Session 8: Sampling Techniques

Mark Buntaine

Outline & Goals

1. Quick review of sampling bias
2. Stratified sampling & re-weighting
3. Clustered sampling

Sampling distribution

- **Sampling distribution**: the distribution of sample values with a repeated draw of a given sampling frame.
- *Standard deviation* of a sample describes the variance in the data ($\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2}$)
- *Standard error* of a sample describes the sampling variance of a parameter over repeated draws

Sampling bias

In practice, it is often difficult to take a random sample from our target population, which leads to sampling bias.

- **Sampling bias** is the difference between the true value of the population parameter we are trying to discover and the *expected value* of that parameter based on the sampling procedure.
  - Sampling bias is **not** the difference between the true value of the population parameter and the realized value in a sample.
  - Sampling procedures that deviate from a random sample cause sampling bias.
- There are two main sources of sampling bias:
  - Population / sample mismatches
  - Reporting bias

Main Road Bias Example

Declaring a population: an example

```r
set.seed(228)
population <- declare_population(
  households = add_level(N=500,
     main=sample(c(rep(0,250),rep(1,250))),
     satisfied=correlate(given = main, rho = 0.5,
                         draw_binary, prob = 0.5)
))
pop <- population()

kable(table(pop$main,pop$satisfied)) %>%
  add_header_above(c("main"=1,"satisfied"=2))
```

main	satisfied	
	0	1
0	173	77
1	76	174

Response bias

Response bias is the difference between the true parameter of interest and the expected sample value of the parameter based on unequal probabilities of reporting.

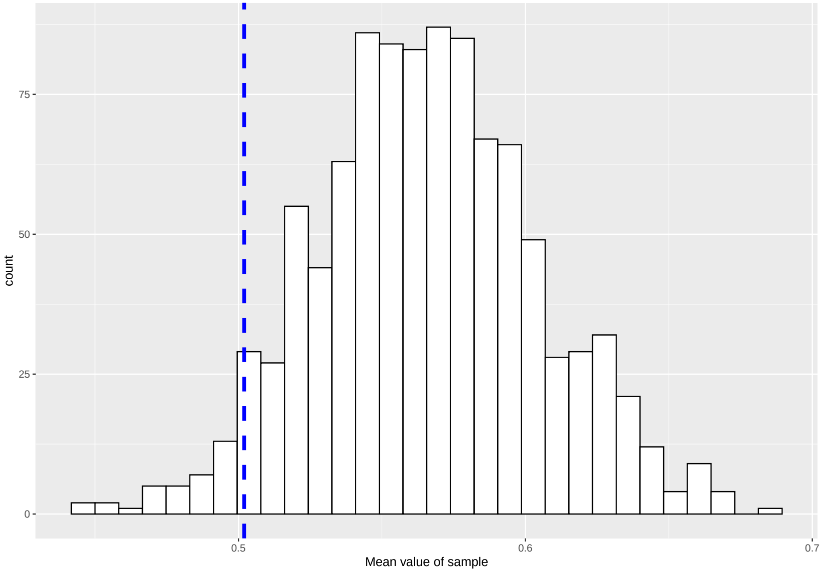
Let's continue with last session's example:

- For main street residents, the chance of being home is 50%
- For main street residents, the chance of being home is 20%

Declaring response bias

```
##
##          0    1
##     0  200   50
##     1  125  125
```

Examining sample characteristics



Sample Weights

Bias in the above example comes from the over-inclusion of main street residents as compared to side street residents. Let's divide them into two groups:

Strata Weights

Stratification: the division of an observed sample or sample frame into non-overlapping groups.

One way to recover the population parameter value would be to compute the weighted average of the strata values:

$$\bar{Y} = \sum^{j} \bar{y}_j w_j$$

Where $\bar{y}$ is the target population parameter, $\bar{y}_j$ is the sample average in strata j, and w_j is the proportion of the population in strata j.

