Analyzing Survey Data with Weights – A Practical Introduction

Social Science Data Lab

Stefan Zins [†]

April 30, 2025

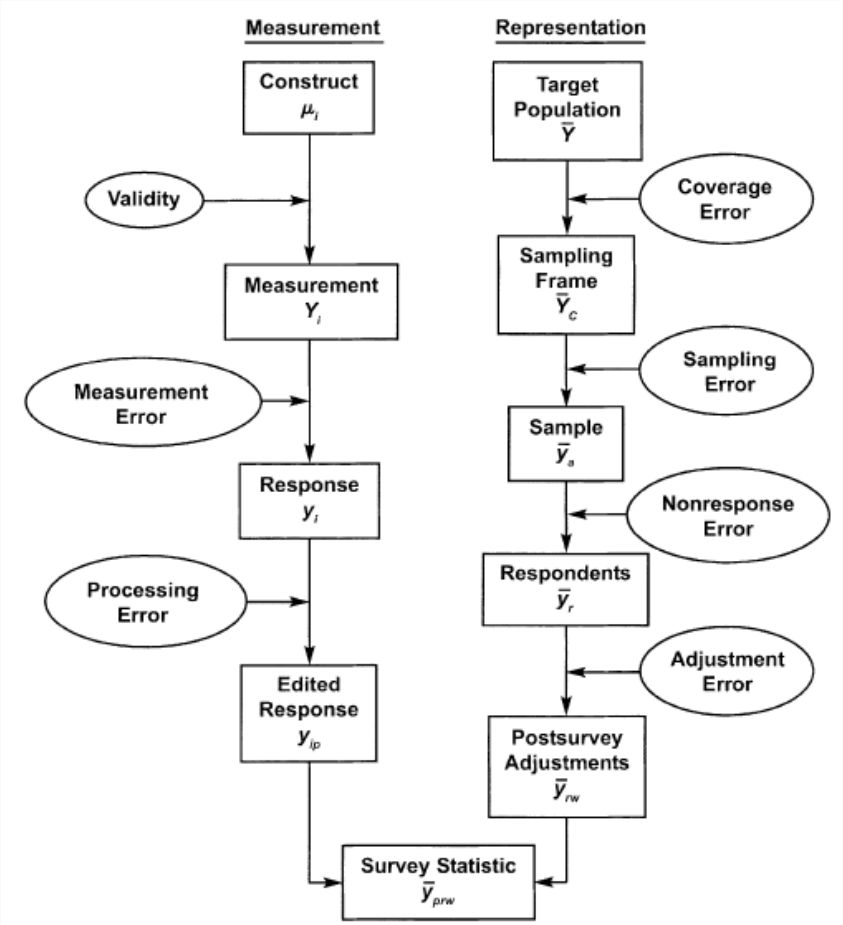
[†] Institute for Employment Research (IAB) of the German Federal Employment Agency

Outline

1. Introduction to Probability Based Sampling
2. Nonresponse
3. The R Survey Package
4. Using Survey Weights - ESS 8
5. Variance Estimation under Nonresponse
6. Summary & Remarks

Introduction to Probability Based Sampling

Total Survey Error Framework



Source: *Survey Methodology*, Groves, Robert M., Floyd J. Fowler Jr, Mick P. Couper, James M. Lepkowski, Eleanor Singer, and Roger Tourangeau. 2009. John Wiley & Sons. Page 48.

