W3 Implementing the Steps

Software

R Packages
• Free software
• https://www.r-project.org
• Over 8,000 packages available
• Packages for this course
  - sampling
  - survey
  - PracTools

Editing code files in R
• Built in editor
• Other options
  - R-studio
  - WinEdt, use with RWinEdt (R Package)
  - Tinn-R

Other software packages
• SAS
• STATA
• SUDAAN
• WesVar

R examples are from the Springer text Practical Tools for Designing and Weighting Survey Samples (2013)

Base Weights

Select a sample and compute base weights
• R sampling package will select samples and compute base weights
• Example: select a stratified sample

```
require(sampling)
require(PracTools)
data(nhis)
table(nhis$educ_r, useNA="always")
   1    2    3    4 <NA>
1964  719  933  295    0
attach(nhis)
table(educ_r, useNA="always")
educ_r
   1    2    3    4 <NA>
1964  719  933  295    0
```

• sampling package will select various kinds of samples and compute base weights
Srswor ➔ simple random sampling without replacement
Size=rep(3,4) ➔ 4 stratum with 3 sample in each stratum

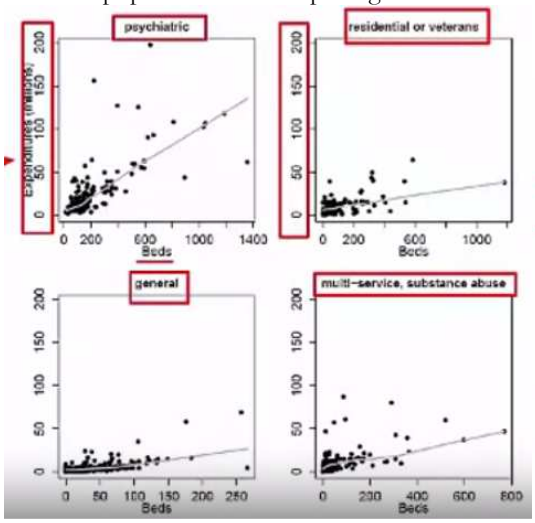
```
nhis <- nhis[order(educ_r), ]
stsam <- strata(data=nhis, stratanames="educ_r", size=rep(3,4),
          method="srswor", description=TRUE)
     educ_r ID_unit       Prob Stratum
506       1     506 0.001527495       1
1913      1    1913 0.001527495       1
1921      1    1921 0.001527495       1
2007      2    2007 0.004172462       2
2124      2    2124 0.004172462       2
2316      2    2316 0.004172462       2
3225      3    3225 0.003215434       3
3283      3    3283 0.003215434       3
3536      3    3536 0.003215434       3
3632      4    3632 0.010169492       4
3715      4    3715 0.010169492       4
3895      4    3895 0.010169492       4
```

```
wts <- 1/stsam$Prob
sum(wts)
    [1] 3911
by(wts, stsam$educ_r, sum)
      stsam$educ_r: 1
  [1] 1964
  ----------------------------
  stsam$educ_r: 2
  [1] 719
  ----------------------------
  stsam$educ_r: 3
  [1] 933
  ----------------------------
  stsam$educ_r: 4
  [1] 295
```

Repeating selection of same sample
• If strata is re-executed, it will use a different random start and select a different set of units
• When any R function that uses random numbers is called, a 626-vector called .Random.seed is formed
• Numbers in position 3:626 can be used a random starts
• Issue a command like this before selecting the sample:
Set.seed(4533223)

More on base weights

Select a pps sample and compute base weights
• pps sampling is efficient if frame has a measure of size that is a good predictor of variables to be collected
• SMHO pop in PracTools R package




```
require(PracTools)
require(sampling)
data(smho98)
    # recode hospitals with 0 beds
size <- smho98$BEDS
size[size <= 5] <- 5
pk <- inclusionprobabilities(size,n=10)
summary(pk)
#      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
#0.0006717 0.0008732 0.0048360 0.0114300 0.0124900 0.3231000

set.seed(858417834)
sam <- UPsystematic(pk) # vector of 875 0s and 1s
samdat <- getdata(smho98, sam)
    # append weights
samdat <- cbind(samdat, wt = 1/pk[sam==1])
samdat[1:3, c(1:4,14)]
    ID_unit STRATUM BEDS  EXPTOTAL       wt
36       36       1  103    9853392 72.27476
103     103       1  139    9853392 53.55612
156     156       2 1048  106775841  7.10334
```