- In Salkind, the equivalent formula is used: $\bar{Y} = \frac{1}{N} \sum_{j=1}^{j} N_j \bar{y}_j$

Strata Weights, Analytical Solution

Using this formula:

$$\bar{Y} = \sum^{j} \bar{y}_j w_j$$

```
prop.table(table(pop$main,pop$satisfied),1)
```

```
##
##            0     1
##    0 0.692 0.308
##    1 0.304 0.696
```

We plug in the relevant values:

$$\bar{Y} = 0.316 * 0.5 + 0.652 * 0.5 = 0.484$$

Strata Weights, Analytical Solution

$$\bar{Y} = 0.316 * 0.5 + 0.652 * 0.5 = 0.484$$

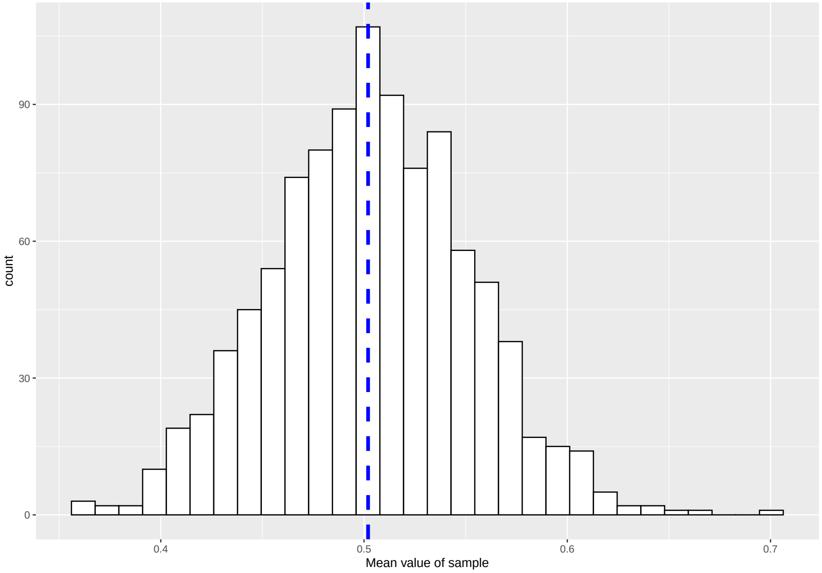
```r
mean(pop$satisfied)
```

```
## [1] 0.502
```

Strata Weights, Sampling Distribution Code

```
sims <- 1000 #simulations
sam.n <- 250 #attempted sample size
store <- rep(NA, sims)

for (i in 1:sims){
  index <- sample(1:500,sam.n) #drawn sample
  pop <- reporting(pop)
  main <- mean(pop[index,] %>%
               filter(R==1 & main==1) %>%
               pull(satisfied))
  side <- mean(pop[index,] %>%
               filter(R==1 & main==0) %>%
               pull(satisfied))
  store[i] <- main * 0.5 + side * 0.5

}
```

Strata Weights, Sampling Distribution



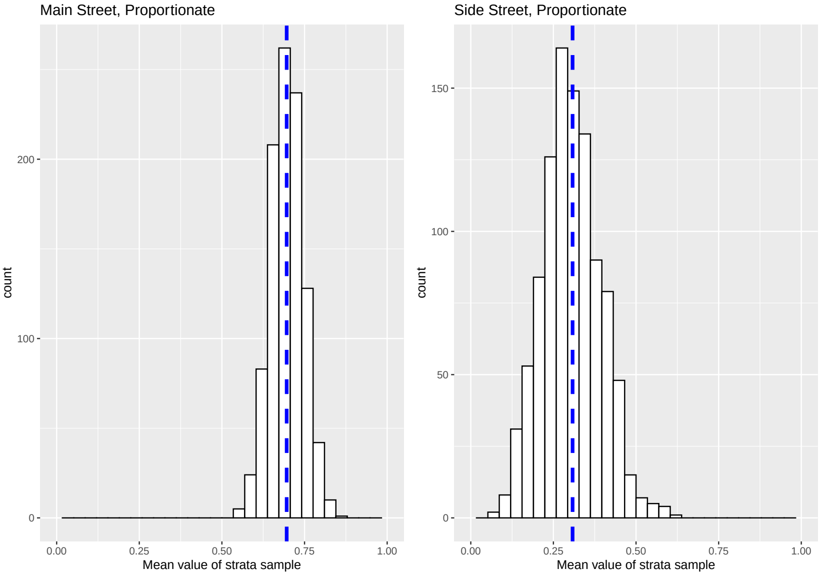
Strata Weights, Assumptions

1. Different responses rates are entirely captured by the strata
    - i.e., missingness is at random within strata
2. The distribution of the population into strata is known

