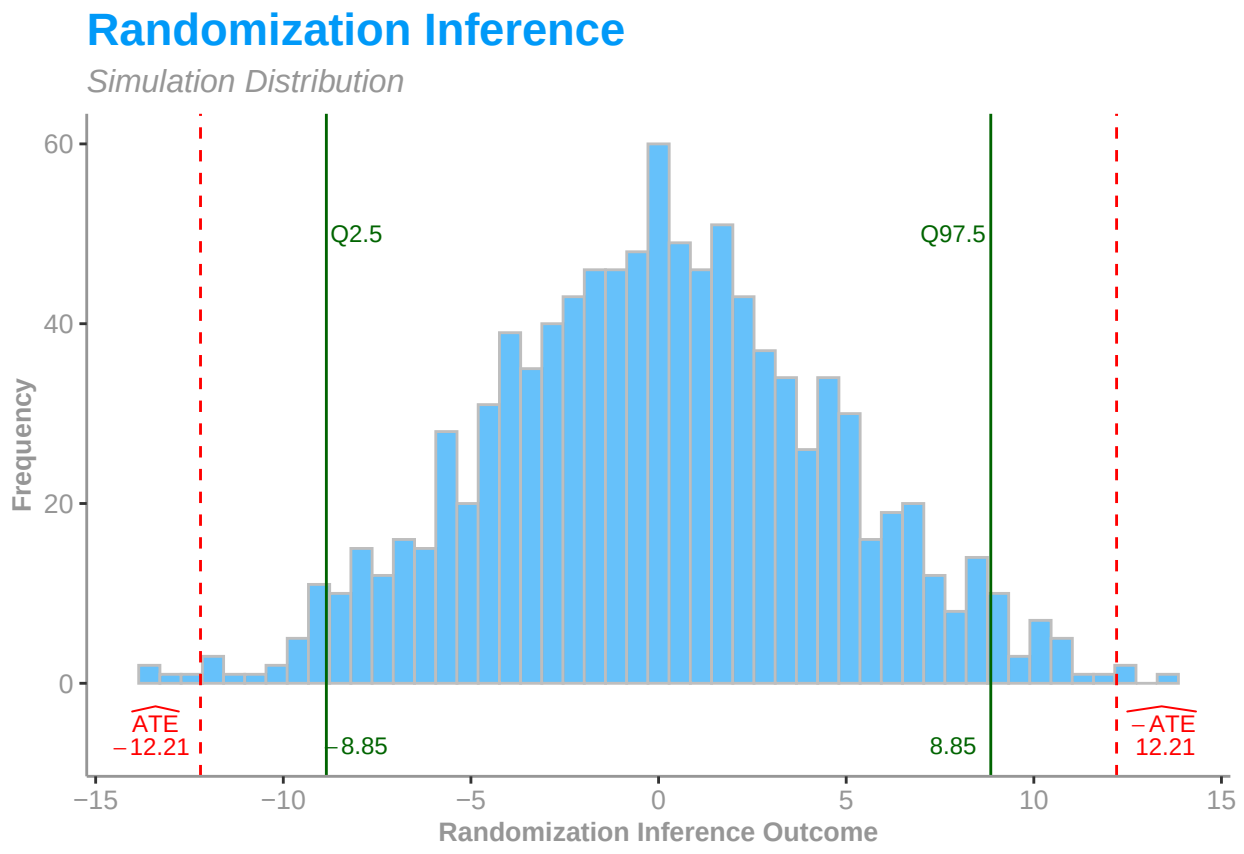
2.3 Randomization inference, and confidence interval?

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
n_ri_loops <- 1000

cards_ate             <- d[ , .(mean_bid= mean(bid)),
                            keyby = .(uniform_price_auction)][ , diff(mean_bid)]

cards_ri_simulation <- function(simulations = 1000){
  vec_ate <- NA
  for(sim in 1:simulations) {
    vec_ate[sim] <- d[ , .(mean_bid= mean(bid)),
                   keyby = .(sample(uniform_price_auction))][ , diff(mean_bid)]
  }
```

```
  return(vec_ate)
}

cards_ri_distribution <- cards_ri_simulation() # numeric vector of length equal
                                               # to your number of RI permutations
cards_ri_quantiles    <- quantile(
  cards_ri_distribution,
  probs = c(0.025,0.975)) # there's a built-in to pull these.