Finite Population And Sample

Finite Population of size N :

$$\mathcal{Y} = \{y_1, y_2, \ldots, y_k, \ldots, y_N\}$$

$$\boldsymbol{\mathcal{X}} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k, \ldots, \mathbf{x}_N\}$$

Index of the Population: Sampling Frame

$$\mathcal{U} = \{1, 2, \ldots, k, \ldots, N\}$$

Sample of Size n

$$\mathcal{s} \subset \mathcal{U}$$

Expectation and Variance of a Random Sample (without Replacement)

$$I_k = \begin{cases} 1 & \text{if } k \in s \\ 0 & \text{else} \end{cases}$$

sampling indicator element k

$$\mathsf{E}(I_k) = \pi_k$$

inclusion probability of element k

$$\mathsf{E}(I_k I_l) = \pi_{kl}$$

joint inclusion probability k **and** l

$$\sum_{k \in \mathcal{U}} \pi_k = \mathsf{E}(n)$$

expected sample size

A Generic Design Based Estimator for a Total

$$\tau = \sum_{k \in \mathcal{U}} y_k$$

$$\hat{\tau}_\pi = \sum_{k \in s} \frac{y_k}{\pi_k} = \mathbf{d}^\top \mathbf{y} = \mathbf{I}^\top \check{\mathbf{y}}$$

with $\mathbf{d} = (\pi_1^{-1}, \ldots, \pi_k^{-1}, \ldots, \pi_n^{-1})$, $\mathbf{y} = (y_1, \ldots, y_k, \ldots, y_n)$, $\mathbf{I} = (I_1, \ldots, I_k, \ldots, I_N)$, and $\check{\mathbf{y}} = (y_1/\pi_1, \ldots, y_k/\pi_k, \ldots, y_N/\pi_N)$ which is also known as *Horvitz-Thompson* (HT) or π-estimator.

$$\begin{aligned} \mathsf{E}(\hat{\tau}_\pi) &= \mathsf{E}\left(\sum_{k \in \mathcal{U}} I_k \frac{y_k}{\pi_k}\right) \\ &= \sum_{k \in \mathcal{U}} \mathsf{E}(I_k) \frac{y_k}{\pi_k} \\ &= \sum_{k \in \mathcal{U}} y_k \end{aligned}$$

A Generic Design Based Variance Estimator for a Total

The variance of $\widehat{\tau}_{\pi}$ is

$$\mathsf{V}\left(\widehat{\tau}_{\pi}\right) = \check{\mathbf{y}}^{\top} \Delta \check{\mathbf{y}} \qquad \text{where}$$
$$\Delta = \mathsf{COV}\left(\mathbf{I}, \mathbf{I}\right)$$

A Generic Design Based Variance Estimator for a Total

The variance of $\hat{\tau}_\pi$ is

$$\mathsf{V}\left(\hat{\tau}_\pi\right) = \check{\mathbf{y}}^\top \Delta \check{\mathbf{y}} \qquad \text{where}$$
$$\Delta = \mathsf{COV}\left(\mathbf{I}, \mathbf{I}\right)$$