If two units have the same measure of size, then they would have the same selection probability (as long as a strictly pps procedure is used). There are methods of selection where the remaining total size is recomputed after each draw, but that type of method makes computing selection probabilities extremely difficult (and is not usually used in practice).

Multistage Samples in R
• sampling package will select some types of multistage samples
• Or, use *sampling* to select first-stage (clusters)
  - extract sample data
  - treat first-stage units as strata and select a stratified sample with *strata* function

SAS & Stata
• SAS
  - proc surveyselect
  - simple random, Bernoulli, Poisson, pps samples
• Stata
  - simple random samples
  - user written packages or your own code for other types of samples
(http://www.ats.ucla.edu/stat/seminars/default.htm)

Non-probability Samples
• Quasi-randomization weights are sometimes computed
• Reference sample
  - census microdata
  - small parallel probability sample
  - large independent probability sample
• Combine non-probability sample with reference sample
  - estimate "pseudo-probability" of being in non-probability sample
  - use inverse of pseudo-probability as "base weight"
  - need extensive set of covariates in reference and non-probability samples to be effective

Nonresponse Adjustments

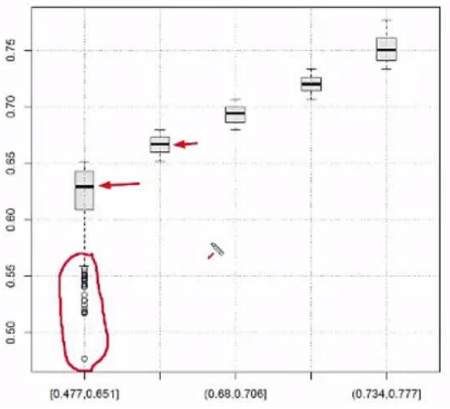
Propensity Classes
• Estimate response propensities using logistic regression
• Sort propensities from low to high
• Divide sample into classes with about same number of cases in each class
  - all sample cases (Rs and NRs) are classified
  - Equal size classes means number of R + NR about same in each class
• Adjust weights for each R in a class
• Approach can be applied to probability samples with NR or to non-probability samples to compute quasi-randomization weights

Pclass in PracTools

```
require(PracTools)
data(nhis)
out <- pclass(formula = resp ~ age + as.factor(sex) +
                as.factor(hisp) + as.factor(race),
                data = nhis,
                type = "unwtd",
                link="logit",
                numcl=5)
table(out$p.class, useNA="always")
[0.477,0.651]  (0.651,0.68]  (0.68,0.706] (0.706,0.734]
          790           789           772           796
  (0.734,0.777]          <NA>
            764             0
summary(out$propensities)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.4769  0.6586  0.6924  0.6901  0.7260  0.7765

boxplot(out$propensities ~ out$p.class,
        col = "gray90",
        boxwex = 0.25)
```

Note spread of estimates in first class



NR weight adjustments
• Multiply input weights by inverse of cell response propensities
• Possibilities for cell adjustments
  - unweighted mean response propensity from model
  - unweighted mean response propensity (base weights)
  - median response propensity
  - unweighted RR
  - weighted RR

Alternative Adjustments

```
round(cbind(
    "mean" = by(data=out$propensities, INDICES=out$p.class,
             FUN=mean),
    "median" = by(data=out$propensities, INDICES=out$p.class,
              FUN=median),
    "wtd RR" = by(data.frame(resp=nhis[,"resp"],
              wt=nhis[,"svywt"]),
              out$p.class,
              function(x) {weighted.mean(x$resp, x$wt)})),3)


               mean median wtd RR
[0.477,0.651] 0.621  0.630  0.645
(0.651,0.68]  0.666  0.667  0.665
(0.68,0.706]  0.693  0.694  0.690
(0.706,0.734] 0.720  0.720  0.732
(0.734,0.777] 0.752  0.750  0.764
```

