W1 Basic Estimation

Overview

Effects of Complex Design
• Weights accounted for in estimating totals, means, model parameters because complex samples are usually not miniatures of a population
• Weights, strata, and multiple stages of selection will affect SEs
• Software for analyzing complex samples allows you to specify all of these

Using Weights in simple estimates

- Total is estimated by $\hat{t} = \sum_{i \in s} w_i y_i$
- Mean is estimated by $\hat{\bar{y}} = \sum_{i \in s} w_i y_i / \sum_{i \in s} w_i \equiv \hat{t}/\hat{N}$
- Model parameter estimates usually depend on estimated totals
- Quantiles
  - Sort file from low to high on y
  - For median cumulate weights until 50% of total sum of weights is reached
  - y value for unit that is the first to have a cumulative of 50% or more of total weight is estimated median
  - Other quantiles (1^{st} or 3^{rd} quartile) are estimated in a similar way

Basic R Examples

Specifying Elements of Complex Design
• Software must be informed of weights, strata, clusters, **finite population corrections(fpc)** for variance estimates
• svydesign in R survey
• Example: stsrs from nhis population in PracTools package
• 4 strata, 100 persons selected per stratum

Nhis example with fpc

```
require(sampling)
data(nhis)
attach(nhis)
nhis <- nhis[order(educ_r), ]
stsam <- strata(data=nhis, stratanames="educ_r",
    size=rep(100,4),
    method="srswor", description=TRUE)

   Stratum 1
   Population total and number of selected units: 1964 100
   Stratum 2
   Population total and number of selected units: 719 100
   Stratum 3
   Population total and number of selected units: 933 100
   Stratum 4
   Population total and number of selected units: 295 100
   Number of strata  4
   Total number of selected units 400
```

```
range(unique(stsam$Prob))
#    [1] 0.0509165 0.3389831

samdat <- getdata(nhis,stsam)
samdat$svywt <- 1/samdat$Prob
nhis.dsgn <- svydesign(ids = ~NULL, strat = ~educ_r,
                       weights = ~svywt,
                       data = samdat,
                       fpc = stsam$Prob)
    # Omit fpc to show difference
nhis.dsgn.nofpc <- svydesign(ids = ~NULL, strat = ~educ_r,
                       weights = ~svywt,
                       data = samdat)
svymean(~age, design = nhis.dsgn)
          mean     SE
age 45.882 0.9757
svymean(~age, design = nhis.dsgn.nofpc)
          mean     SE
age 45.882 1.0117
```

Compare SEs with and without fpc's

Ratio of SEs without and with fpc: 1.0117/0.9757 = 1.036897

```
mns.fpc <- svyby(~age, by = ~educ_r, design = nhis.dsgn,
                 FUN = svymean)
  educ_r   age       se
1      1 42.98 1.752865
2      2 40.75 1.634159
3      3 45.86 1.428145
4      4 50.56 1.107398
mns.nofpc <- svyby(~age, by = ~educ_r, design = nhis.dsgn.nofpc,
                   FUN = svymean)
  educ_r   age       se
1      1 42.98 1.799269
2      2 40.75 1.761219
3      3 45.86 1.511439
4      4 50.56 1.362063

mns.nofpc[,3]/mns.fpc[,3]
1.026474 1.077753 1.058323 1.229968
```

Summary

- Strictly speaking, fpc's are appropriate for simple random samples selected without replacement, stratified srswor, or multistage samples with srswor at every stage
- Software allows ad hoc inclusion of fpc's for other designs
- Omitting fpc's (where they should be used) gives SEs that are too large

Multistage Design Example

- Example using nhis.large dataset in R PracTools package
  - US national health interview Survey
  - 21,588 persons
  - 75 strata, 2 PSUs per stratum

Design object in R

```
require(PracTools)
require(survey)
data(nhis.large)
nhis.dsgn <- svydesign(ids = ~psu, strat = ~stratum,
                       weights = ~svywt,
                       data = nhis.large,
                       nest = TRUE)
```

• Specifies that design is a stratified PSU sample with survey weights svywt
• Ultimate cluster (with replacement) variance estimators will be used since no other design information given