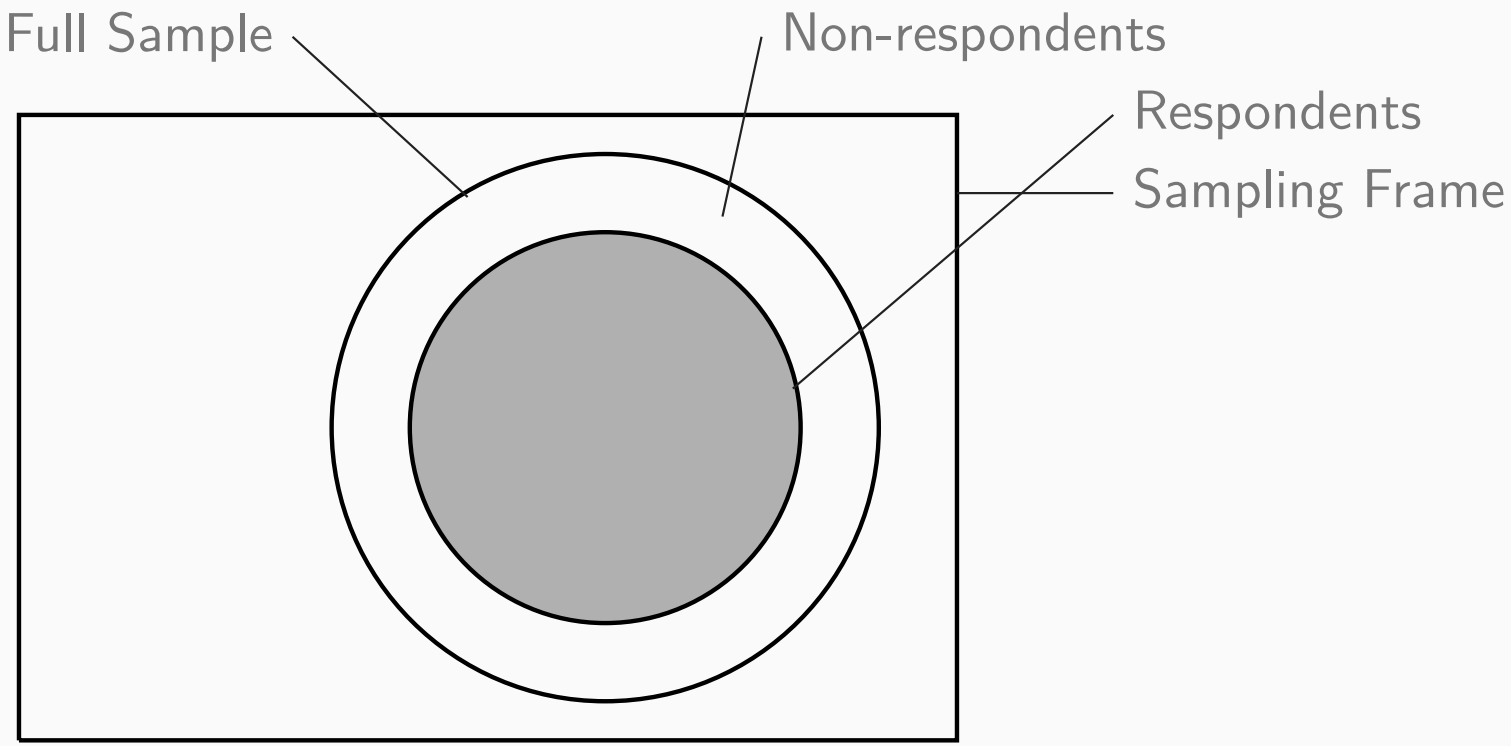
With an unbiased estimate

$$\widehat{\mathsf{V}}\left(\hat{\tau}_\pi\right) = \check{\mathbf{y}}^\top \check{\Delta} \check{\mathbf{y}} \qquad \text{where}$$
$$\check{\Delta} = \left[\frac{\pi_{kl} - \pi_k \pi_l}{\pi_{kl}}\right]_{k,l=1,\ldots,N}$$

Resampling can also be used for variance estimation.

Nonresponse

Survey Imperfections



Nonresponse

The $\mathscr{r}$ is the set of respondents (net sample).

$$\mathscr{r} \subseteq \mathscr{s}$$

$$R_k = \begin{cases} 1 & \text{if } k \in \mathscr{r} \\ 0 & \text{else} \end{cases} \qquad \text{response indicator element } k$$

$$\mathsf{E}(R_k) = \theta_k \qquad \textbf{response probability of element } k$$

$$\mathsf{E}(R_k R_l) = \theta_k \theta_l \qquad \textbf{joint response probability } k \textbf{ and } l$$

Estimation Error Under Nonresponse

The *gross sample* $\mathscr{s}$ is fielded. The set of respondents $\mathscr{r} \subseteq \mathscr{s}$ constitutes the *net sample*, i.e. the net sample is a subsample of the gross sample.

$$\begin{aligned}
\mathsf{E}\left(\hat{\tau}_{\pi}\right) &= \mathsf{E}\left(\sum_{k \in \mathcal{U}} I_k R_k \frac{y_k}{\pi_k}\right) \\
&= \sum_{k \in \mathcal{U}} \mathsf{E}\left(R_k\right) y_k \neq \sum_{k \in \mathcal{U}} y_k \quad (\exists\, k \in \mathcal{U} \text{ with } \theta_k \neq 1)
\end{aligned}$$

$$\hat{\tau}_{w} = \sum_{k \in \mathscr{r}} w_k y_k$$

Estimation Error Under Nonresponse

The *gross sample* $\mathcal{s}$ is fielded. The set of respondents $\mathcal{r} \subseteq \mathcal{s}$ constitutes the *net sample*, i.e. the net sample is a subsample of the gross sample.