A reason for grouping estimated response propensities into classes rather than using individual propensities is … Individual estimated propensities can be extremely variable and can lead to extreme weight adjustments

Calibration

Examples of Calibration
• Probability sample input weights: base weights adjusted for NR
• Non-probability sample input weights: set all weights to 1 or use quasi-randomization weights
• Use auxiliaries to reduce variances or correct for coverage errors
• Need population totals for each auxiliary used
• Examples
- Post stratification
- Raking
- GREG – allows both qualitative and quantitative x's

- PS estimator of a total is defined as

$$\hat{T}_{yPS} = \sum_{\gamma=1}^{G} N_\gamma \left(\hat{t}_{y\gamma} \Big/ \hat{N}_\gamma \right)$$

where $\hat{t}_{y\gamma} = \sum_{s_\gamma} d_i y_i$ is estimated total of y in weighting class (or poststratum) γ based on input weights

  - s_γ is set of sample units in poststratum γ
  - $\hat{N}_\gamma = \sum_{s_\gamma} d_k$ is estimated population size of poststratum γ based on the input weights
  - N_γ is population count (aka a control or control total) for poststratum γ
  - G is total number of poststrata

Post Stratification

- The implied final weight for unit i in poststratum γ is

$$w_i = d_i \frac{N_\gamma}{\hat{N}_\gamma}$$

where $g_i = N_\gamma \Big/ \hat{N}_\gamma$ is poststratification adjustment (factor).

- g_i is called a g-weight in general calibration equation $w_i = d_i g_i$.
- The PS estimator can be written as $\hat{T}_{yPS} = \sum_{i \in s} w_i y_i$, i.e., a weighted sum of the data values.
- Weighting classes are called poststrata because they are applied after the sample is selected
- Crosses of variables can be used: (age group) × (gender)

Post Stratification Example
• Use survey package in R

```
require(survey)
data(api)
dclus1 <- svydesign(id = ~dnum, weights = ~pw, data=apiclus1, fpc = ~fpc)

    # post-stratify on school type
pop.types <- data.frame(stype=c("E","H","M"), Freq=c(4421,755,1018))
dclus1p <- postStratify(dclus1, ~stype, pop.types)

rbind(summary(weights(dclus1)), summary(weights(dclus1p)))
          Min. 1st Qu. Median  Mean 3rd Qu.  Max.
    [1,] 33.85   33.85  33.85 33.85   33.85 33.85
    [2,] 30.70   30.70  30.70 33.85   30.70 53.93
```

```
svymean(~enroll, dclus1)
          mean     SE
enroll 549.72 45.191
svymean(~enroll, dclus1p)
          mean     SE
enroll 594.27 65.594
svytotal(~enroll, dclus1)
          total     SE
enroll 3404940 932235
svytotal(~enroll, dclus1p)
          total     SE
enroll 3680893 406293

c(cv(svymean(~enroll, dclus1)), cv(svymean(~enroll, dclus1p)))
[1] 0.0822086 0.1103788

c(cv(svytotal(~enroll, dclus1)), cv(svytotal(~enroll, dclus1p)))
[1] 0.2737890 0.1103788
```

➔ In the following code for instance
svydesign(id = ~ county, weights = ~ bwt, strata = ~region, data=mysample, fpc = ~ finpopc)
"county" variable is the what specifies the "cluster"

Model Post Stratification

- Every estimator has an implied model behind it
- PS model: common mean and variance within each PS

$$E_M(y_i) = \beta_\gamma, Var_M(y_i) = \sigma_\gamma^2$$

- If model is approximately right, PS estimator will be efficient (low variance)
- If model is wrong, PS estimator will approximately design-unbiased but inefficient
- Checking model for important y's is a good step
  - Omitted covariates is one type of model failure
  - For example, PS defined by age $\times$ gender, but race-ethnicity and income level also important
  - If so, consider raking or GREG