

Session 9: Measurement Practicum Example

Mark Buntaine

WWF Strategic Plan for Nepal 2017-2021 Theory of Change

Measurement Goal & Measure

Goal: Estimate the numbers of tigers remaining in the Terai Arc landscape.

Source & Rationale & Unit

Source: WWF team can set up camera traps.

Rationale: The most cost-effective and efficient way to estimate the population count for a dangerous nocturnal species is through the use of camera traps.

Unit: Parcels of land that are $XX \text{ mi}^2$

Responsibility & Frequency

Responsibility: Manta Trust will directly hire survey firm, but will coordinate with the PFDA to expand the monitoring in the future.

Frequency: The intention is to track changes in attitudes over time. Annual surveys will allow tracking and matches the time period over which it is feasible for attitudes to change.

Declaring the population

Using administrative data, I found the approximate number of vessel visits per month that land tuna.

Landing site	Approximate vessels per month
General Santos	700
Navotas	670
Iloilo	150
Lucena	140
Zamboanga	40
Davao	30
Sual	8

Target Population & Challenges

Target Population: fishing vessel captains who use purse seine techniques

Challenge of drawing a representative sample: sampling effort must be done in landing site clusters, but clusters are of unequal size. It is unclear how often captains return to the same port over the course of the year.

Sampling procedure: Cluster-based sampling at ports with unequal effort according to potential sample size.

DeclareDesign()

```
set.seed(232)
population <- declare_population(
  port = add_level(N=7,
    baseline=c(0.6,0.6,0.4,0.4,0.1,0.1,0.1)),
  captain = add_level(N=c(700,670,150,140,40,30,8),
    know=draw_binary(baseline))
)
pop <- population()
pop.vector <- c(700,670,150,140,40,30,8)

my_estimand <- declare_estimands(mean(know),
  label = "Ybar")
```


DeclareDesign()

```
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(port) %>%  
      summarise(mean=mean(know)) %>%  
      mutate(prop=pop.vector/sum(pop.vector)) %>%  
      mutate(sub.mean=mean*prop) %>% pull(sub.mean) %>%  
      sum()  
  } #just use this function, custom
```

DeclareDesign()

```
answer <- declare_estimator(  
  handler = tidy_estimator(strata_weighted_mean),  
  estimand = my_estimand)  
  
design <- population + my_estimand + reporting +  
  sampling + answer  
diagnosis <- diagnose_design(design, sims = 1000)  
  
diagnosis$diagnosands_df[,c(4,5,12,14)] %>%  
  kable()
```

bias	se(bias)	mean_estimate	sd_estimate
6.63e-05	0.0011201	0.5446641	0.0354159