Sampling Strategy/Donors

Anna Calle

5/2/2020

Measurement Goal & Measure

Goal: Assess the level of satisfaction of donors with the program.

"Overall, how satisfied or dissatisfied are you with the restoration work done by the Freshwater Trust?"

a. Very Dissatisfied

b. Somewhat dissatisfied

c. Neither satisfied nor dissatisfied

d. Somewhat Satisfied

e. Very Satisfied

Source & Rationale & Unit

Source: New survey. Donor contact information will be provided by the Fundraising Department of the Freshwater Trust

Rationale: Tests assumption that outreach activities result in increased funding for restoration projects. Practical way of gathering information that donors might not feel comfortable disclosing in a non-confidential way. It is useful in deciding if the outreach activities (or the program) need to be adjusted to increase funding. Online surveys are a low cost option for gathering data

Unit: individual survey respondent

Responsibility & Frequency

Responsibility: Independent survey firm to ensure confidentiality and credibility of results

Frequency: Annual surveys will allow enough time for changes in outreach activities (or the program) to be made and for donors to respond to these changes. Sending too many surveys could reduce response rate and quality of responses.

Declaring the population

There is no publicly available data describing the Freshwater Trust program donor population. For this sampling procedure it was assumed that 60% of the donor population are recurring donors and 40% one-time donors. We assumed recurring donors are more likely to be satisfied with the program than one-time donors.

Target Population & Challenges

Target Population: Donors

Challenge of drawing a representative sample: Donors satisfied with the program are more likely to respond than unsatisfied donors, creating a response bias

Sampling procedure: Stratified sampling of one-time donors and recurring donors.

Declaration of hypothetical population

```
population <- declare_population(fabricate(
  N = 200,
  recurring = draw_binary(prob = 0.6, N), # 60% of donors are recurring donors
  satisfaction = draw_ordered(
    x = rnorm(N, mean = 3.5 + 0.5 * recurring),
    # centered at 4 (satisfied) for recurring and at 3.5 (break between neutral and satisfied) for one-
    breaks = c(1.5, 2.5, 3.5, 4.5),
    break_labels = c(1:5)), #  interval representation
  sat_num = as.numeric(satisfaction) # changed satisfaction to numeric
  )
)

pop <- population()

# population table (1 = Very Dissatisfied, 2 = Somewhat dissatisfied, 3 = Neither satisfied nor dissati
pop_table <- kable(table(pop$satisfaction, pop$recurring), row.names = T) %>% add_header_above(c("Satis

pop_table
```

Satisfaction	Recurring	
	0	1
1	3	0
2	12	7
3	28	28
4	25	43
5	9	45

```
# we are interested in the mean satisfaction
my_estimand <- declare_estimands(mean(sat_num),
                                 label = "Ybar")
```

Declaration of reporting probabilities

```
# assign probability of responding to survey (50% for recurring, 30% for one-time)
reporting <- declare_assignment(blocks=recurring,
                  assignment_variable = "R",
                  block_prob=c(0.3,0.5)) # order here?
```

Declaration of sampling procedure

```
# sample 20 recurring donors and 40 one-time donors
sampling <- declare_sampling(strata=recurring,
                             strata_n=c(20, 40))
```

Declaration of estimator

```
strata_weighted_mean <- function(data){
  data.frame(
  estimator_label = "strata_w_mean",
  estimand_label = "Ybar",
  n = nrow(data),
```

```
  stringsAsFactors = FALSE,

  estimate = data %>% filter(R==1) %>%
    group_by(recurring) %>%
    summarise(mean=mean(sat_num)) %>%
    mutate(prop=c(0.5,0.5)) %>%
    mutate(sub.mean=mean*prop) %>% pull(sub.mean) %>%
    sum())
}
```

Diagnosis

```
answer <- declare_estimator(
  handler = tidy_estimator(strata_weighted_mean), # weighted mean
  estimand = my_estimand) # mean

design <- population + my_estimand + reporting +
          sampling + answer
diagnosis <- diagnose_design(design, sims = 100) # simulate design 100 times

diagnosis$diagnosands_df[,c(4,5,12,14)] %>%
  kable()
```

bias	se(bias)	mean_estimate	sd_estimate
-0.0232471	0.0255973	3.715453	0.2501021

```
# As expected, mean is between 3.5 and 4
```