`Survey` package: Two stage cluster sampling

Week 7

Stat 260, St. Clair

Two-stage cluster sampling estimation

The data format for two-stage cluster sampling data must be at the **SSU-level** ("raw" data)

- unlike one-stage data, the `survey` package cannot get correct estimates/SE from "cluster-level" data summaries

Two-stage cluster sampling estimation

clusters

obs. units.

```
> twostage_design <- svydesign(id = ~PSU + SSU,
+                              fpc = ~N + Mi,
+                              weights = ~wts,
+                              data = mydata)
> svymean(~y, twostage_design)   # ratio mean estimate
> svytotal(~y, twostage_design)  # unbiased total estimate
```

-> optional - include if you can

- `svytotal` will give you unbiased total estimates $\hat{t}_{unb}$
  - for **mean**: divide by M_0, if known, to get $\hat{\bar{y}}_{unb} = \hat{t}_{unb}/M_0$, SE and CI
- `svymean` will give you ratio (biased) mean estimates $\hat{\bar{y}}_r$
  - for **total**: multiply by M_0, if known, to get $\hat{t}_r = M_0\hat{\bar{y}}_r$, SE and CI

California API scores

- A SRS of 40 school districts was selected from the 757 districts in the state.
- Data from a SRS of schools within each selected district was collected.

```
> schools <- read.csv("http://math.carleton.edu/kstclair/data/califor
> glimpse(schools)
Rows: 126
Columns: 10
$ sname         <chr> "Alta-Dutch Flat Elementary", "Tenaya Elementar
$ snum          <int> 3269, 5979, 4958, 4957, 4956, 4915, 2548, 2550,
$ dname         <chr> "Alta-Dutch Flat Elem", "Big Oak Flat-Grvlnd Ur
$ dnum          <int> 15, 63, 83, 83, 83, 117, 132, 132, 132, 152, 15
$ api00         <int> 821, 773, 600, 740, 716, 811, 472, 520, 568, 59
$ growth        <int> 36, 55, -32, 0, 5, 32, 40, 26, -21, 6, -10, 29,
$ meals         <int> 27, 43, 33, 11, 5, 25, 78, 76, 68, 42, 63, 54,
$ ell           <int> 0, 0, 5, 4, 2, 5, 38, 34, 34, 23, 42, 24, 3, 6,
$ enroll        <int> 152, 312, 173, 201, 147, 234, 184, 512, 543, 33
$ district_size <int> 1, 1, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4,
```

California API scores: `id` entries

- PSU: district
  - `dname` or `dnum`
- SSU: school
  - `sname` or `snum`

California API scores: SSU (schools)

M_i = number of schools in district i

- district_size

→ 40 district

```
> schools_by_district <- schools %>%
+ * group_by(dnum) %>%  # group by cluster (district number)
+   summarize(M_i = first(district_size),
+             m_i = n())  # sample size per cluster
> summary(schools_by_district$M_i)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.750   3.000   6.775   5.000  72.000
```

$1 \le M_i \le 72$

California API scores: SSU

m_i = number of *sampled* schools in district i

- not given in the data set

$1 \leq m_i \leq 5$

```
> summary(schools_by_district$m_i)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    1.75    3.00    3.15    5.00    5.00
```

Add m_i to the original data set with `mutate`:

```
> schools <- schools %>%
+     group_by(dnum) %>% # group by cluster (district number)
+   mutate(m_i = n())   #  m_i = sample size per cluster
> schools %>% select(dnum, m_i) %>% glimpse()
Rows: 126 = # schools
Columns: 2
Groups: dnum [40]
$ dnum <int> 15, 63, 83, 83, 83, 117, 132, 132, 132, 152, 152, 152, 1
$ m_i  <int> 1, 1, 3, 3, 3, 1, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 1, 4, 4,
```

