# Sampling and Estimation

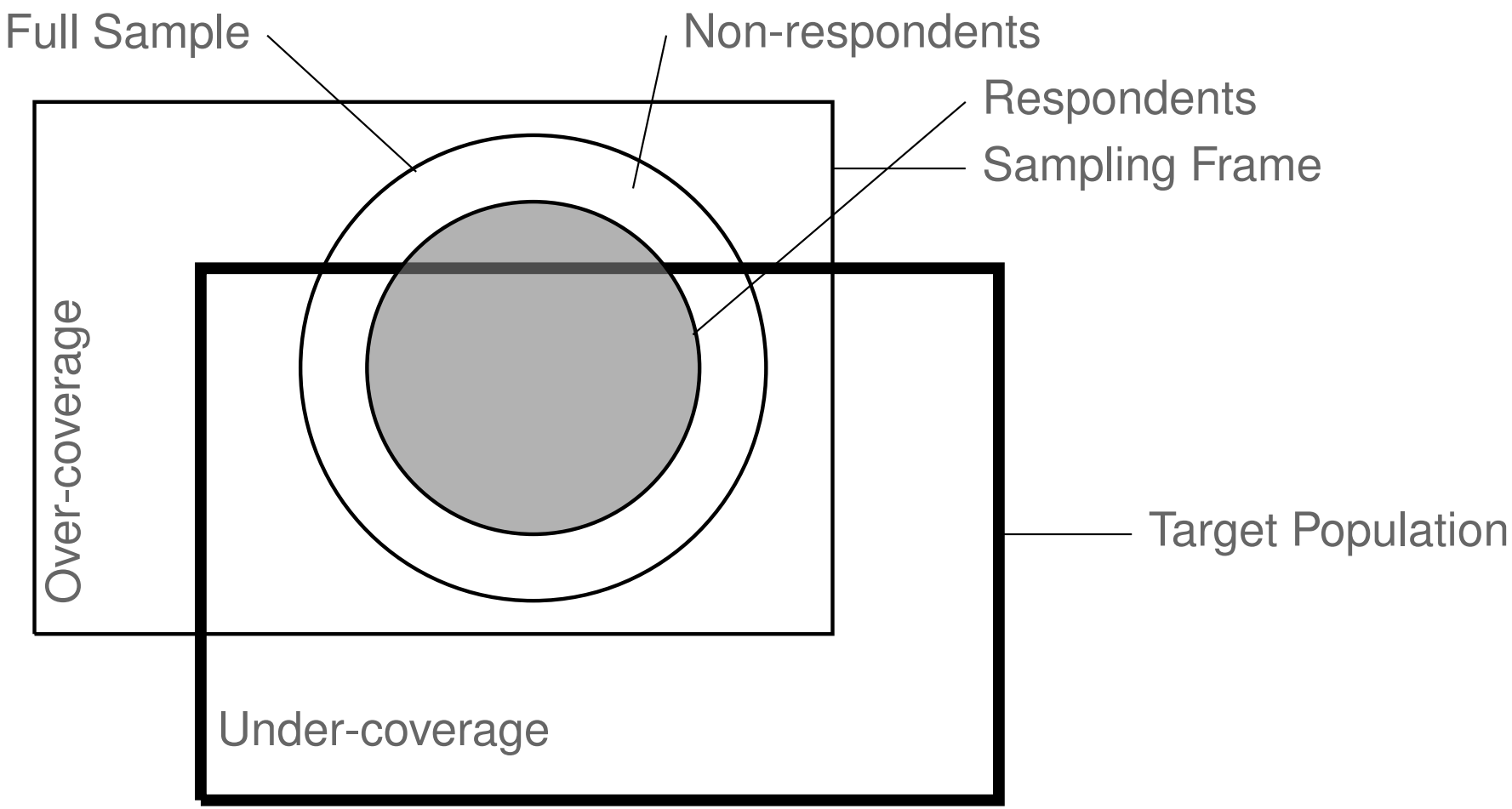
## Day 3: Estimation in Complex Survey Designs

Stefan Zins[1] and Matthias Sand[2]

January 22, 2016

[1] Stefan.Zins@gesis.org
[2] Matthias.Sand@gesis.org

# Survey Imperfections



Procedures based on the **available cases** only, i.e., only those cases that are completely recorded for the variables of interest.

Procedures based on the **available cases** only, i.e., only those cases that are completely recorded for the variables of interest.

**Weighting procedures** that adjust design weights to compensate the bias that a MAR non-response might inflict on HT-type estimators.

Procedures based on the **available cases** only, i.e., only those cases that are completely recorded for the variables of interest.

**Weighting procedures** that adjust design weights to compensate the bias that a MAR non-response might inflict on HT-type estimators.

**Single imputation** and correction of the variance estimates to account for imputation uncertainty.

Procedures based on the **available cases** only, i.e., only those cases that are completely recorded for the variables of interest.

**Weighting procedures** that adjust design weights to compensate the bias that a MAR non-response might inflict on HT-type estimators.

**Single imputation** and correction of the variance estimates to account for imputation uncertainty.

**Multiple imputation** (MI) according to Rubin (1978, 1987).

Procedures based on the **available cases** only, i.e., only those cases that are completely recorded for the variables of interest.

**Weighting procedures** that adjust design weights to compensate the bias that a MAR non-response might inflict on HT-type estimators.