$$\begin{aligned}
\mathsf{E}\left(\hat{\tau}_{\pi}\right) &= \mathsf{E}\left(\sum_{k \in \mathcal{U}} I_k R_k \frac{y_k}{\pi_k}\right) \\
&= \sum_{k \in \mathcal{U}} \mathsf{E}\left(R_k\right) y_k \neq \sum_{k \in \mathcal{U}} y_k \quad (\exists\, k \in \mathcal{U} \text{ with } \theta_k \neq 1)
\end{aligned}$$

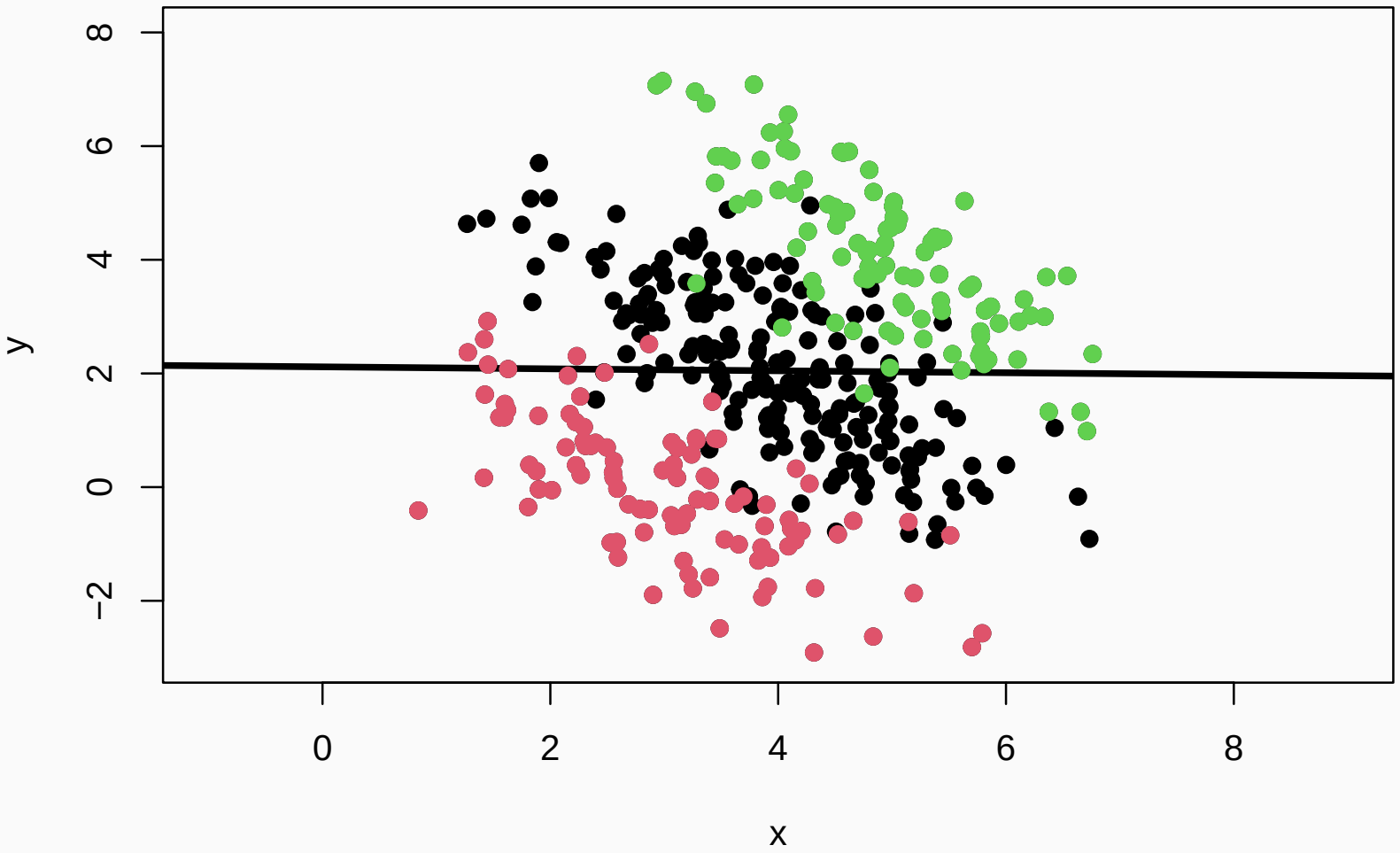
$$\hat{\tau}_{w} = \sum_{k \in \mathcal{r}} w_k y_k$$