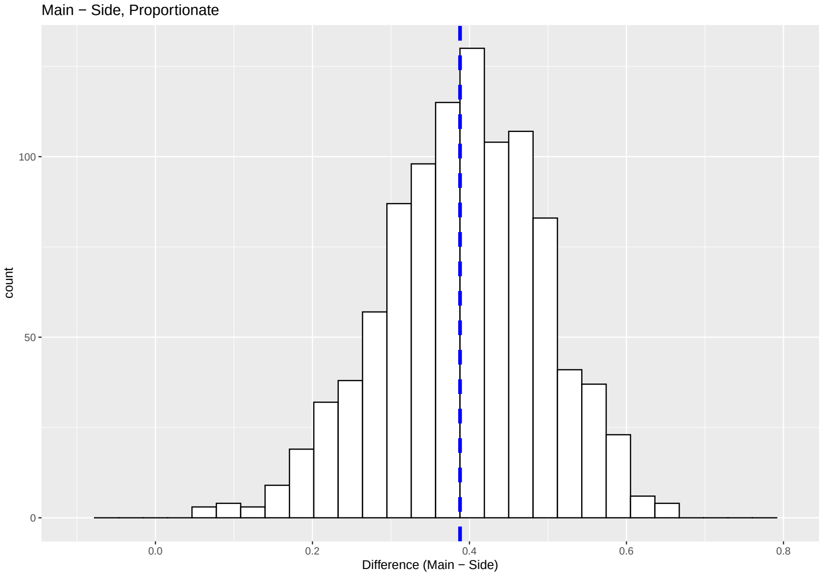
Note: we have not assumed any advanced knowledge about response rates within strata and have still recovered the population parameter

Within-strata descriptive inference

In many situations, we are interested in strata parameters:



Difference between strata



Disproportionate Stratification

We are not required to sample all strata at equal intensity.

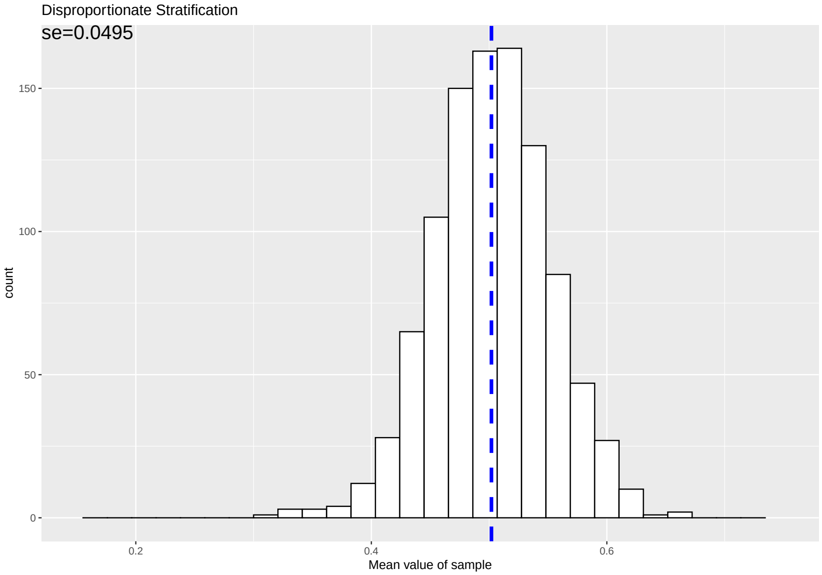
```
+ Main: n=75
+ Side: n=175
```

```
main.index <- which(pop$main==1)
side.index <- which(pop$main==0)

sam <- c(sample(main.index,75),
         sample(side.index,175))
```