**Single imputation** and correction of the variance estimates to account for imputation uncertainty.

**Multiple imputation** (MI) according to Rubin (1978, 1987).

$\longrightarrow$ Methods for handling coverage errors are not so widely spread, simply because there is often no reliable auxiliary information on just the target population. However if there is, it can receive a treatment similar to that of weighting by non-response.

# Missing Data

*Missing data is the norm, rather than the expectation!*

Missingness may be either

# Missing Data

*Missing data is the norm, rather than the expectation!*

Missingness may be either

**MCAR** missing completely at random,

every unit has same response propensity (RP)

respondents are a random sample of the initial sample

# Missing Data

*Missing data is the norm, rather than the expectation!*
Missingness may be either

**MCAR** missing completely at random,

every unit has same response propensity (RP)
respondents are a random sample of the initial sample

**MAR** missing at random, or

RP depends on auxiliary variables $\mathcal{X}$
can be modeled, if $\mathcal{X}$ is known for both respondents & non-respondents

# Missing Data

gesis Leibniz Institute for the Social Sciences

*Missing data is the norm, rather than the expectation!*
Missingness may be either

**MCAR** missing completely at random,

every unit has same response propensity (RP)
respondents are a random sample of the initial sample

**MAR** missing at random, or

RP depends on auxiliary variables $\mathcal{X}$
can be modeled, if $\mathcal{X}$ is known for both respondents & non-respondents

**MNAR** missing not at random

RP depends on variables of interest $\mathcal{Y}$
cannot be modeled, because $\mathcal{Y}$ not known for non-respondents

[Rubin and Little 2002]

$\longrightarrow$ In multivariate analysis often 30% to 40% of the data are lost with case deletion assuming MCAR!

**Calibration approach** The design weighs are calibrated to the totals of some auxiliary variables $\mathcal{X}$.

Sample estimates using the calibrated weights will exactly replicated those totals.

If the used auxiliary variables help to explain the response process the calibrated weight can reduce the non-response error.

**Two-phase approach** The response process is modeled to obtain the response propensities $\psi_k$ for all $k \in \mathcal{s}$. The new weight of element $k$ is $\frac{d_k}{\psi_k}$. (Two phases: 1. Sampling $\rightarrow$ 2. Responding).

In addition the new weights $\frac{d_k}{\psi_k}$ might then also be calibrated.

Often used models are:

Response homogeneity classes, every element in a class has the same probability to respond.
Generalized liner models (*probit*, *logit*, *log-log*), treating response as a latent variable.

The calibration approach is more direct as the design weights are directly calibrated without considering the response propensities. Also, if the same models are used for both the modeling of the response propensities and the calibration the two approaches can be equivalent.

Generic estimators for a total and a mean

$$\hat{\tau}_w = \sum_{k \in s} w_k y_k \quad \text{and} \quad \overline{y}_w = \frac{\sum_{k \in s} w_k y_k}{\sum_{k \in s} w_k} \ ,$$

where $w_k$ is the survey weight of element $k$, with

Generic estimators for a total and a mean

$$\hat{\tau}_w = \sum_{k \in s} w_k y_k \quad \text{and} \quad \overline{y}_w = \frac{\sum_{k \in s} w_k y_k}{\sum_{k \in s} w_k} \ ,$$

where $w_k$ is the survey weight of element $k$, with

$$w_k = \begin{cases} d_k g_k & \text{for } k \in s \\ 0 & \text{else} \end{cases} \ .$$

$$w_k = \begin{cases} d_k g_k & \text{for } k \in \mathcal{s} \\ 0 & \text{else} \end{cases} .$$

Sometimes called base weights or design weights, the inverse of inclusion probabilities $d_k = \pi^{-1}$ is usually the first step in weighting. If we have $g_k = 1$ the $\hat{\tau}_w$ would be the HT estimator or $\pi$-estimator. The factor $g_k$ adjusts the design weights to reduce

- the sampling error (i.e. variance),
- the non-response error, and
- the coverage error

of estimator $\hat{\tau}_w$ or $\overline{y}_w$. Thereby the $w_k$'s should not deviate to much from the $d_k$'s as these weights ensure an unbiased estimation.

# Weights Calibration I

The general idea is to exploit the relationship between auxiliary variables and the variable of interest to improve the efficiency of estimators.

The following problem is solved with weight calibration:
For a give design $p(.)$ and a sample $\mathcal{s}$ weights $w_k$ for all $k \in \mathcal{s}$ have to be found that minimize

$$\sum_{k \in \mathcal{s}} G_k(w_k, d_k, c_k) \ ,$$

subject to constraints

$$\sum_{k \in \mathcal{s}} w_k \mathbf{x}_k = \sum_{k \in \mathcal{U}} \mathbf{x}_k = \tau_x$$

where $\mathbf{x}_k = (x_{k1}, x_{k2}, \ldots, x_{kQ})^\top$ is a vector of $q$ auxiliary variables for element $k$. $G_k$ is a measure of distance between $w_k$ and $d_k$ and $c_k$ is a factor that can be freely chosen for additional flexibility.

# Weights Calibration II

To calculate the weights the $\mathbf{x}_k$'s are only needed for the elements in the net sample (i.e. typically only for the respondents), but $\tau_x$, their population totals need to be know.

The auxiliary variables can be metric (e.g. income or age) or categorical (e.g. gender or age groups).

Depending on the choice of $G_k$ different calibration estimators can be obtained, some of the most common are:

- Post-stratification Estimator
- Raking Estimator
- Generalized Regression Estimator

Note that the $w_k$'s typically depend on the sample $s$, in contrast to the $d_k$, which are given by the sampling design.