Table of Proportions in R

```
age.mns <- svyby(formula = ~factor(delay.med), by = ~age.grp,
                 FUN=svymean, design = nhis.dsgn, na.rm=TRUE)
age.mns <- age.mns[, c(2,4)]
rownames(age.mns) <- c("< 18 years", "18-24 years", "25-44 years",
                       "45-64", "65+")
colnames(age.mns) <- c("Proportion", "se(p)")
round(age.mns,4)
             Proportion  se(p)
< 18 years       0.0336 0.0035
18-24 years      0.0929 0.0081
25-44 years      0.0997 0.0050
45-64            0.0884 0.0045
65+              0.0366 0.0036
```

- Age group is a domain
- Young and old are less likely to delay medical care because of cost

Comparison to srs SEs

```
age.mns.srs <- by(abs(nhis.large$delay.med-2),
        INDICES = nhis.large$age.grp, FUN=mean, na.rm=TRUE)
age.mns.srs.SE <- sqrt(age.mnsB*(1 - age.mnsB)/table(nhis.large$age.grp))
round(cbind("Ratio of p.hats" = age.mns.srs/age.mns[,1],
            "Ratio of SEs" = age.mns.srs.SE / age.mns[,2]), 2)
  Ratio of p.hats Ratio of SEs
1            1.09         0.70
2            0.99         0.80
3            0.95         0.75
4            1.03         0.90
5            0.97         1.03
```

- Point estimates of proportions are similar whether weights are used or not
- SEs are too small if complex design (weights, strata, clustering) is ignored

Test for Independence

```
svychisq(~ delay.med + age.grp, nhis.dsgn,
    statistic="F")

        Pearson's X^2: Rao & Scott adjustment
data:  svychisq(~delay.med + age.grp, nhis.dsgn, statistic = "F")
F = 48.295, ndf = 3.6918, ddf = 276.8900, p-value < 2.2e-16
```

- Rao-Scott test is adjustment of Pearson's chi-square to account for complex design
- Age and delaying medical care because of cost are not independent

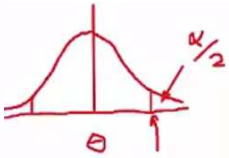
Degrees of Freedom

Degrees of Freedom
• Degrees of freedom (df) are associated with a variance estimator
• Related to stability of estimated variance
• As df increases, precision of variance estimator increases (variance of valance estimator decreases)

Rule of Thumb
• df = sum over strata of (n_h – 1) = (# of PSUs) – (# of strata)
• Using rule of thumb, degrees of freedom is largely determined by number of first stage units in sample
Numbers of sample units within each PSU not accounted for in rule-of-thumb
• Rule is not always accurate but is easy to apply
• Used by all survey software packages
➔ Suppose that a sample design has 4 strata, 2 clusters sampled per stratum, and 100 persons selected per sample cluster. What is the standard rule-of-thumb value for the degrees of freedom of a variance estimator? **Answer: 4**

Confidence Intervals

- $1-\alpha$ level confidence interval for some quantity θ computed as $\hat{\theta} \pm t_{1-\alpha/2}(df)\sqrt{v(\hat{\theta})}$
- $t_{1-\alpha/2}(df)$ is the $1-\alpha/2$ upper percentile of a t distribution with df degrees of freedom
- Validity depends on sample of first-stage units being large enough so that t approximation works



- Single-stage design with 150 school sample
  $df = 149; t_{0.975}(149) = 1.976$ 1.96
- Single-stage stratified sample of establishments with 3 strata and 25, 45, 75 establishments sampled in the 3 strata
  $df = (25-1)+(45-1)+(75-1) = 145-3 = 142$
  $t_{0.975}(142) = 1.977$
- Multistage sample with 10 strata, 2 PSUs selected per stratum with probability proportional to size, and 50 households selected per PSU by *srswor*
  $df = 20-10 = 10$
  $t_{0.975}(10) = 2.228$
  Note that the sample size of households per PSU does not enter into rule-of-thumb calculation

Basic STATA Examples

Specifying a Design in STATA
• svyset statement defines design features
• svyset psu field [pweight = *weight field*], strata(*stratum field*) fpc(*fpc field*)
• other design features can be specified: replicate weights (jackknife, balanced repeated replication (BRR), boostrap)