cards_ri_p_value      <- sum(
  abs(
    cards_ri_distribution
    ) > abs(cards_ate)
  ) / length(cards_ri_distribution)

card_sim_hist <-  ggplot() +
  geom_histogram(
    aes(cards_ri_distribution),
    fill = "#0099F8",
    color="gray",
    bins = 50,
    alpha = 0.6) +
  geom_vline(xintercept = -cards_ate, linetype = "dashed", color = "red") +
  geom_vline(xintercept = cards_ate, linetype = "dashed", color = "red") +
  geom_vline(xintercept = cards_ri_quantiles[1], color = "darkgreen") +
  geom_vline(xintercept = cards_ri_quantiles[2], color = "darkgreen") +
  annotate(geom='text', x= -cards_ate + 1.2, y=-4,
           label= TeX("$\\widehat{-ATE}", output='character'),
           parse=TRUE, color= "red", size = 3) +
  annotate(geom='text', x= cards_ate - 1.2, y=-4,
           label= TeX("$\\widehat{ATE}", output='character'),
           parse=TRUE, color= "red", size = 3) +
  annotate(geom='text', x= c(-cards_ate + 1.3, cards_ate - 1.3),
           y= -7, label= c(round(-cards_ate,2), round(cards_ate,2)),
           parse=TRUE, color= "red", size = 3) +
  annotate(geom='text',
           x= c(cards_ri_quantiles[1] + 0.8, x=cards_ri_quantiles[2] - 1),
           y = 50, label= c("Q2.5", "Q97.5"),
           parse=TRUE, color= "darkgreen", size = 3) +
  annotate(geom='text',
           x= c(cards_ri_quantiles[1] + 0.8, x=cards_ri_quantiles[2] - 1),
           y = -7,
           label= c(
             round(cards_ri_quantiles[1],2),
             round(cards_ri_quantiles[2],2)),
           parse=TRUE, color= "darkgreen", size = 3) +
  labs(
    title = "Randomization Inference",
    subtitle = "Simulation Distribution",
    x = "Randomization Inference Outcome",
    y = "Frequency"
  ) +
  theme_classic() +
```

```
  theme(
    plot.title = element_text(color = "#0099F8",
                              size = 17,
                              face = "bold"),
    plot.subtitle = element_text(color="#969696",
                                 size = 12,
                                 face = "italic"),
    axis.title = element_text(color = "#969696",
                              size = 10,
                              face = "bold"),
    axis.text = element_text(color = "#969696", size = 10),
    axis.line = element_line(color = "#969696")
  )

card_sim_hist
```



Narrative: ... Under the sharp null we find that quartiles 2.5% and 97.5% (summing up to 0.05 or 5% of the distribution in both tails) are in the positions -8.85 and 8.85 respectively, therefore our estimated $\widehat{ATE} = -12.21$ is more extreme on either side. Hence, the two-tailed p-value $p = 0.007 < 0.05$, rejecting the null hypothesis of no treatment effect.

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'.
mod <- lm(bid ~ uniform_price_auction, data = d)
cards_ci <- confint(mod, "uniform_price_auction", level = 0.95)
```

Narrative: ... Using classic standard errors our 95% confidence interval would be (-20.8442, -3.5676) for the regression method. Which seems similar to what we obtained using the t.test, although in this one we also have the sign (which explains the side to which the treatment effect is going), and not only the magnitude of the difference.

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
cards_robust_ci <- coefci(mod,
                          parm = "uniform_price_auction",
                          vcov. = vcovHC(mod),
                          level = 0.95
                          )
```

Narrative: ... Finally when using robust standard errors the mentioned 95% confidence interval becomes (-20.9741, -3.4377), a range a little bit wider but ensuring at least the 95% we're looking for.

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: ... Using any method the result seems clear and consistent, there seems to be a negative and statistically significant treatment effect. But when adding robust standard errors, the CI gets a bit wider so variance in our effect might not be constant across values, although the difference doesn't seem too perceptible, so in this specific case, any chosen method will do the work relatively well

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: ... Since the sample is small, it might be tempting to increase the significance level (α) to find statistically significant results more easily, or if we are believing that the treatment effect goes in a specific direction we could use a one-tailed test. Although, I think we should stick to the standard two-tailed t.test for the difference in means between the two groups using $\alpha = 5\%$, because we don't really know the direction we should expect from the effect, and moving the significance level in a so small sample could easily lead to false positives.

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?*

