table 1 cold reproduction

```r
#| include: false
#| warning: false
#| message: false
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(knitr)
library(numDeriv)
library(purrr)
library(rlang)
```

```
##
## Attaching package: 'rlang'

## The following objects are masked from 'package:purrr':
##
##     %@%, as_function, flatten, flatten_chr, flatten_dbl, flatten_int,
##     flatten_lgl, flatten_raw, invoke, list_along, modify, prepend,
##     splice
```

ATE function

```r
ate <- function(outcome, treatment){

## first get rid of non-missing observations
treatment <- treatment[!is.na(outcome)]
outcome <- outcome[!is.na(outcome)]

## then add treated and untreated vectors
treated <- outcome[treatment == 1]
not_treated <- outcome[treatment == 0]

## the number of observations in each category
N1 <- length(na.omit(treated))
N0 <- length(na.omit(not_treated))

## Calculate the difference in means
mean_treated <- mean(treated)
mean_control <- mean(not_treated)
ate <- mean_treated - mean_control

## Standard Error Calculation
## Use the usual Neyman Approximation
var1 <- sum((treated - mean(treated))^2) / (N1 - 1)
var0 <- sum((not_treated - mean(not_treated))^2) / (N0 - 1)

## Take sqrt to get SE
se_ate <- sqrt(var1/N1 + var0/N0)

## Degree of Freedom calculation
df_numerator <- (var1/N1 + var0/N0)^2
df_denominator <- (var1^2/(N1^2*(N1-1))) + (var0^2/(N0^2*(N0-1)))

degrees_free <- df_numerator/df_denominator

## Now plug it in with the inverse
## cdf of the t-distribution qt(x, df)

lwr_bound <- ate - qt(.975, degrees_free)*se_ate
upp_bound <- ate + qt(.975, degrees_free)*se_ate

## give the output for the ATE
return(c(ATE = ate,
         lwr_bound = lwr_bound,
         upp_bound = upp_bound))
}
```

Replicating Line 1 of Table 1

```r
## this is only the data for the mturk sample
load(file = "df_mtg.rda")

## Overall
overall <- df_mtg |>
  filter(C == "Experiment") |>
  select(dv_pca_metoo, treatment) |>
  mutate(treatment = if_else(treatment == "Control",
                             0, 1))

## Men
men <- df_mtg |>
```

```r
  filter(C == "Experiment", gender == 0) |>
  select(dv_pca_metoo, treatment) |>
  mutate(treatment = if_else(treatment == "Control",
                             0, 1))

## women
women <- df_mtg |>
  filter(C == "Experiment", gender == 1) |>
  select(dv_pca_metoo, treatment) |>
  mutate(treatment = if_else(treatment == "Control",
                             0, 1))

## Republicans
reps <- df_mtg |>
  filter(C == "Experiment",
         Partisanship == "Republicans") |>
  select(dv_pca_metoo, treatment) |>
  mutate(treatment = if_else(treatment == "Control",
                             0, 1))


## Democrats
dems <- df_mtg |>
  filter(C == "Experiment",
         Partisanship == "Democrats") |>
  select(dv_pca_metoo, treatment) |>
  mutate(treatment = if_else(treatment == "Control",
                             0, 1))
```

Loop over the data frames to get the Table row 1

```r
ATEs <- matrix(nrow = 5, ncol = 3)
colnames(ATEs) <- c("ATE", "lwr_bound", "upp_bound")
dfs <- list(overall, men, women, reps, dems)


for(i in 1:length(dfs)){
  tmp <- round(ate(dfs[[i]]$dv_pca_metoo,
              dfs[[i]]$treatment),2)
  ATEs[i,1] <- tmp["ATE"]
  ATEs[i,2] <- tmp["lwr_bound"]
  ATEs[i,3] <- tmp["upp_bound"]
}

kable(cbind(Treatments = c("Overall", "Men",
                      "Women", "Republicans",
                      "Democrats"), ATEs) |>
  as.data.frame() |>
  mutate(across(.cols = ATE:upp_bound,as.numeric))
)
```

Treatments	ATE	lwr_bound	upp_bound
Overall	0.22	0.03	0.41
Men	0.03	-0.27	0.33
Women	0.41	0.18	0.64
Republicans	0.29	-0.10	0.68
Democrats	0.16	-0.05	0.36

This fully reproduces the first row of the table.