Disproportionate Stratification

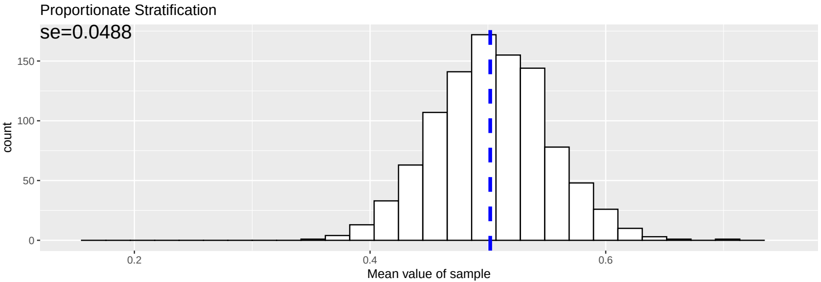
```r
sims <- 1000 #simulations
store <- rep(NA, sims)

for (i in 1:sims){
  sam <- c(sample(main.index,75),
           sample(side.index,175)) #drawn sample
  pop <- reporting(pop)
  main <- mean(pop[sam,] %>%
               filter(R==1 & main==1) %>%
               pull(satisfied))
  side <- mean(pop[sam,] %>%
               filter(R==1 & main==0) %>%
               pull(satisfied))
  store[i] <- main * 0.5 + side * 0.5

}
```

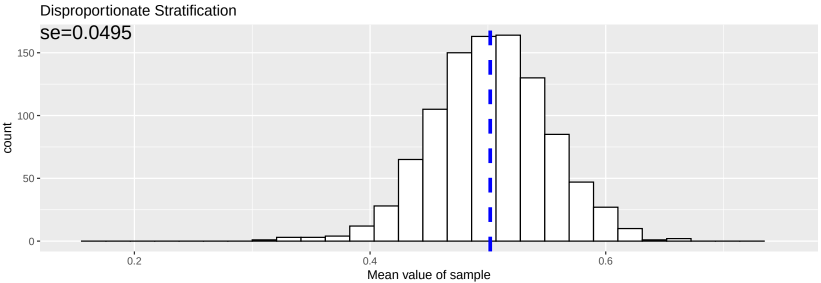
Disproportionate Stratification



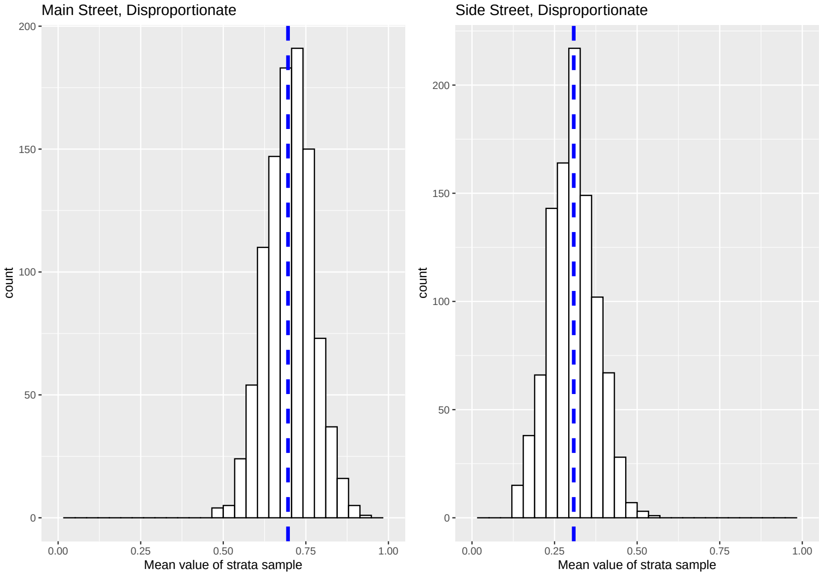
Disproportionate Stratification, Sampling Variation

We do not add much sampling variance!