```
alpha <- 0.05
d <- fread('../data/list_data_2019.csv')

sampling_people <- function(data, ppl_per_group = 5, repl = FALSE){
  samp <- data.table(
    'bid' = c(
      data[uniform_price_auction == 0 , sample(bid, ppl_per_group, replace = repl)],
      data[uniform_price_auction == 1 , sample(bid, ppl_per_group, replace = repl)]
      ),
    'treatment' = rep(c(0,1), each = ppl_per_group)
  )
  return(samp)
}

ten_people_sample <- sampling_people(d, 5)

t_test_ten_people <- t.test(bid ~ treatment,
                            data=ten_people_sample)

ten_people_pval <- t_test_ten_people$p.value

t_test_ten_people
```

```
##
##  Welch Two Sample t-test
##
## data:  bid by treatment
## t = 1.4498, df = 6.372, p-value = 0.1945
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -11.29135  45.29135
```

```
## sample estimates:
## mean in group 0 mean in group 1
##              39              22
```

Answer: ... With this specific sample we'd get a p-value of 0.1945 which would mean that, using $\alpha = 5\%$, we'd fail to reject our null hypothesis of no difference between treatment and control.

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
t_test_p_values <- rep(NA, 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)
t_test_p_values <- replicate(
  1000,
  t.test(
    bid ~ treatment,
    data=sampling_people(d, 5, repl = FALSE)
    )$p.value
  )
```

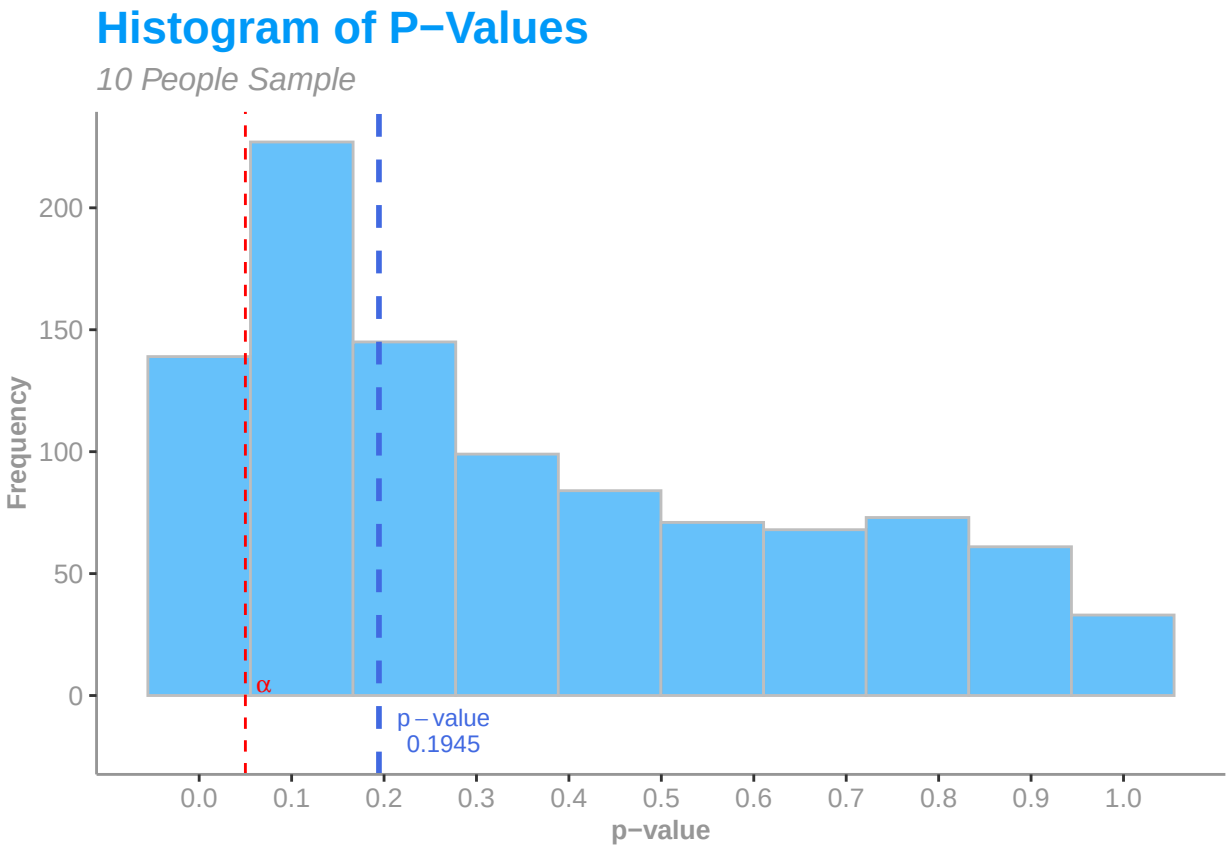
Answer: ... Since we're only using 10 people out of the 68 in the existing data, I think that it would be better not to replace people back into the sampled data, so we don't duplicate people when we have more than enough to sample from, even when results might not differ too much.