Ideally $w_k = \pi_k^{-1}\theta_k^{-1}$.

However we don't know θ_k so we have to approximated or estimated it.

Note: Non-response will always effect the variance of your estimator.

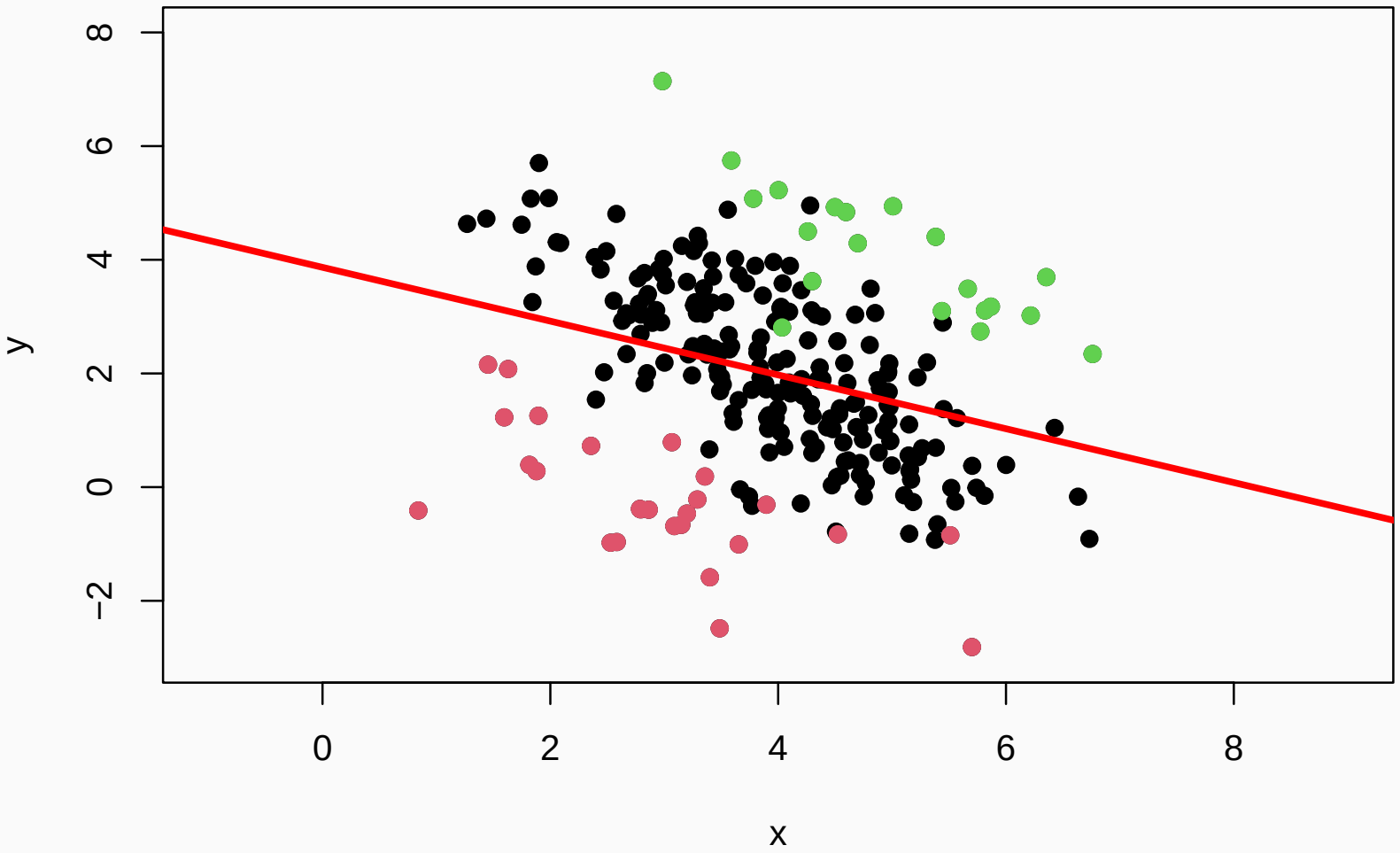
Example of Non-Response Bias



Example of Non-Response Bias

```
##
## Call:
## lm(formula = y ~ x, data = gross)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9350 -1.4739 -0.0179  1.4755  5.0778
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.11618    0.35716   5.925 6.76e-09 ***
## x           -0.01734    0.08596  -0.202     0.84
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.044 on 398 degrees of freedom
## Multiple R-squared:  0.0001022,^^IAdjusted R-squared:  -0.00241
## F-statistic: 0.04069 on 1 and 398 DF,  p-value: 0.8402
```

