Sampling, Weighting and Estimation
Exercise 1

Stefan Zins, Matthias Sand
and Jan-Philipp Kolb

GESIS - Leibniz Institute
for the Social Sciences

February 2, 2016

Exercise 1

gesis
Leibniz Institute
for the Social Sciences

1. Download the ESS dataset for Sweden (Survey Data and Sampling Design Data File (SDDF)) of the 5th round

   `http://www.europeansocialsurvey.org/data/country.html?c=sweden`
2. Setup your workspace and load the R-packages `foreign` and `survey`
3. Load the ESS dataset and the SDDF
4. Merge both data frames by their ID-variable, using the `merge()` command

Exercise 1

gesis Leibniz Institute for the Social Sciences

5. Determine the sampling strategy (Inspect the variables PSU,STRATFY and PROB)
6. Add the variable `N` for the population size to your data frame. `N` can be calulated by

$$\texttt{N} = \texttt{dweight} * \texttt{pweight} * 10000 * n,$$

   where n refers to the sample size
7. Create a `svydesign` object from the dataset for Sweden using the `survey` package
8. Estimate the total and mean of the variable `tvtot`

gesis
Leibniz Institute
for the Social Sciences

- The survey package provides a large range of applications for complex survey samples
- Typically, the first step is to define a survey object with the `svydesign()` command

Simple Survey Object (Simple Random Sample)

```
data(api)

surv.obj <- svydesign(id=~1,fpc = ~fpc, data = apisrs)
```

```
surv.obj <- svydesign(id=~1,fpc = ~fpc, data = apisrs)
```

- `id` specifies the identifier of PSU and SSU; `id= ~0` or `id=~1` stipulates a single stage sampling
- For multi-stage samples the `id` argument should always specify a formula with the (cluster-) identifier at each stage
- `fpc` should be used for the finite population correction

$\Rightarrow$ Either as the total population size of each stratum or as a fraction of the total population that has been sampled

- `data` reflects the data set for which the design object should be defined

Important commands

`svytotal`	returns the estimated total of a variable and its standard error (+*deff*)
`svymean`	returns the estimated mean of a variable and its standard error (+*deff*)
`svyquantile`	Computes quantiles for data from complex surveys
`svyvar`	Computes variances for data from complex surveys
`weights`	Returns the (design) weights of a survey object
`calibrate`	Calibration of a data set (uses the GREG-Estimator by default)
...	...

```
svytotal(~api00,surv.obj)

##          total    SE
## api00 4066888 57293
```