Post-stratification is typically used if only categorical auxiliary variables are available. It is implemented by forming weighting cells by crossing *all* categories of the auxiliary variables. These weighting cells are the post-strata $\mathcal{U}_q$ with $q = 1, \ldots, Q$. The weight are then adjusted to replicate the counts in these cells. For $k \in \mathcal{U}_q$ we have

$$g_k = \frac{\tau_{x_q}}{\hat{\tau}_{x_q}} \ ,$$

where $\tau_{x_q} = \sum_{k \in \mathcal{U}} x_{kq}$ and

$$x_{kq} = \begin{cases} 1 & \text{if } k \in \mathcal{U}_q \\ 0 & \text{else} \end{cases} .$$

$\hat{\tau}_{x_q \pi} = \sum_{k \in \mathcal{S}} d_k x_{kq}$ its estimator for $\tau_{x_q}$ based on the design weights. The auxiliary variables are the post-stratum indicators, i.e. $\mathbf{x}_k = (x_{k1}, x_{k2}, \ldots, x_{kQ})^\top$. An adjustment to the totals of a metric variable within the post-strata would also be possible.

Table: Population Counts $\tau_{Xq}$ for Hair and Eye Colour

| | Brown | Blue | Hazel | Green |
|---|---|---|---|---|
| Black | 68 | 20 | 15 | 5 |
| Brown | 119 | 84 | 54 | 29 |
| Red | 26 | 17 | 14 | 14 |
| Blond | 7 | 94 | 10 | 16 |

Table: Population Counts $\tau_{x_q}$ for Hair and Eye Colour

| | Brown | Blue | Hazel | Green |
|---|---|---|---|---|
| Black | 68 | 20 | 15 | 5 |
| Brown | 119 | 84 | 54 | 29 |
| Red | 26 | 17 | 14 | 14 |
| Blond | 7 | 94 | 10 | 16 |

Table: Sample counts $\sum_{k \in \mathcal{S}} x_{kq}$ in a SRS with $n = 150$

| | Brown | Blue | Hazel | Green |
|---|---|---|---|---|
| Black | 14 | 7 | 2 | 2 |
| Brown | 36 | 22 | 17 | 5 |
| Red | 7 | 3 | 1 | 4 |
| Blond | 1 | 23 | 1 | 5 |

Table: Population Counts $\tau_{x_q}$ for Hair and Eye Colour

| | Brown | Blue | Hazel | Green |
|---|---|---|---|---|
| Black | 68 | 20 | 15 | 5 |
| Brown | 119 | 84 | 54 | 29 |
| Red | 26 | 17 | 14 | 14 |
| Blond | 7 | 94 | 10 | 16 |

Table: Estimated totals $\hat{\tau}_{x_q\,\pi} = \sum_{k \in \mathcal{S}} x_{kq} d_k$

| | Brown | Blue | Hazel | Green |
|---|---|---|---|---|
| Black | 55.2533 | 27.6267 | 7.8933 | 7.8933 |
| Brown | 142.0800 | 86.8267 | 67.0933 | 19.7333 |
| Red | 27.6267 | 11.8400 | 3.9467 | 15.7867 |
| Blond | 3.9467 | 90.7733 | 3.9467 | 19.7333 |

Table: Population Counts $\tau_{x_q}$ for Hair and Eye Colour

| | Brown | Blue | Hazel | Green |
|---|---|---|---|---|
| Black | 68 | 20 | 15 | 5 |
| Brown | 119 | 84 | 54 | 29 |
| Red | 26 | 17 | 14 | 14 |
| Blond | 7 | 94 | 10 | 16 |

Table: Post-stratification $g_k = \frac{\tau_{x_q}}{\hat{\tau}_{x_q}}$

| | Brown | Blue | Hazel | Green |
|---|---|---|---|---|
| Black | 1.2307 | 0.7239 | 1.9003 | 0.6334 |
| Brown | 0.8376 | 0.9674 | 0.8048 | 1.4696 |
| Red | 0.9411 | 1.4358 | 3.5473 | 0.8868 |
| Blond | 1.7736 | 1.0355 | 2.5338 | 0.8108 |

Beware, there must be at least one element in the sample from each post-stratum, otherwise we divide by null!

In raking only the marginal totals are need, *not* the totals for all the cross-categories. Raking can be implemented as iterative post-stratification to adjust the design weights to the margins of the different auxiliary variables.

The design weights of a SRSC cluster sample of school districts are raked to variables school type (`stype`) and the accomplishment of the growth target (`sch.wide`).

```
##                   dname            name stype sch.wide
## 1 Alameda City Unified    Alameda High     H      Yes
## 2 Alameda City Unified    Encinal High     H      Yes
## 3 Alameda City Unified  Chipman Middle     M      Yes
## 4 Alameda City Unified Lum (Donald D.)     E      Yes
## 5 Alameda City Unified Edison Elementa     E      Yes
## 6 Alameda City Unified Otis (Frank) El     E      Yes
```

TABLE: Population Counts $\tau_{x_q}$ for School Type (`stype`) and School Target (`sch.wide`)

| | No | Yes | SUM |
|---|---|---|---|
| E | 472 | 3949 | 4421 |
| H | 334 | 421 | 755 |
| M | 266 | 752 | 1018 |
| SUM | 1072 | 5122 | 6194 |