ACTEs

Here's the formula for the ACTEs from the article.

$$ACTE = \frac{E[Y|D=1] - E[Y|D=0, T=T_C]}{\alpha}$$

According to the article, this is a difference between a weighted average of the outcome in the selection condition $E[Y|D=1]$ and the average of those in the experimental control condition $E[Y|D=0, T=T_C]$ divided by the proportion of people who select into treatment.The avoid ACTE is the same logic except dividing by $1-\alpha$ to reflect the proportion avoiding treatment. Standard errors are constructed via the delta method because we will eventually need a ratio estimator.

Estimates and Variances of each term

We have weighted averages to calculate, leading to a first question. How do we get a weighted average. What's the variance of a weighted sample? Define $W = \sum_i^n w_i$ and $\bar{x} = \frac{1}{W}\sum_i^n w_i x_i$ which is the weighted average. Now, to get the variance.

$$V[\bar{x}] = V[\frac{1}{W}\sum_i^n w_i x_i]$$

$$V[\bar{x}] = \sum_i^n V[\frac{w_i}{W}x_i]$$

$$V[\bar{x}] = \sum_i^n \left(\frac{w_i}{W}\right)^2 V[x_i]$$

$$V[\bar{x}] = V[x]\sum_i^n \left(\frac{w_i}{W}\right)^2$$

Line 1 is the definition of variance. Line 2 is a property of the sum operator. Line 3 is an application of the fact that $V[aX] = a^2V[X]$. Line 4 is due to the fact that sum of variances is just $V[x]$ which is a constant and can be pulled out of the expectation. So to for $\frac{1}{W^2}$ for the same reason. Note that if all the weights are equal, the weights fraction reduces to $\frac{1}{N}$

We'll start with Line 2 of Table 1, which is the ACTE of the selecting treatment respondents, and just work through the first estimate and confidence interval for the "Overall" group. First we're going to need a weighted variance function.

```r
weighted.var <- function(x, w, na.rm = TRUE) {
  # https://seismo.berkeley.edu/~kirchner/Toolkits/Toolkit_12.pdf
    if (na.rm) {
      i <- !is.na(x)
        w <- w[i]
        x <- x[i]
    }
    sum.w <- sum(w)
    sum.w2 <- sum(w^2)
    mean.w <- sum(x * w) / sum(w)
    part1 <- (sum(w*x^2, na.rm = T)/sum.w)-mean.w^2
    part2 <- sum.w2/(sum.w^2 - sum.w2)
    part1*part2
}
```

The next part is to get the appropriate numbers of respondents who fit into different categories.

```r
## just putting this here for a different data
acte <- df_mtg
```

Step 1: Get the Weights

We get the sums of the number of respondents who were in the choice experiment, who avoided the treatment, who selected into the treatment, and who were in the control condition. From there we create the set of weights.

```r
Ns <- acte |>
  mutate(n_c = if_else(C == "Choice",1,0),
         n_avoid = if_else(C == "Choice"&
                             avoid01 ==1,1,0),
         n_select = if_else(C == "Choice"&
                              avoid01 == 0, 1,0),
         n_control = if_else(D_ch == "Control",1,0),
         ## Deal with the NAs to not worry about them
         ## in the next step
         n_control = if_else(is.na(n_control),
                             0, n_control)) |>
  summarise(
    n_c = sum(n_c),
    n_avoid = sum(n_avoid),
    n_select = sum(n_select),
    n_control = sum(n_control))


weight <- Ns |>
  summarise(select_weights = 1/(n_select/n_c),
            avoid_weights = 1/(n_control/n_avoid))
```

Step 2: Get Weighted Means and SEs

The individuals who didn’t avoid the treatment and received control receive the selection weights. The ones who did avoid the treatment receive the avoid weights.

```r
acte_weights <- acte |>
  mutate(weights = case_when(
    C=="Choice" & avoid01 == 0 & D_ch == "Control"~ weight$select_weights,
    C=="Choice" & avoid01 == 1 & D_ch == "Control"~weight$avoid_weights,
    TRUE ~ 1)
  )

### Get weighted means and weighted SEs
weighted_means_sds <- acte_weights |>
  filter(!is.na(treatment),
         !is.na(dv_pca_metoo)) |>
  group_by(treatment) |>
  summarise(avg = sum(weights*dv_pca_metoo,
                      na.rm = T)/sum(weights, na.rm =T),
            se = weighted.var(dv_pca_metoo,
                              weights, na.rm = T))

kable(weighted_means_sds)
```

treatment	avg	se
Control	-0.1489988	0.0051259
Selection	0.0042718	0.0018183
Treatment	0.0723203	0.0045346

