Dealing with Complex Surveys in R

Dino P. Christenson

Department of Political Science
Ohio State University
www.polisci.osu.edu/grads/dchristenson

May 21, 2009

Resources Online

- These slides (.pdf) and the R script (.R) for these examples are online
- Visit http://polisci.osu.edu/grads/dchristenson/research.htm
- Download the resources
- Note: you will have to install R before using the script

Costs vs Precision

- Costs relative to sample size
- Know something about the population
- Want more precision for particular groups

Benefits of Complex Surveys

- Clusters exchange precision for costs
- Stratifications increase precision given knowledge of groups
- Subsamples lead to better precision for that group
- Finite Populations lead to little or no variability

Problems of Treating Complex as Simple

- Clustered sample will usually underestimate standard errors
- Usually, unequal probability sample will underestimate standard errors
- Stratified sample will overestimate standard errors
- Samples with units of unequal probability require weights
- Weight units according to their probability of inclusion

Solution: Weighting

Example (Basic Idea of Inverse Probability Weighting)

Suppose you were curious about different OSU graduate student reactions to the change to the semester system.

- Survey: You had a limited budget, so you stratified and sampled 10 graduate students from each department at OSU.
- Issue: Different number of students per department. For eg, PoliSci has 50 students and Communications has 25 students.
- Solution: Weight the PoliSci Dept twice as much as the Communications Dept.
- But what about the variances?

Estimating Variances

- Many options...
- Linearisation (Taylor Series)
- Replicates (Jackknife)
- For a technical explanation see Wolter, K. M. (1985) Introduction to Variance Estimation.