California API scores: weights

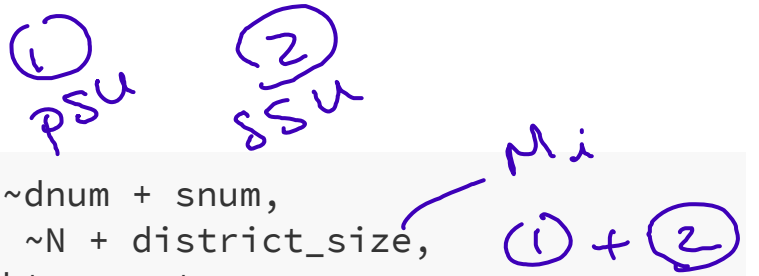
- Two-stage weights are $\frac{NM_i}{nm_i}$

```
> schools$N <- 757  # N
> schools$n <- 40   # n
> schools$wts <- (757/40)*schools$district_size/schools$m_i
> summary(schools$wts)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  18.93   18.93   18.93   40.70   26.50  272.52
```

PSU size

$18.93 \leq w_{ij} \leq 272.5$

California API scores: design




```
> schools_design <- svydesign(id= ~dnum + snum,
+                             fpc= ~N + district_size,
+                             weights = ~wts,
+                             data=schools)
```

California API scores

```
> svymean(~growth, schools_design, deff = TRUE) # ratio est.
          mean     SE   DEff
growth 25.778  2.842 1.5794
> svytotal(~growth, schools_design, deff = TRUE) # unbiased est.
         total     SE   DEff
growth 132206  41184 12.609
```

Lohr Examples 5.7

```
> coots <- read.csv("http://math.carleton.edu/kstclair/data/coots.csv
> library(dplyr)
> glimpse(coots)
Rows: 368 = # eggs sampled
Columns: 6
$ clutch  <int> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9,
$ csize   <int> 13, 13, 13, 13, 6, 6, 11, 11, 10, 10, 13, 13, 9, 9, 1
$ length  <dbl> 44.30, 45.90, 49.20, 48.70, 51.05, 49.35, 49.20, 48.5
$ breadth <dbl> 31.10, 32.70, 34.40, 32.70, 34.25, 34.40, 31.55, 33.1
$ volume  <dbl> 3.7957569, 3.9328497, 4.2156036, 4.1727621, 0.9317646
$ tmt     <int> 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

- `clutch`: clutch (nest) identifier (cluster)
- `csize`: clutch size (cluster size) M_i
- `volume`: egg volume

Lohr Examples 5.7: `id` entries

- PSU: clutch
  - `clutch` identifies clusters
- SSU: egg
  - Nothing to identify eggs
  - Add in a row number variable to do so (using `~1` doesn't work!)

```
> coots$elem.id <- 1:nrow(coots)    # unique id for each egg
```

368

Lohr Examples 5.7: SSU info

M_i = number of eggs in clutch i

- csize

```
> clutch_summary <- coots %>%
+   group_by(clutch) %>%
+   summarize(Mi = first(csize),   # size of each clutch
+             mi = n() )            # sample size of each clutch
> summary(clutch_summary$Mi)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  5.000   8.750  10.000   9.549  11.000  13.000
```

$5 \leq M_i \leq 13$

Lohr Examples 5.7: SSU info

m_i = number of *sampled* eggs in clutch i

- not given in the data set

$M_i = 2$

```
> summary(clutch_summary$mi)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      2       2       2       2       2       2
```

Add m_i to the original data set with `mutate`:

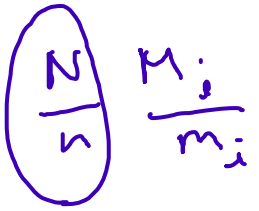
```
> coots <- coots %>%
+   group_by(clutch) %>%
+   mutate(mi = n())
> coots
# A tibble: 368 x 8
# Groups:   clutch [184]
   clutch csize length breadth volume   tmt elem.id    mi
    <int> <int>  <dbl>   <dbl>  <dbl> <int>   <int> <int>
 1      1    13   44.3    31.1   3.80     1       1     2
 2      1    13   45.9    32.7   3.93     1       2     2
 3      2    13   49.2    34.4   4.22     1       3     2
 4      2    13   48.7    32.7   4.17     1       4     2
 5      3     6   51.0    34.2   0.932    0       5     2
```

Lohr Examples 5.7: PSU info

N = number of clutches in the population

- unknown!
- But this is fine for estimating a **mean/proportion**
  - cluster sampling weights N/n: self-weighting design
  - like not knowing N for a SRS

Lohr Examples 5.7: weights with N unknown



- **Sampling weights** are proportional to $\frac{M_i}{m_i}$:

```
> coots$wts <- coots$csize/coots$mi
> summary(coots$wts)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.500   4.375   5.000   4.774   5.500   6.500
```

Lohr Examples 5.7: design

FPC – missing N = ?

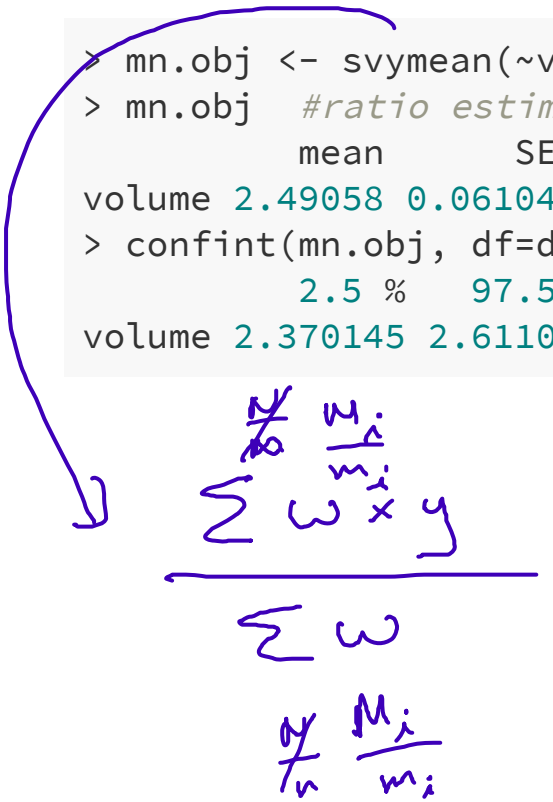
```
> coots_design <- svydesign(id = ~clutch + elem.id,
+                           weights = ~wts,
+                           data = coots)
> summary(coots_design)
2 - level Cluster Sampling design (with replacement)
With (184, 368) clusters.
svydesign(id = ~clutch + elem.id, weights = ~wts, data = coots)
Probabilities:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.1538  0.1818  0.2000  0.2187  0.2292  0.4000
Data variables:
[1] "clutch"  "csize"   "length"  "breadth" "volume"  "tmt"     "eler
[8] "mi"      "wts"
```

→ $\frac{m_i}{M_i}$

Lohr Examples 5.7

- The SE won't include the FPCs

```
> mn.obj <- svymean(~volume, coots_design, deff=T)
> mn.obj  #ratio estimate of pop. mean without FPC in SE
           mean      SE   DEff
volume 2.49058 0.06104 2.5755
> confint(mn.obj, df=degf(coots_design))
           2.5 %   97.5 %
volume 2.370145 2.611012
```



Lohr Examples 5.7

- The sampling weights are just $\frac{M_i}{m_i}$ instead of $\frac{NM_i}{nm_i}$
  - We need exact sampling weights to estimate any **total**

```
> svytotal(~volume, coots_design, deff=T)   # INCORRECT
         total      SE   DEff
volume 4375.95  165.89 6.1621
> sum(coots$volume*coots$wts)
[1] 4375.947
```

- 4375.95 is the estimated **total volume** of the **sampled clutches**
  - NOT the **total volume** of **all** clutches in the **population**.

$\sum wxy$ missing $\frac{N}{n}$ $\sum_i \sum_j \frac{M_i}{m_i} y_{ij}$