# Raking

```r
data(api)
set.seed(-57844)
#selection the SRCS
apiclus <- apipop[apipop$dnum%in%sample(unique(apipop$dnum),10),]
apiclus$fpc <- length(unique(apipop$dnum))

dclus1<- svydesign(id=~dnum, data=apiclus, fpc=~fpc)
#initial weight
w1    <- weights(dclus1)
#convergence is declared if the maximum change in a
#table entry is less than 'eps' ...
eps   <- 1
#... otherwise the process stops after 'maxit' iterations
maxit <- 100

tau_stype    <- table(apipop$stype)
tau_sch.wide <- table(apipop$sch.wide)

#Raking (i.e. iterative post-stratification) for two variables
tab_x <- tab_y <- list()
```

# Raking

```
for (i in 1:maxit) {
    ## Post-stratification to the first variable
    w1 <- split(w1, apiclus$stype)
    adj1 <- tau_stype/sapply(w1, sum)
    # new weight
    w1. <- w1 <- mapply(function(x, y) x * y, w1, adj1)
    # return to original order
    w1 <- unlist(w1.)
    names(w1) <- unlist(sapply(w1., names))
    w1 <- w1[as.character(sort(as.numeric(names(w1))))]
    tab_x[[i]] <- tapply(w1, list(apiclus$stype, apiclus$sch.wide), sum)

    ## Post-stratification to the second variable
    w2 <- split(w1, apiclus$sch.wide)
    adj2 <- tau_sch.wide/sapply(w2, sum)
    # new weight
    w2. <- w2 <- mapply(function(x, y) x * y, w2, adj2)
    # return to original order
    w2 <- unlist(w2.)
    names(w2) <- unlist(sapply(w2., names))
    w2 <- w2[as.character(sort(as.numeric(names(w2))))]
    tab_y[[i]] <- tapply(w2, list(apiclus$stype, apiclus$sch.wide), sum)

    if (i > 1) {
        tab.diff <- abs(tab_y[[i - 1]] - tab_y[[i]])

        if (max(tab.diff) < eps)
            break
    }
    w1 <- w2

}
```

TABLE: Estimated Totals $\hat{\tau}_{x_q\,\pi} = \sum_{k \in \mathcal{S}} x_{kq} d_k$ from a SRCS of Districts (`dname`) with $n_l = 10$

| | No | Yes | SUM |
|---|---|---|---|
| E | 984.1 | 4087.8 | 5071.9 |
| H | 378.5 | 302.8 | 681.3 |
| M | 378.5 | 832.7 | 1211.2 |
| SUM | 1741.1 | 5223.3 | 6964.4 |

TABLE: Estimated Totals after Adjustment to 'stype' in the 1 Interation

| | No | Yes | SUM |
|---|---|---|---|
| E | 857.8 | 3563.2 | 4421.0 |
| H | 419.4 | 335.6 | 755.0 |
| M | 318.1 | 699.9 | 1018.0 |
| SUM | 1595.4 | 4598.6 | 6194.0 |

TABLE: Estimated Totals after Adjustment to 'sch.wide' in the 1 Interation

| | No | Yes | SUM |
|---|---|---|---|
| E | 576.4 | 3968.7 | 4545.1 |
| H | 281.8 | 373.7 | 655.6 |
| M | 213.8 | 779.5 | 993.3 |
| SUM | 1072.0 | 5122.0 | 6194.0 |

TABLE: Estimated Totals after Adjustment to 'stype' in the 2 Interation

| | No | Yes | SUM |
|---|---|---|---|
| E | 560.7 | 3860.3 | 4421.0 |
| H | 324.6 | 430.4 | 755.0 |
| M | 219.1 | 798.9 | 1018.0 |
| SUM | 1104.3 | 5089.7 | 6194.0 |

TABLE: Estimated Totals after Adjustment to 'sch.wide' in the 2 Interation

| | No | Yes | SUM |
|---|---|---|---|
| E | 544.2 | 3884.9 | 4429.1 |
| H | 315.1 | 433.2 | 748.2 |
| M | 212.7 | 804.0 | 1016.7 |
| SUM | 1072.0 | 5122.0 | 6194.0 |

TABLE: Estimated Totals after Adjustment to 'stype' in the 3 Interation

| | No | Yes | SUM |
|---|---|---|---|
| E | 543.3 | 3877.7 | 4421.0 |
| H | 317.9 | 437.1 | 755.0 |
| M | 212.9 | 805.1 | 1018.0 |
| SUM | 1074.1 | 5119.9 | 6194.0 |

TABLE: Estimated Totals after Adjustment to 'sch.wide' in the 3 Interation

| | No | Yes | SUM |
|---|---|---|---|
| E | 542.2 | 3879.4 | 4421.5 |
| H | 317.3 | 437.3 | 754.6 |
| M | 212.5 | 805.4 | 1017.9 |
| SUM | 1072.0 | 5122.0 | 6194.0 |

TABLE: Estimated Totals after Adjustment to 'stype' in the 4 Interation

| | No | Yes | SUM |
|---|---|---|---|
| E | 542.1 | 3878.9 | 4421.0 |
| H | 317.5 | 437.5 | 755.0 |
| M | 212.5 | 805.5 | 1018.0 |
| SUM | 1072.1 | 5121.9 | 6194.0 |

TABLE: Estimated Totals after Adjustment to 'sch.wide' in the 4 Interation

| | No | Yes | SUM |
|---|---|---|---|
| E | 542.0 | 3879.0 | 4421.0 |
| H | 317.4 | 437.5 | 755.0 |
| M | 212.5 | 805.5 | 1018.0 |
| SUM | 1072.0 | 5122.0 | 6194.0 |