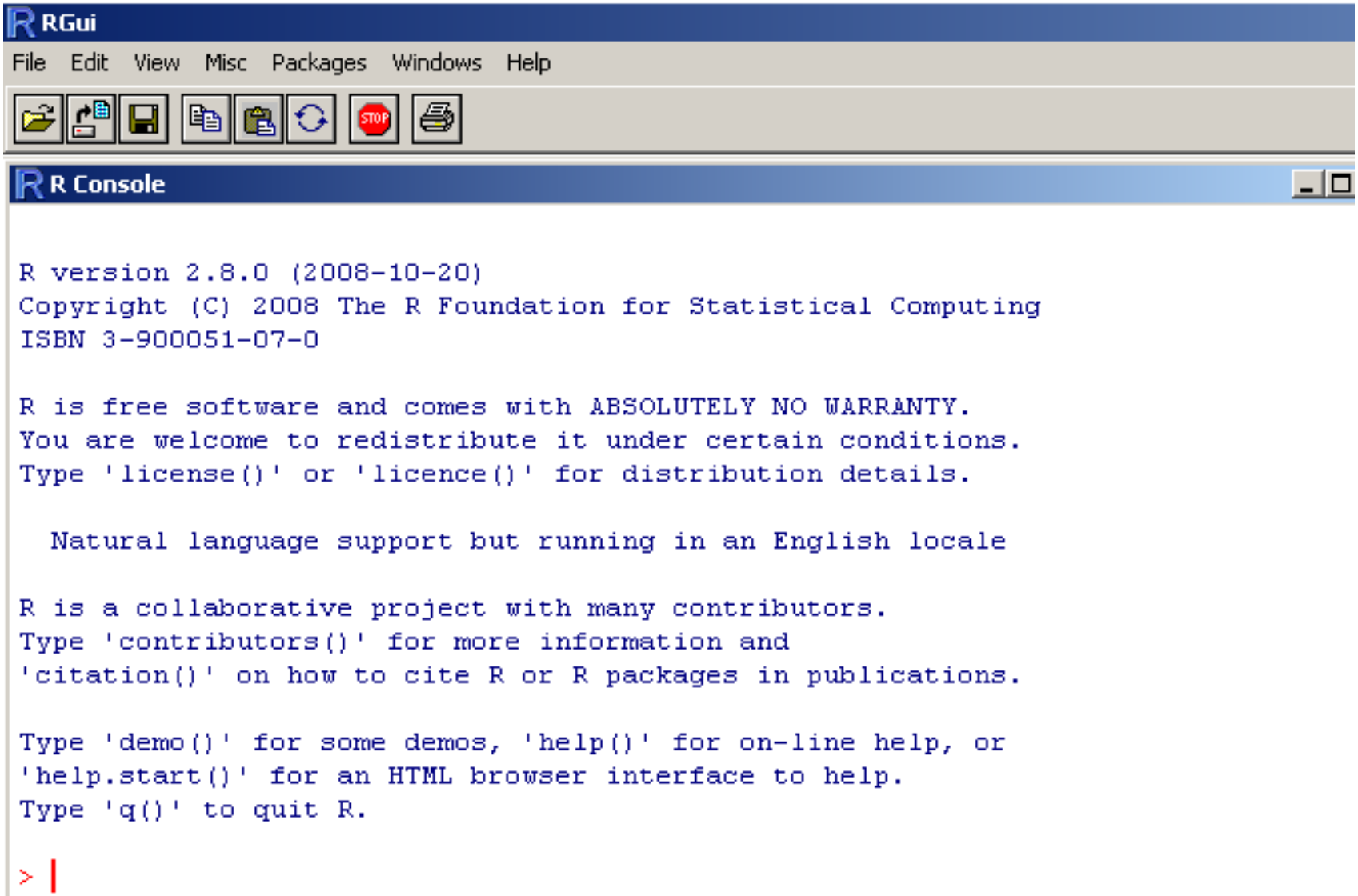
R Environment

- "R is a language and environment for statistical computing and graphics."
- Software used for data manipulation, data analysis, and pretty graphical output
- Elements of the "environment": programming language, runtime environment, graphics and a debugger
- Bottom Line: It's a statistics package
- Get it at the R Project web page: http://www.r-project.org/

R Flexibility

- Design based on computer language (similar to S)
- No reliance on preexisting tools/functions
- Users can program their own code
- Packages offer handy shortcuts (all packages available at the site)
- Flexibility is well suited to statistical simulation
- Graphical capabilities: Publication quality with high degree of manipulation
- Highly Interactive: User has to know what's going on "under the hood"
- It's Free!

R GUI


RGui
File Edit View Misc Packages Windows Help
R Console
R version 2.8.0 (2008-10-20)
Copyright (C) 2008 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>

Survey

- Binds meta data and computes appropriate variance statistics
- Then acts as a simple wrapper for typical R analyses
- Combines these features in one package
- Previously needed specialized software like SUDANN, WesVar or Stata
- Lumley offers great supporting vignettes (from which the examples in this presentation are taken)
- Lumley: http://cran.r-project.org/web/packages/survey/index.html

Reweight

- Reweight marginal distributions of survey data based on meta data
- Chen: http://cran.r-project.org/web/packages/reweight/reweight.pdf

Zelig

- Run common models with survey weights
- Inclusive package acts as a wrapper for all (kinds of) functions
- King: http://cran.r-project.org/web/packages/Zelig/index.html

Create Design Object

- Given design info and Taylor series preference, use *svydesign*
- Given design info and replicate weights preference, use *svydesign* then *as.svrepdesign*
- Given replication weights, use *svrepdesign*
- Once you've combined the meta data, proceed with analyses in wrapper
- That's it.

CA Public Schools

Example (Tests of Public School Students)

Data on CA public schools from http://www.cde.ca.gov/psaa/api.
Variables:

- *snum* is the school identifier
- *strat* is a stratum identifier based on stype
- *stype* is the school level (elementary, middle, high)
- *fpc* is the number of schools in a stratum
- *pw* is the sampling weights
- *api*00 is the performance score
- *dnum* is the district identifier
- *ell*, *meals* and *mobility* are social disadvantage measures

Example Procedure

For all these examples we will

- Take full dataset and peform some kind of sample from it
- Appropriately amend new dataset
- Run analyses with amended dataset

Stratified Set-Up

- Sample stratified by level of school
- Create survey design object with *svydesign*
- Look at design description

Stratified Design

```
> dstrat<-svydesign(id=~1,strata=~stype, weights=~pw,
data=apistrat, fpc=~fpc)

> dstrat  # design of survey
Stratified Independent Sampling design
svydesign(id = ~1, strata = ~stype, weights = ~pw,
data = apistrat,
    fpc = ~fpc)
```

Stratified Design

```
> summary(dstrat)  # more detail
Stratified Independent Sampling design
svydesign(id = ~1, strata = ~stype, weights = ~pw,
data = apistrat,
    fpc = ~fpc)
Probabilities:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.02262 0.02262 0.03587 0.04014 0.05339 0.06623
Stratum Sizes:
             E  H  M
obs        100 50 50
design.PSU 100 50 50
actual.PSU 100 50 50
Population stratum sizes (PSUs):
   E    M    H
4421 1018  755
```