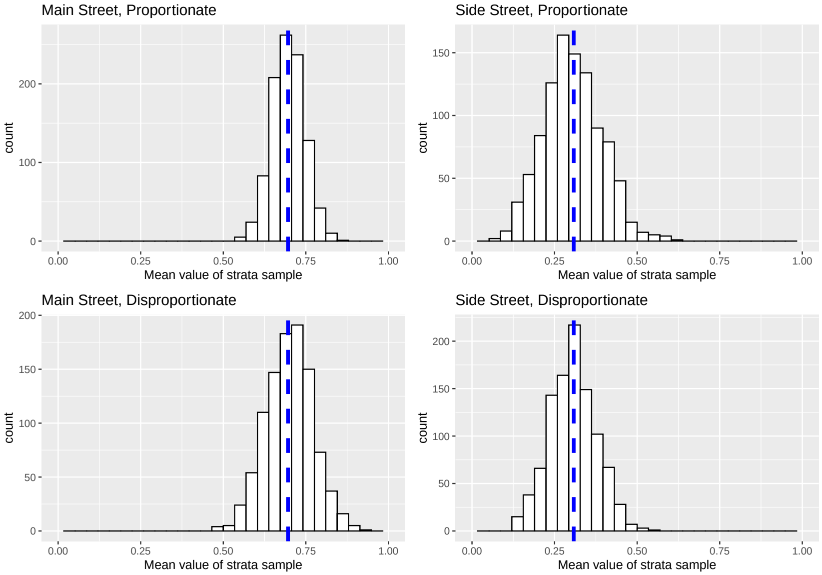




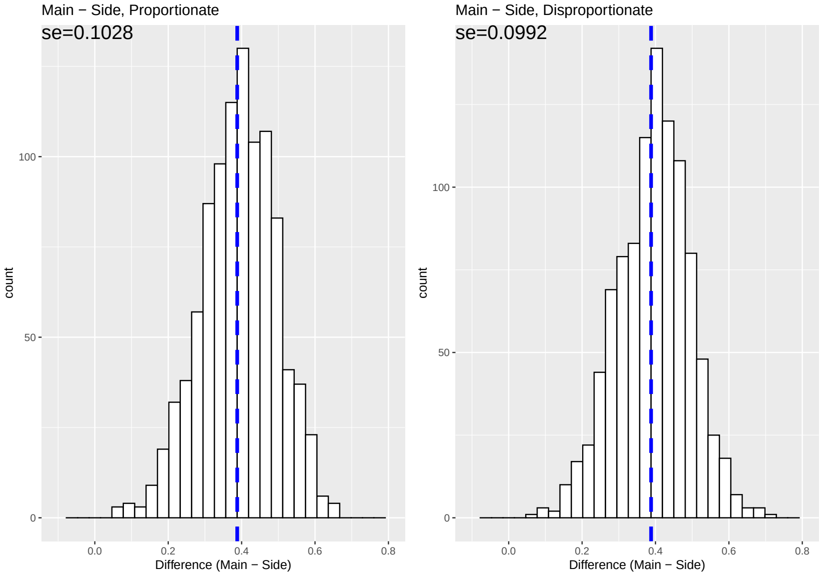
Within-strata sampling variance, disproportionate sampling



Proportionate vs. disproportionate stratified sampling



Sampling distribution of difference between strata



Conceptual practice: stratification

- Describe a monitoring situation where you might want to use stratified sampling
  - What are the strata?
  - How would you allocate sampling effort across the strata?

DeclareDesign()

```
set.seed(228)
population <- declare_population(
  households = add_level(N=500,
     main=draw_binary(N=N, prob = 0.5),
     satisfied=correlate(given = main, rho = 0.5,
                         draw_binary, prob = 0.5)
))

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

DeclareDesign()

```
reporting <- declare_assignment(blocks=main,
                 assignment_variable = "R",
                 block_prob=c(0.2,0.5))

sampling <- declare_sampling(strata=main,
                             strata_n=c(175,75))
```

DeclareDesign()

```r
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(main) %>%
    summarise(mean=mean(satisfied)) %>%
    mutate(prop=c(0.5,0.5)) %>%
    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
0.0015043	0.0013906	0.5025683	0.0564495