# Raking with the survey Package

```
dclus1r <- rake( dclus1, list(~stype, ~sch.wide)
                ,list( table(stype=apipop$stype)
                      ,table(sch.wide=apipop$sch.wide)
                ))
svytable(~stype+sch.wide, dclus1r , round=TRUE)

##      sch.wide
## stype   No  Yes
##     E  542 3879
##     H  317  438
##     M  213  805

(w1/weights(dclus1r))[1:10]

##       863      1138      1139      1140      1141      1142      1143
## 0.9999724 1.0001319 0.9999724 0.9999724 0.9999724 0.9999724 0.9999724
##      1144      1145      1146
## 0.9999724 0.9999724 0.9999724

summary(w1/weights(dclus1r))

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1
```

For the linear generalized regression estimator (GREG) the measure of distance $G_k$ is

$$G_k(w, \pi, c) = G(w_k, d_k, c_k) = \frac{(w_k - d_k)^2}{2 d_k c_k} \ ,$$

and we have

$$\hat{\tau}_{\text{GREG}} = \hat{\tau}_\pi + (\boldsymbol{\tau}_x - \hat{\boldsymbol{\tau}}_{x\pi})^\top \hat{\boldsymbol{\beta}},$$

where

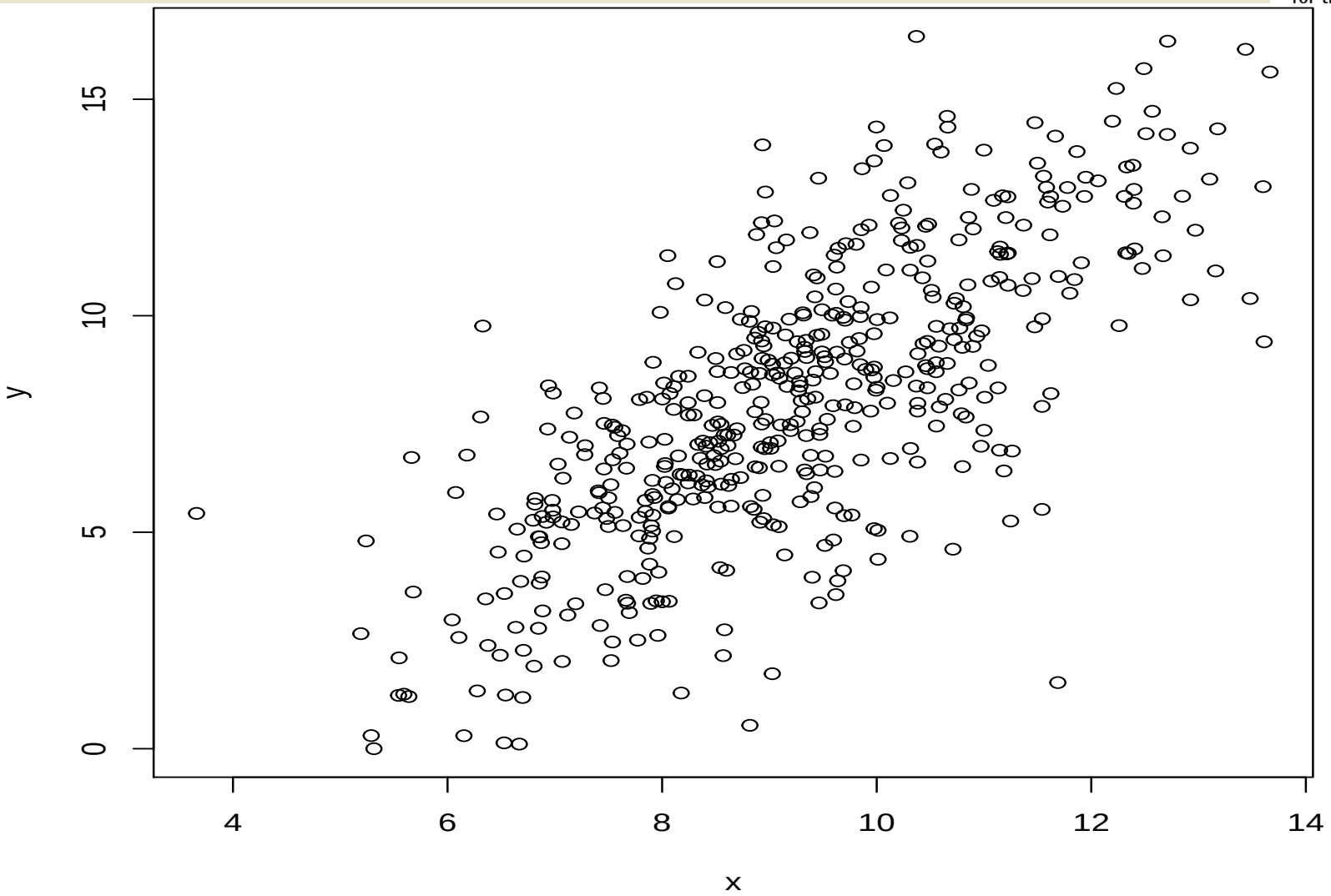
$$\hat{\boldsymbol{\beta}} = \left( \sum_{k \in s} d_k c_k \mathbf{x}_k (\mathbf{x}_k)^\top \right)^{-1} \sum_{k \in s} d_k c_k \mathbf{x}_k y_k \ ,$$

and $\hat{\boldsymbol{\tau}}_{x\pi} = (\hat{\tau}_{x_1 \pi}, \ldots, \hat{\tau}_{x_Q \pi})^\top$.
The adjustment to the design weight $g_k$ can be written as:

$$g_k = 1 + \left( \left( \sum_{k \in \mathcal{U}} \mathbf{x}_k - \sum_{k \in s} d_k \mathbf{x}_k \right)^\top \left( \sum_{k \in s} d_k c_k \mathbf{x}_k (\mathbf{x}_k)^\top \right)^{-1} \right)^\top c_k \mathbf{x}_k$$