Stratified Analyses

```
> svymean(~api00+I(api00-api99), dstrat) # estimate mean api score
                    mean     SE
api00             662.287 9.4089
I(api00 - api99)   32.893 2.0511

> svytotal(~enroll, dstrat)  # estimate total enrollment
         total     SE
enroll 3687178 114642
```

Stratified Analyses

```
> summary(svyglm(api00~ell+meals+mobility, design=dstrat))  # Regressio

Call:
svyglm(api00 ~ ell + meals + mobility, design = dstrat)
Survey design:
svydesign(id = ~1, strata = ~stype, weights = ~pw, data = apistrat,
    fpc = ~fpc)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 820.8873    10.0777  81.456   <2e-16 ***
ell          -0.4806     0.3920  -1.226    0.222
meals        -3.1415     0.2839 -11.064   <2e-16 ***
mobility      0.2257     0.3932   0.574    0.567
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
(Dispersion parameter for gaussian family taken to be 5171.966)
Number of Fisher Scoring iterations: 2
```

2 Stage Cluster Set-Up

- Two-stage cluster-sampled design
- 40 school districts sampled
- Then sample again up to five schools from each district

2SC Design

```
> dclus2<-svydesign(id=~dnum+snum, fpc=~fpc1+fpc2, data=apiclus2)

> summary(dclus2)
2 - level Cluster Sampling design
With (40, 126) clusters.
svydesign(id = ~dnum + snum, fpc = ~fpc1 + fpc2, data = apiclus2)
Probabilities:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.003669 0.037740 0.052840 0.042390 0.052840 0.052840
Population size (PSUs): 757
```

Stratified Cluster Design

```
> dstratclus<-svydesign(id=~dnum, strata=~stype, weights=~pw, data=apis

> summary(dstratclus)
Stratified 1 - level Cluster Sampling design (with replacement)
With (162) clusters.
svydesign(id = ~dnum, strata = ~stype, weights = ~pw, data = apistrat,
    nest = TRUE)
Probabilities:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.02262 0.02262 0.03587 0.04014 0.05339 0.06623
Stratum Sizes:
             E  H  M
obs        100 50 50
design.PSU  75 42 45
actual.PSU  75 42 45
```

JackKnife

```
> rclus1<-as.svrepdesign(dclus1)

> summary(rclus1)
Call: as.svrepdesign(dclus1)
Unstratified cluster jacknife (JK1) with 15 replicates.
```

Boostrap

```
> bclus1<-as.svrepdesign(dclus1,type="bootstrap", replicates=100)

> summary(bclus1)
Call: as.svrepdesign(dclus1, type = "bootstrap", replicates = 100)
Survey bootstrap with 100 replicates.
```

Linearization Vs Replicates

```
> svymean(~api00, dclus1)
        mean     SE
api00 644.17 23.542
> svytotal(~enroll, dclus1)
         total     SE
enroll 3404940 932235
> svymean(~api00, rclus1)
        mean     SE
api00 644.17 26.329
> svytotal(~enroll, rclus1)
          mean     SE
enroll 3404940 932235
> svymean(~api00, bclus1)
        mean     SE
api00 644.17 22.968
> svytotal(~enroll, bclus1)
          mean     SE
enroll 3404940 972072
```

Survey Software and Packages

- Lumley's vignettes and presentations: http://faculty.washington.edu/tlumley/survey/
- Verbeke's replications: http://cran.fhcrc.org/web/packages/SDaA/SDaA.pdf
- ATS at UCLA: http://statcomp.ats.ucla.edu/survey/

Survey Design Books

- Weisberg (2005). Total Survey Error Approach
- Groves, Fowler, Couper, Lepkowski, Singer and Tourangeau (2004). Survey Methodology
- Weisberg, Krosnick and Bowen (1996). An Intro to Survey Research, Polling and Data Analysis

Survey Sampling Books

- Lohr (1999). Sampling: Design and Analysis (used in sampling course in Statistics Department)
- Cochran (1977). Sampling Techniques
- Levy and Lemeshow (1999). Sampling of Populations: Methods and Applications

Discussion

- Questions?
- Comments?
- End