Clustered sampling

- Sometimes it might be logistically difficult to sample at the level of *units* and we instead want to sample at the level of *clusters*. Examples:
  - students vs. classrooms
  - households vs. neighborhoods
  - volunteers vs. volunteer teams
  - employees vs. branches
- We can still recover a population parameter by randomly sampling clusters
  - (assuming responses are missing at random within clusters)
- However, we pay a cost in terms of sampling variance when units within clusters are similar
  - i.e., we draw a large number of similar units into the final sample

Example: How well do agents serve the rural poor in India?

```
population <- declare_population(
  district = add_level(N=3,
    u = runif(N, min=0.3, max=0.7)),
  office = add_level(N=30,
    v = runif(length(office), min=-0.1, max=0.1)),
  agent = add_level(N=5,
    w=runif(length(agent), min=-0.3, max=0.3)),
  shg = add_level(N=10,
    x=runif(length(shg), min=-0.1, max=0.1)),
  individual = add_level(N=20,
    y=runif(length(individual), min=-0.3, max=0.3),
    prob=case_when(u+v+w+x+y<0 ~ 0,
                   u+v+w+x+y>1 ~ 1,
          u+v+w+x+y>=0 & u+v+w+x+y<=1 ~ u+v+w+x+y),
     satisfied=draw_binary(prob = prob))
  )
```

Comparing sampling distributions

Let's compare what happens when we sample 5000 people in three ways:

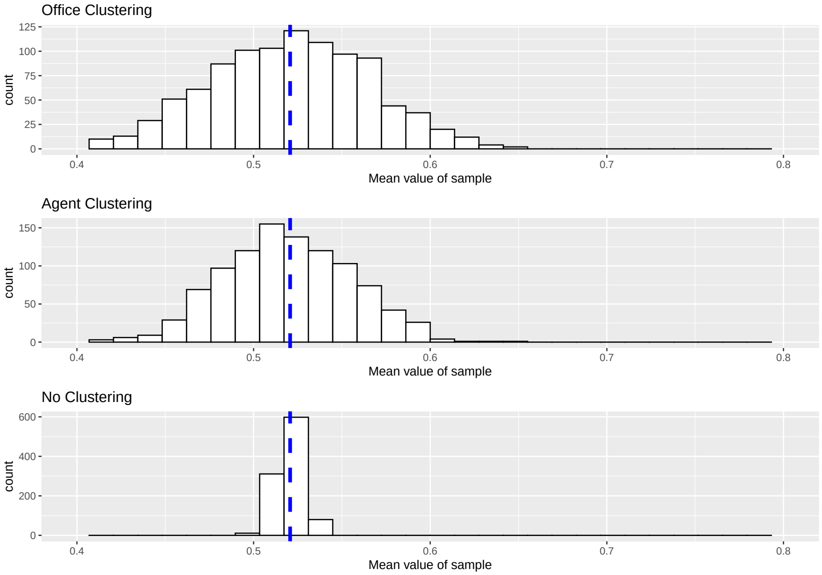
- Sample 5 offices
- Sample 25 agents
- Sample 5000 individuals

```
pop <- population()
```

Three clustered sampling designs

```
sims <- 1000 #simulations

store.o <- rep(NA, sims)
for (i in 1:sims){
  sam <- sample(unique(pop$office),5)
  store.o[i] <- mean(pop[pop$office %in% sam,"satisfied"])
}

store.a <- rep(NA, sims)
for (i in 1:sims){
  sam <- sample(unique(pop$agent),25)
  store.a[i] <- mean(pop[pop$agent %in% sam,"satisfied"])
}

store.i <- rep(NA, sims)
for (i in 1:sims){
  sam <- sample(unique(pop$individual),5000)
  store.i[i] <- mean(pop[pop$individual %in% sam,"satisfie
```

Comparing sampling distributions



Conceptual practice: clusters

- Describe a monitoring situation where you might want to use clustered sampling
  - What are the clusters?
  - How would you choose the level of clustering?