# Graphical presentation of $\pi$- and GREG Estimator

gesis
Leibniz Institute
for the Social Sciences



# Generalized Regression Estimator



We want to estimate total expenditures of hospitals. To improve a possible estimate we use data from survey in 1998 to explore if there are any useful predictors for our variable of interest.

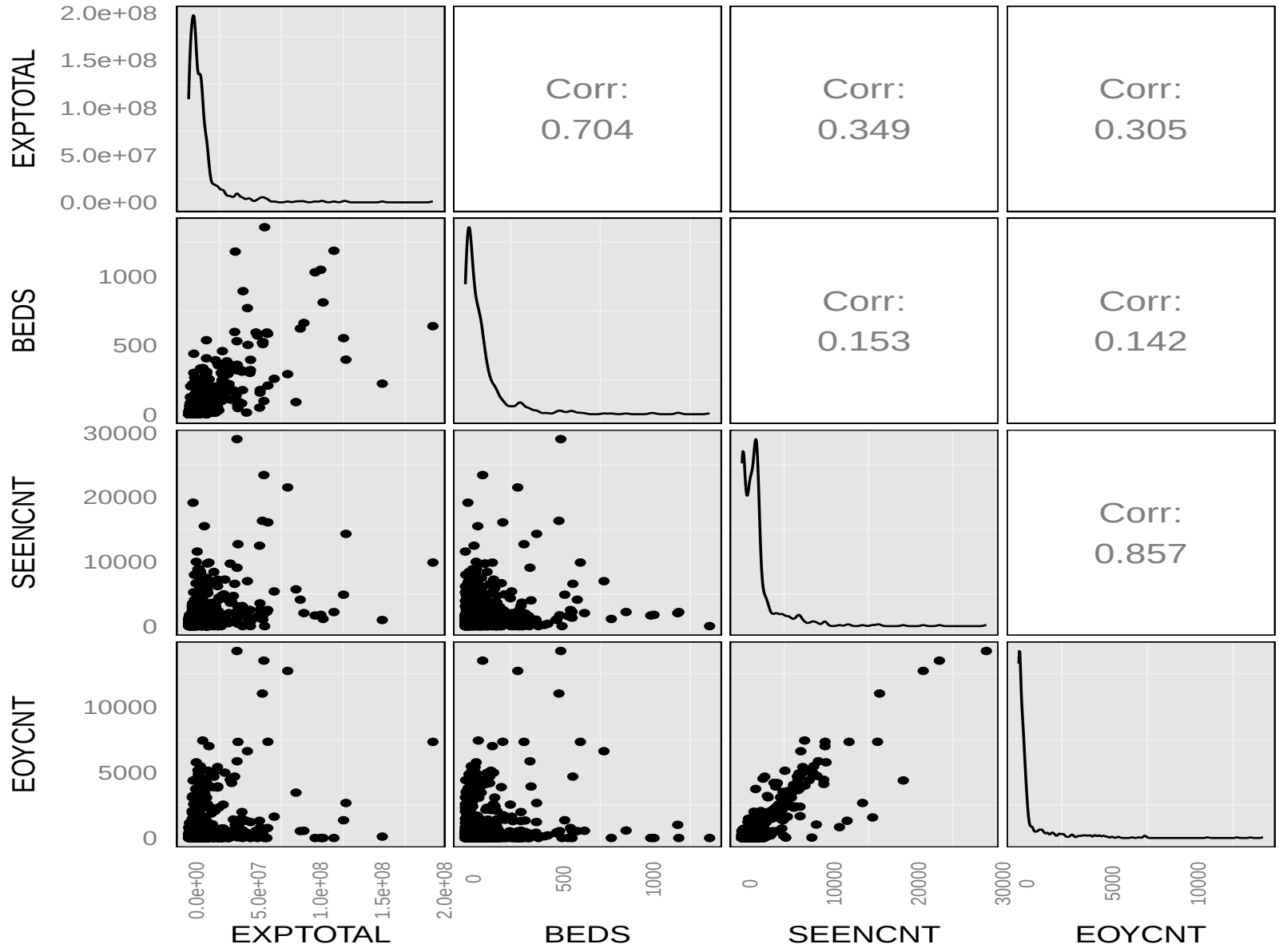
```
library(PracTools) #load the package
data(smho.N874)    #load the data set
head(smho.N874)

##   EXPTOTAL BEDS SEENCNT EOYCNT FINDIRCT hosp.type
## 1  9066430   81    1791    184        2         1
## 2  9853392   80    1870    244        2         1
## 3  3906074   26    1273      0        2         1
## 4  9853392   90    1781    154        2         1
## 5  9853392   71    1839    206        2         1
## 6  9853392   81    1823    196        2         1

#?smho.N874          #for a description of the variables

#only hospitals other than 'type 4' are considered
smho <- smho.N874[smho.N874$hosp.type != 4, ]
```