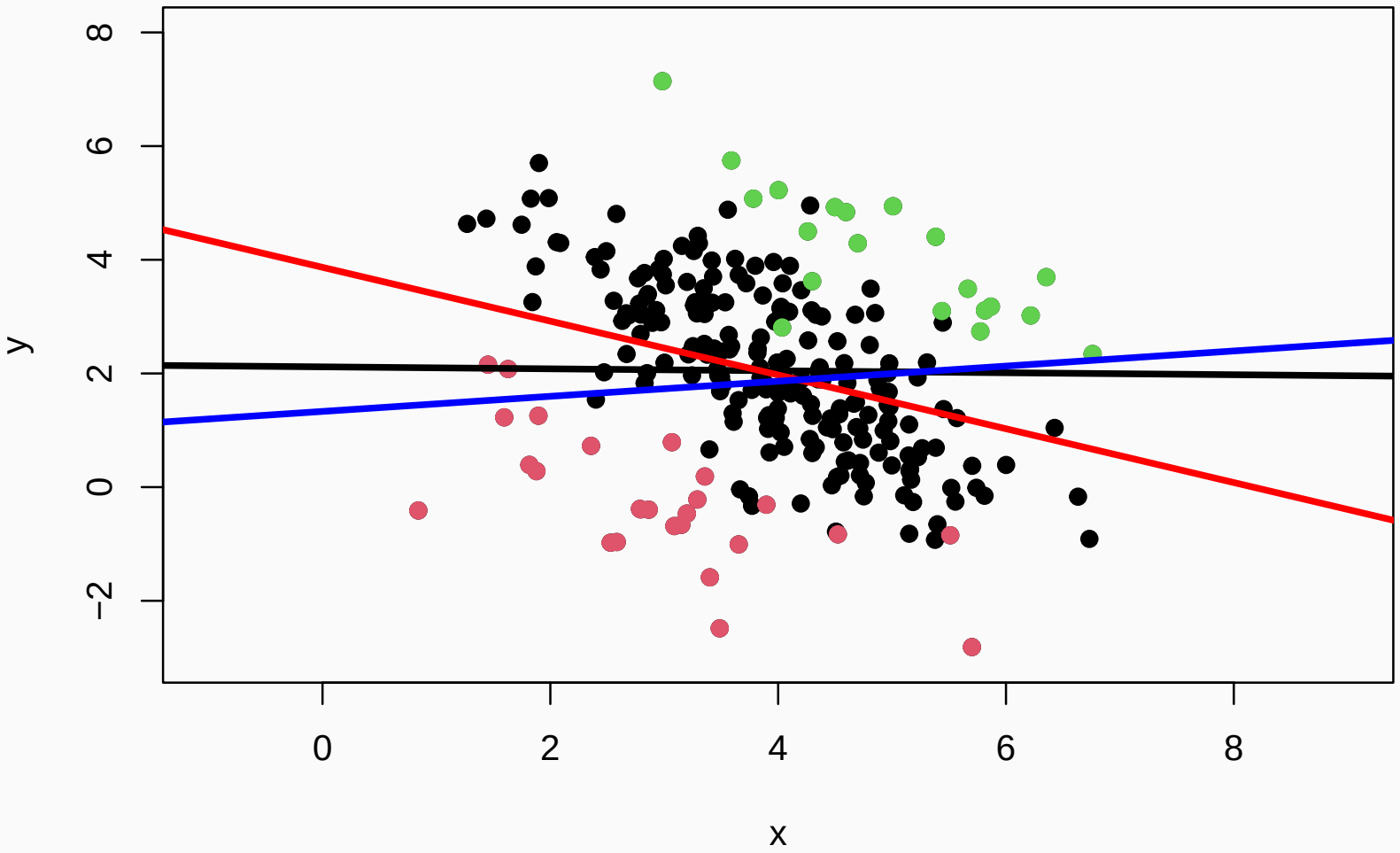
Example of Non-Response Bias



Example of Non-Response Bias

```
##
## Call:
## lm(formula = y ~ x, data = net)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7007 -0.9991  0.0387  1.1773  4.6878
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.86706    0.37863  10.213  < 2e-16 ***
## x           -0.47333    0.09235  -5.125 6.04e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.602 on 244 degrees of freedom
## Multiple R-squared:  0.0972,^^IAdjusted R-squared:  0.0935
## F-statistic: 26.27 on 1 and 244 DF,  p-value: 6.039e-07
```

Example of Non-Response Bias



Example of Non-Response Bias

```
##
## Call:
## svyglm(formula = y ~ x, design = svyd_null)
##
## Survey design:
## svydesign(id = ~1, weights = ~w, data = net)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.3315     0.5563   2.393   0.0174 *
## x             0.1330     0.1334   0.997   0.3199
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 4.204757)
##
## Number of Fisher Scoring iterations: 2
```

Non-Probability Sample

Sample $\mathcal{s}$ has been selected under an unknown sampling design. We might be able to describe a sampling algorithm but we lack the information to compute inclusion probabilities (of any order).