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?*

```
pval_hist <-  ggplot() +
  geom_histogram(
    aes(t_test_p_values),
    fill = "#0099F8",
    color="gray",
    bins = 10,
    alpha = 0.6) +
  geom_vline(xintercept = ten_people_pval,
             linetype = "dashed",
             color = "royalblue",
             size = 1) +
  geom_vline(xintercept = alpha,
             linetype = "dashed",
             color = "red") +
  annotate(geom='text', x= c(ten_people_pval + .07, ten_people_pval + .07),
           y= c(-10,-20), label= c("p-value", round(ten_people_pval,5)),
```

```
          parse=TRUE, color= "royalblue", size = 3) +
  annotate(geom='text', x= alpha + 0.02, y= 4,
           label= TeX("$\\alpha", output='character'),
           parse=TRUE, color= "red", size = 3) +
  labs(
    title = "Histogram of P-Values",
    subtitle = "10 People Sample",
    x = "p-value", y = "Frequency"
  ) +
  theme_classic() +
  theme(
    plot.title = element_text(color = "#0099F8", size = 17, face = "bold"),
    plot.subtitle = element_text(color="#969696", size = 12, face = "italic"),
    axis.title = element_text(color = "#969696", size = 10, face = "bold"),
    axis.text = element_text(color = "#969696", size = 10),
    axis.line = element_line(color = "#969696")
  ) +
  scale_x_continuous(breaks=seq(0, 1, 0.1))

pval_hist
```



Answer: ... This doesn't seem like a study with much power. Since we have many samples with p-values on the right side of our threshold (0.05), there doesn't seem to be too much chance to actually reject the null (even when we know from the previous section that there's a significant effect).

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: ... This result doesn't seem reliable at all, by looking at the graph and the chance to reject the null it might well be a false positive.

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.

```
t_test_rejects <- sum(t_test_p_values < alpha) / length(t_test_p_values)
```