# Generalized Regression Estimator

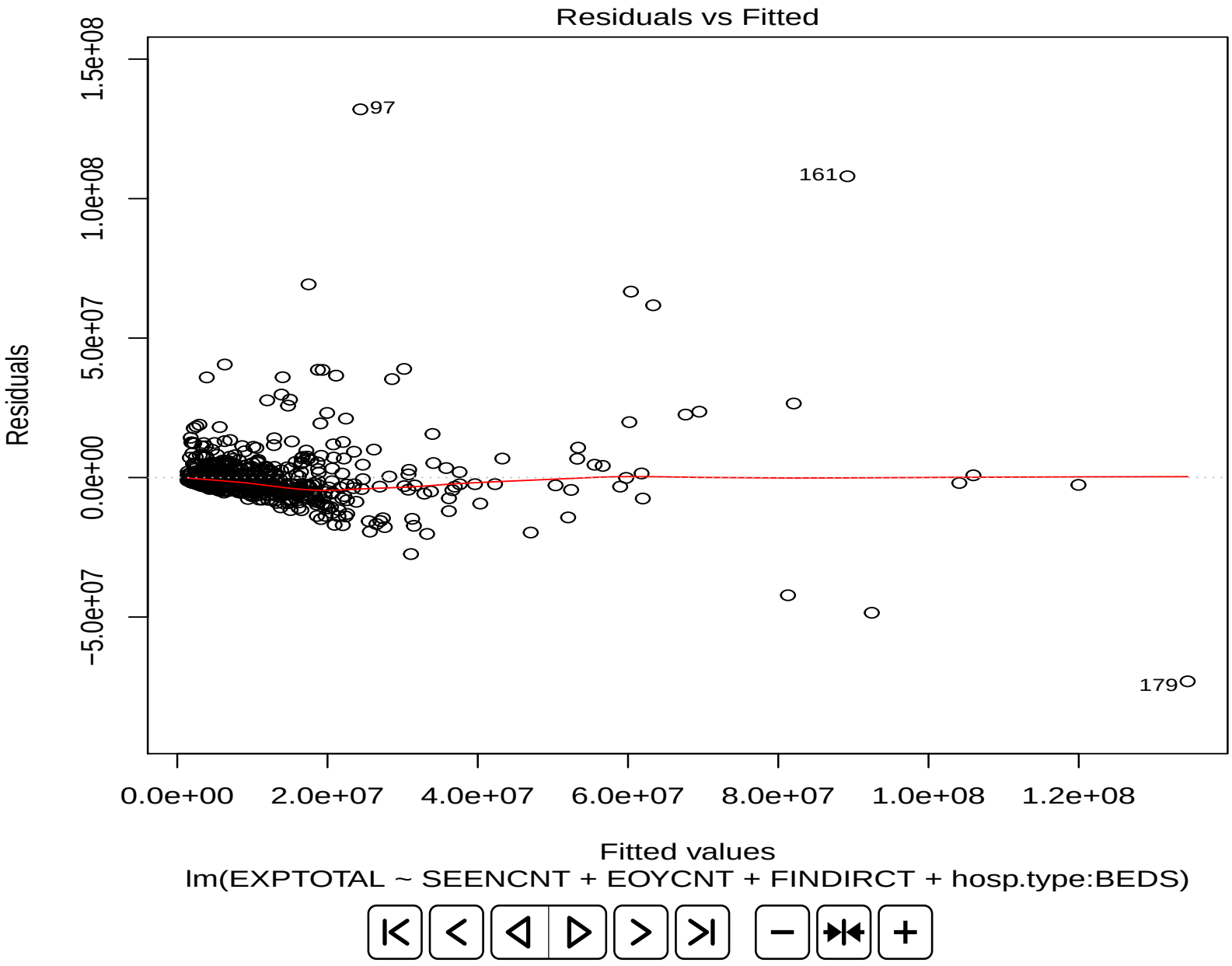


Fitting a linear model for EXPTOTAL with common slopes for SEENCNT and EOYCNT but a different slope for BEDS in each hospital type.

TABLE: Model Summary

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 1318589.11 | 912432.21 | 1.45 | 0.15 |
| SEENCNT | 1033.94 | 310.63 | 3.33 | 0.00 |
| EOYCNT | 2036.15 | 603.58 | 3.37 | 0.00 |
| FINDIRCT2 | 78026.06 | 965237.62 | 0.08 | 0.94 |
| hosp.type1:BEDS | 98139.28 | 3318.84 | 29.57 | 0.00 |
| hosp.type2:BEDS | 39489.35 | 5644.51 | 7.00 | 0.00 |
| hosp.type3:BEDS | 77578.37 | 15082.20 | 5.14 | 0.00 |
| hosp.type5:BEDS | 36855.78 | 8650.48 | 4.26 | 0.00 |

# Generalized Regression Estimator



# Generalized Regression Estimator

We select a sample of hospitals with probability proportional to the square root of BEDS using a systematic sample.

```
#######################################
### Select a pps to sqrt(BEDS) sample
#######################################
library(sampling)   #load the 'sample' package
                    #for the 'UPsystematic' function
smho. <-    # before sampling order the data set by hospital type
  smho.[order(smho.$hosp.type),]

x <- smho.[,"BEDS"]
x[x <= 5] <- 5      # recode small hospitals to have a minimum size
x <- sqrt(x)

n <- 80             #sample size
IP  <- n*x/sum(x)

set.seed(428274453)
sam <- UPsystematic(IP)

sam.dat <- smho.[sam==1, ]
sam.dat$d <- 1/IP[sam==1] #the design weight
```