Step 3: Get the Proportions selecting into treatment

All of that work gets the first part of that expression. Now we need the second part, the proportion selecting treatment. This is actually straightforward because it's just the mean value of Y given the experiment arm, which here is the Choice.

```r
## Get the first part of the expression and its SE
## This is just the mean and se for the select01 group
y_select <- acte |>
  filter(C == "Choice") |>
  summarise(avg = mean(select01),
            se = sd(select01)/sqrt(n()))

kable(y_select)
```

avg	se
0.8139205	0.0146779

Step 4: Delta Method

```r
avg_x <- weighted_means_sds %>%
  filter(treatment != "Treatment")%>%
  summarise(diff = sum(abs(avg)))%>%
  pull()
x_se <- weighted_means_sds %>%
  filter(treatment != "Treatment")%>%
  summarise(se = sqrt(sum(se)))%>%
  pull()

kable(c(xbar = avg_x, xse = x_se))
```

	x
xbar	0.1532705
xse	0.0833322

```
estimate <- avg_x /y_select$avg
kable(round(estimate,2))
```

x
0.19

That corresponds to the estimate of the ACTE for the Overall group in Line 2.

Next, we need to get the standard errors via the delta method, which is equivalent to the square root of the variance of the distribution.

Let's go in order. The gradient is just the vector of all the partial derivatives of $g(x, y)$ with respect to each variable.

$$\nabla g(\beta) = \begin{bmatrix} \frac{1}{y} \\ \frac{-x}{y^2} \end{bmatrix}$$

Evaluate each of those points at our estimates we have

$$\nabla g(b) = \begin{bmatrix} \frac{1}{.813} \approx 1.23 \\ \frac{-.1532}{.813^2} \approx -0.23 \end{bmatrix}$$

The variance covariance matrix here is just the variances of x and y on the diagonals because due to the randomization the covariance terms are 0.

$$\sum_b = \begin{bmatrix} \sigma_x^2 \approx .07 & 0 \\ 0 & \sigma_y^2 \approx -0.0002 \end{bmatrix}$$

Now multiply out

$$\begin{aligned} V[b] &= \nabla g(b)' \sum_b \nabla g(b) \\ V[b] &= \begin{bmatrix} 1.23 & -.23 \end{bmatrix}' \begin{bmatrix} .07 & 0 \\ 0 & -0.0002 \end{bmatrix} \begin{bmatrix} 1.23 \\ -0.23 \end{bmatrix} \\ V[b] &= .010 \\ SE[b] &= \sqrt{(}V[b]) \\ SE[b] &= .102 \end{aligned}$$

```r
## function that we seek to estimate
## pass this to the jacobian function
## which calculates the gradient
ratio <- function(x){
  x[1]/x[2]
}

## Variance covariance matrix
vcov <- diag(c(x_se,y_select$se)^2)

## this function by default will output this in
## transpose form, so take the transpose to match
## discussion above
grad_g <- t(numDeriv::jacobian(func = ratio, c(avg_x, y_select$avg)))

se_b <- sqrt(t(grad_g) %*% vcov %*%grad_g)

lwr_b = estimate - 1.96*se_b
upp_b = estimate + 1.96*se_b

kable(round(c(estimate = estimate, lwr_b = lwr_b, upp_b = upp_b),2))
```

	x
estimate	0.19
lwr_b	-0.01
upp_b	0.39

This gives us the results for ACTE select overall.

Now we do this for men, women, republicans, and democrats

Let's make a function for the overall process, that way we can just plug in the subsetted data each time.

```r
## Men
men <- df_mtg |>
  filter(gender == 0) |>
  select(dv_pca_metoo, treatment) |>
  mutate(treatment = if_else(treatment == "Control",
                             0, 1))

## women
women <- df_mtg |>
  filter(gender == 1) |>
  select(dv_pca_metoo, treatment) |>
  mutate(treatment = if_else(treatment == "Control",
                             0, 1))

## Republicans
reps <- df_mtg |>
  filter(Partisanship == "Republicans") |>
  select(dv_pca_metoo, treatment) |>
  mutate(treatment = if_else(treatment == "Control",
                             0, 1))


## Democrats
dems <- df_mtg |>
  filter(Partisanship == "Democrats") |>
  select(dv_pca_metoo, treatment) |>
  mutate(treatment = if_else(treatment == "Control",
                             0, 1))
```

Create the function