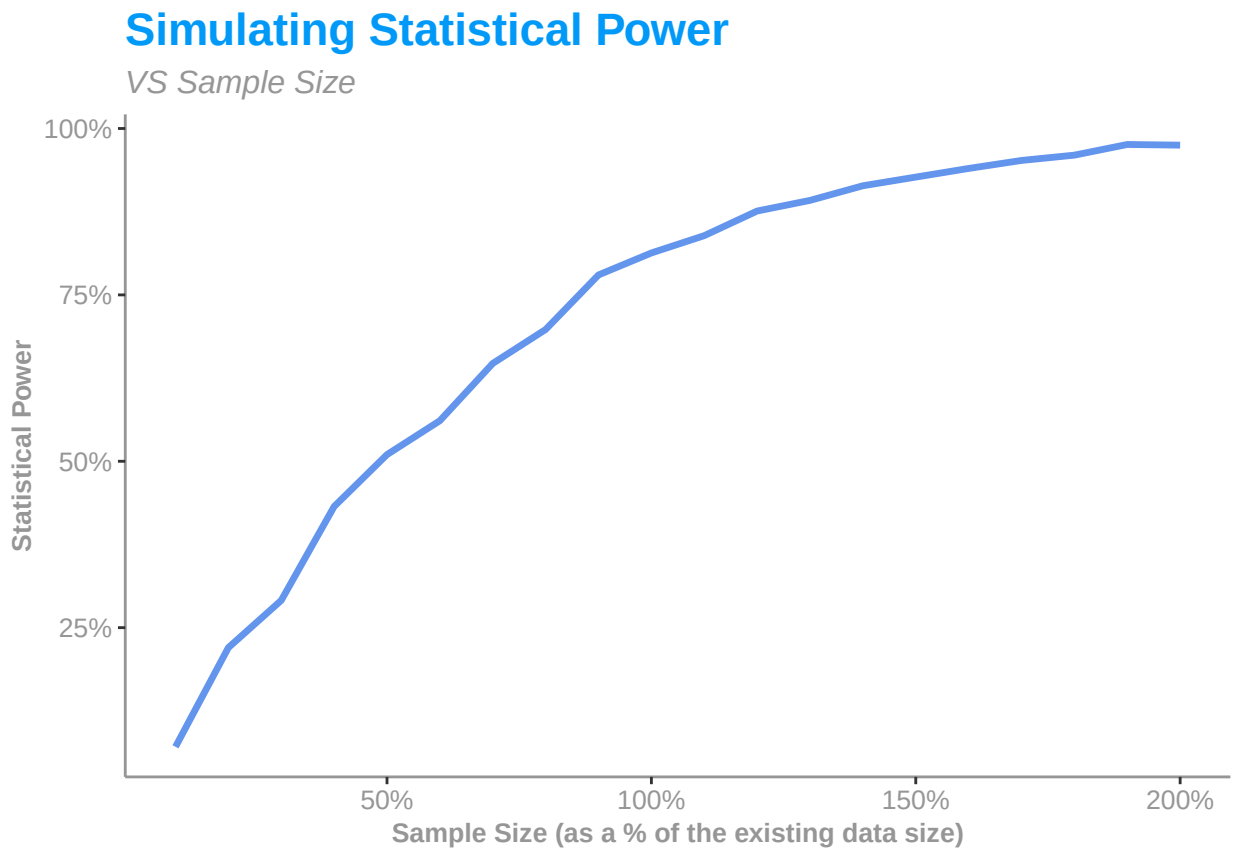
Answer: ... The proportion of simulations that would have rejected the null hypothesis meaning that $p_{val} < \alpha_{0.05}$ would be 12.4%. This seems quite low, any result we get will make us doubt of the veracity in it. So it seems like, if we want to be confident in the results, we need to improve the power of our study.

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).

```
# Every 10%
percentages_to_sample <- seq(0.1, 2, 0.1)

power <- rep(NA, length(percentages_to_sample))

for(i in 1:length(percentages_to_sample)) {
  # dimension of the sample
  sample_dim <- 2 * round(dim(d)[1] * percentages_to_sample[i]/2)

  p_values_t <- replicate(
    1000,
    t.test(
      bid ~ treatment,
      data = sampling_people(d, sample_dim/2, repl = TRUE)
      )$p.value
    )

  power[i] <- sum(p_values_t < alpha) / length(p_values_t)
  }

pow_vs_size <- data.table(
```

```
  'size' = percentages_to_sample,
  'power' = power
)

pow_plot <- ggplot(
  data = pow_vs_size,
  aes(x = percentages_to_sample,
      y = power, group = 1
      )
  ) +
  geom_line(color="cornflowerblue", size=1.2) +
  scale_x_continuous(labels = scales::percent) +
  scale_y_continuous(labels = scales::percent) +
  labs(
    title = "Simulating Statistical Power",
    subtitle = "VS Sample Size",
    x = "Sample Size (as a % of the existing data size)",
    y = "Statistical Power"
  ) +
  theme_classic() +
  theme(
    plot.title = element_text(color = "#0099F8",
                              size = 17,
                              face = "bold"),
    plot.subtitle = element_text(color="#969696",
                                 size = 12,
                                 face = "italic"),
    axis.title = element_text(color = "#969696",
                              size = 10,
                              face = "bold"),
    axis.text = element_text(color = "#969696", size = 10),
    axis.line = element_line(color = "#969696")
  )

pow_plot
```



Answer: ... We can see that as we increase the size of our sample, power gets higher, although it doesn't grow linearly, so we might consider having an optimal sample size, high enough to have enough power to be confident in our results, but not too big since we won't be obtaining too much of a difference while we might be consuming a lot more resources. *NOTE: Since we're increasing the size of the sample even to a 200% of the actual data size, this time we're allowing for replacement in our sampling, pretending we're getting these results from a bigger population with similar values to the ones we found in the existing data (if we didn't do this, as soon as we hit 100% of the existing data size, we'd also have 100% of statistical power).*