# Generalized Regression Estimator

Leibniz Institute for the Social Sciences

Now we use the `survey` package to calibrate the weights.

```
library(survey) #load the 'survey' package
#1. build a 'design' object
sam.dsgn <-
  svydesign(ids = ~1,                  # no clusters
            strata = NULL,             # no strata
            data = sam.dat,            # the sample data
            weights = ~d)              # the design weight
    #the model we use for the GREG
lmod2 <- lm(EXPTOTAL ~ SEENCNT + EOYCNT + hosp.type:BEDS, data=smho.)
#2. compute pop totals of auxiliaries
pop.tots <- colSums(model.matrix(lmod2)) #Inefficient but convenient!

#3. use 'calibrate' to compute the new weights
sam.cal <-
  calibrate(design = sam.dsgn,
            formula = ~ SEENCNT + EOYCNT + hosp.type:BEDS,
            population = pop.tots,
            calfun='linear' )
```

Setting argument `calfun='linear'` in `'calibrate'` results in the GREG weights, other calibration function are possible, already built-in are `'raking'` and `'logit'`.

# Generalized Regression Estimator

Leibniz Institute for the Social Sciences

Now we check if the calibration constrains are satisfied:

```
#BEDS by hospital type
svyby(~BEDS, by=~hosp.type, design=sam.cal, FUN=svytotal)

##   hosp.type  BEDS           se
## 1         1 37978 3.951866e-12
## 2         2 13066 1.421532e-12
## 3         3  9573 5.260079e-13
## 5         5 10077 5.811345e-13

#SEENCNT and EOYCNT
svytotal(~SEENCNT+EOYCNT, sam.cal)

##           total SE
## SEENCNT 1349241  0
## EOYCNT   505345  0

pop.tots

##     (Intercept)          SEENCNT          EOYCNT hosp.type1:BEDS
##             725          1349241          505345           37978
## hosp.type2:BEDS hosp.type3:BEDS hosp.type5:BEDS
##           13066            9573            10077
```

Nothing prevents the GREG weights from becoming negative, which is theoretically not a problem, as long as we infer to the population (or sub-populations) to which we calibrated.

Nothing prevents the GREG weights from becoming negative, which is theoretically not a problem, as long as we infer to the population (or sub-populations) to which we calibrated.

However the effects might be catastrophic of domain estimation, in case of estimation domains that where not considered in the calibration.

Nothing prevents the GREG weights from becoming negative, which is theoretically not a problem, as long as we infer to the population (or sub-populations) to which we calibrated.

However the effects might be catastrophic of domain estimation, in case of estimation domains that where not considered in the calibration.

In general it is advisable to only use calibrated weights to infer to the whole population or sub-populations that are found in the marginal totals used for the calibration!

Nothing prevents the GREG weights from becoming negative, which is theoretically not a problem, as long as we infer to the population (or sub-populations) to which we calibrated.

However the effects might be catastrophic of domain estimation, in case of estimation domains that where not considered in the calibration.

In general it is advisable to only use calibrated weights to infer to the whole population or sub-populations that are found in the marginal totals used for the calibration!

Design weights can always be used to do unbiased domain estimation, although the precision of these estimates can be very poor.

# The Variance of the Generalized Regression Estimator

Calibrated weights are **not** independent of the selected sample, i.e. they are random variables. Thus the variance of the GREG estimator cannot be estimated as straightforwardly as for the $\pi$-estimator. We can write its approximate variance as

$$\mathrm{AV}\left(\hat{\tau}_{\mathrm{GREG}}\right) = \sum_{k \in \mathcal{U}} \sum_{l \in \mathcal{U}} \left(\pi_{kl} - \pi_k \pi_l\right) \frac{E_k}{\pi_k} \frac{E_l}{\pi_l} \ ,$$

where $E_k = y_k - y_k^0$, $y_k^0 = \mathbf{x}_x^\top \boldsymbol{\beta}$ and

$$\boldsymbol{\beta} = \left( \sum_{k \in \mathcal{U}} c_k \mathbf{x}_k (\mathbf{x}_k)^\top \right)^{-1} \sum_{k \in \mathcal{U}} c_k \mathbf{x}_k y_k \ .$$

# The Variance of the Generalized Regression Estimator

Calibrated weights are **not** independent of the selected sample, i.e. they are random variables. Thus the variance of the GREG estimator cannot be estimated as straightforwardly as for the $\pi$-estimator. A variance estimator for $\hat{\tau}_{\text{GREG}}$ is given by

$$\hat{\mathrm{V}}\left(\hat{\tau}_{\text{GREG}}\right) = \sum_{k \in s} \sum_{l \in s} \frac{\left(\pi_{kl} - \pi_k \pi_l\right)}{\pi_{kl}} g_k \frac{e_k}{\pi_k} g_l \frac{e_l}{\pi_l} \ ,$$

where $e_k = y_k - \hat{y}_k$ and $\hat{y}_k = \mathbf{x}_x^\top \hat{\boldsymbol{\beta}}$.

# Literature I



R.J.A. Little, D.B. Rubin.
Statistical Analysis with Missing Data.
*Wiley Interscience*, 1999.


D.B. Rubin.
Inference and Missing Data
*Biometrika*, 1976.


D.B. Rubin.
Multiple Imputations for Nonresponse in Surveys
*Wiley*, 1987.