Complex Sampling Designs - Exercise 3

Stefan Zins, Matthias Sand and Jan-Philipp Kolb

6 February 2016

Exercise 3a

1. Download the dataset for Germany of the 5th ESS-Round (SDDF File and Sampling Data)
2. Create a `svydesign` object to estimate the mean of the variable `agea`
3. To acknowledge that the sample has been collected by a multi stage design, estimate the design effect of your estimate above using the PSU-Indicator variable (Use the model based approach described on slide 20 of today's lecture)

   Advice: the variable `PSU` has to be a factor

4. Calculate the effective sample size

Obtaining MSB, MSW and b^*

```
Ger.d <- read.spss("ESS5DE.spss/ESS5DE.sav",
                   to.data.frame = TRUE,
                   use.value.labels = TRUE)
Ger.ctry <- read.spss("ESS5_DE_SDDF.spss/ESS5_DE_SDDF.por",
                      to.data.frame = TRUE,
                      use.value.labels = TRUE)

colnames(Ger.d)[5] <- "IDNO"
Ger <- merge(Ger.d,Ger.ctry,by="IDNO", all.x = TRUE)
Ger$PSU <- as.factor(Ger$PSU)
n <- nrow(Ger)
L <- length(unique(Ger$PSU))
```

```
## deffc
b.star <- sum(tapply(Ger$dweight,Ger$PSU,
                function(x)sum(x)^2))/sum(Ger$dweight^2)
# Calculate an anova for the regression model Age by PSU
# (Could also be any other variable)
lin.mod <- lm(as.numeric(Ger$agea)~Ger$PSU)
SS <- anova(lin.mod)
#  MSB and MSW are the means of SSB and SSW
MSB <- SS$`Mean Sq`[1]
MSW <- SS$`Mean Sq`[2]
```

- Execute the following R-Script to generate a Multistage- and a Cluster- Sample for the belgianmunicipalities dataset

```
url <- "http://raw.githubusercontent.com/BernStZi/SamplingAndEstimation/short/tutorial/Samples_for_EX3b
source(url)
```

- Your workspace now contains the objects: `true_income`, `Data.be` and `Data.be2`. `true_income` resembles the mean of the income variable for the population of the `belgianmunicipalities` dataset. `Data.be` is a multistage sample with 80 PSUs and 300 individual datapoints whithin each PSU. `Data.be2` is a clustersample of 10 communes
- Estimate the mean income from both samples using the `survey` package and compare the results to the population mean

---

```
surv <- svydesign(id=~Commune+id,fpc=~prob1+prob2,
                  data=Data.be,pps="brewer")
```

- In **Exercise 1** we had a single-stage sample, therefore the argument `id` has been set to 0 or 1

In case of a multistage sampling approach, every sampling stage has to be defined

PSU: `Commune`; SSU: `id`

- This also applies for the `fpc`-argument
  `prob1` reflects the porbability of inclusion for each PSU in the sample and `prob2` the probability of inclusion for each SSU

Note: altough $prob1 * prob2 = n/N$ in this sample, it cannot be treated like a SRS

- `pps` should be used to define the design information; usually the second order probability of inclusion
  If the second order probability of inclusion are unknown (or too complex to calculate), a brewer approximation can be applied to estimate the joint inclusion probabilities