```
> library(knitr)
> opts_chunk$set(
+ concordance=TRUE
+ )
```

Sampling, Weighting and Estimation Exercise 1

Stefan Zins, Matthias Sand
and Jan-Philipp Kolb

GESIS - Leibniz Institute
for the Social Sciences

January 25, 2016

Exercise 1

gesis
Leibniz Institute
for the Social Sciences

1. Download the ESS dataset for Sweden (Sampling Data and Country File) 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 country file
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 SAMPPOIN,STRATFY and PROB)
6. Add the variable `N`$=$ `dweight` $*$ `pweight` $*$ 10000 $*$ n to your data frame that equals the total population size for the target population of the ESS Round 5 of Sweden
7. Convert your data frame to a survey object (`survey`-Package)
8. Estimate the total and mean of the variable `tvtot`

- 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)

```
library(Matrix)
library(survey)

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

gesis Leibniz Institute for the Social Sciences

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)
...	...

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

```
        total    SE
api00 4066888 57293
```