Means example in STATA
• Use stratified srswor from the nhis sample
• Stratrum level fpc's need to be used

```
use samdat.dta, clear
svyset ID [pweight=svywt], strata(educ_r) fpc(Prob)
      pweight: svywt
          VCE: linearized
  Single unit: missing
     Strata 1: educ_r
         SU 1: ID
        FPC 1: Prob
```

```
svy: mean age, over(educ_r)
Survey: Mean estimation

Number of strata =       4        Number of obs    =       400
Number of PSUs   =     400        Population size  =     3,911
                                  Design df        =       396

             1: educ_r = 1
             2: educ_r = 2
             3: educ_r = 3
             4: educ_r = 4

--------------------------------------------------------------
             |             Linearized
        Over |       Mean   Std. Err.     [95% Conf. Interval]
-------------+------------------------------------------------
age          |
           1 |      42.98   1.752865      39.53392    46.42608
           2 |      40.75   1.634159      37.53729    43.96271
           3 |      45.86   1.428145      43.05231    48.66769
           4 |      50.56   1.107398      48.38289    52.73711
--------------------------------------------------------------
```

Summary

• Stata and R survey use the same without-replacement SE estimator and give the same SEs
• By default STATA produces confidence intervals for each domain mean
t-intervals are used with $df = \sum_{h=1}^{4}(n_h - 1) = 396$

$\Rightarrow$ CIs use normal approximation since df is so large

Multistage Samples – example in stata
• Use the nhis.large dataset from R PracTools

```
use nhislarge.dta, clear
label define age_lab 1 "< 18" 2 "18-24" 3 "25-44" 4 "45-64" 5 "65+"
label values age_grp age_lab

svyset psu [pweight = svywt], strata(stratum)
svy: tabulate age_grp delay_med, row
```

- Specifies that design is a stratified PSU sample with survey weights `svywt`
- Ultimate cluster (with replacement) variance estimators will be used since no other design information given

```
Number of strata   =        75     Number of obs      =      21,464
Number of PSUs     =       150     Population size    =  66,261,032
                                   Design df          =          75
--------------------------------
           |      delay.med
   age.grp |     1      2  Total
-----------+--------------------
       <18 | .0336  .9664      1
     18-24 | .0929  .9071      1
     25-44 | .0997  .9003      1
     45-64 | .0884  .9116      1
       65+ | .0366  .9634      1
           |
     Total | .0719  .9281      1
--------------------------------
  Key:  row proportion

  Pearson:
    Uncorrected   chi2(4)         =  274.7136
    Design-based  F(3.69, 276.89) =   48.2948     P = 0.0000
```

Summary
• Stata estimated proportions are same as those from R
• By default, STATA computes the Rao-Scott test of independence
• *svy:tabulate* has other output options: estimated totals, Cls, def fs, unweighted cell counts

Quantiles

Quantiles

- Special methods required to get precision estimates
- Standard approach is to compute confidence intervals first
- If SE estimate desired, compute as $SE = L/(2t_{1-\alpha/2(df)})$ where L is length of CI and $t_{1-\alpha/2(df)}$ is a multiplier from a t distribution with df degrees of freedom

  For example, if $1 - \alpha = 0.95$, $SE = L/(2 \times 1.96)$
- Two options for getting CI's are Woodruff and Francisco-Fuller

Select a pps sample

```
require(PracTools)
require(sampling)
data(smho.N874)
    # recode hospitals with 0 beds
size <- smho.N874$BEDS
size[size <= 5] <- 5
pk <- inclusionprobabilities(size,n=100)
summary(pk)
#    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
#0.007214 0.009018 0.051940 0.114400 0.133800 1.000000
```

```
set.seed(858417834)
sam <- UPsystematic(pk) # vector of 874 0s and 1s
samdat <- getdata(smho.N874, sam)

    # append weights
samdat <- cbind(samdat, svywt = 1/pk[sam==1])
samdat$EXPTOTm <- samdat$EXPTOTAL/10^6

smho.dsgn <- svydesign(ids = ~NULL, strat = NULL,
                       weights = ~svywt,
                       data = samdat)
smho.dsgn
    Independent Sampling design (with replacement)
```

Compute Quantiles

```
svyquantile(~EXPTOTm, design = smho.dsgn,
            quantiles = c(0.25, 0.50, 0.75), ci=TRUE)
    $quantiles
                0.25      0.5     0.75
    EXPTOTm 4.520402 7.255868 11.67096

    $CIs
    , , EXPTOTm
                0.25      0.5     0.75
    (lower 1.047000  4.50318 7.510493
    upper) 7.314093 12.13289 23.740537
```

- `interval.type="Wald"` is default and is Woodruff method for CIs
- `interval.type="score"` gives Francisco-Fuller

Summary: quantiles

• survey software that will estimate quantiles and their SEs
- R survey: svyquantile
- SAS: proc surveymeans
- WesVar: replicate variance estimates only
- Stata: no SEs for quantiles (yet)