The inference problem can be addressed in a similar way as estimation under nonresponse, only that we now have to model the inclusion probability with the help of auxiliary information. (Chen, Yilin, Pengfei Li, and Changbao Wu. 2019. *Doubly Robust Inference With Nonprobability Survey Samples.*)

> The selection indicator I_k and the response variable y_k are independent given covariates $\mathbf{x}_k$.
>
> All units have a nonzero propensity score, that is, $E(I_k) > 0$ for all $k \in \mathcal{U}$.
>
> The indicator variables I_k and I_l are independent given $\mathbf{x}_k$ and $\mathbf{x}_l$ for $k \neq l$.

Weighting Methods - Estimate Response Probability

We try to estimate $E(R_k) = \theta_k$ with the help of auxiliary information $\mathbf{x}_k$ that is available for the elements in the gross sample $\mathcal{s}$, regardless of survey response.

We hope to find: $E(R_k|\mathbf{x}_k, y_k) = E(R_k|\mathbf{x}_k)$, $k \in \mathcal{U}$.

Generalized liner models (*probit*, *logit*, *log-log*), have traditionally been used to model nonresponse.

More recently Machine Learning Methods, like Random Forest Regression, have also become popular.

Information on the level of the gross sample is needed.

Weighting Methods - Calibration Weights

Input weights (e.g. π_k^{-1}, or $\pi_k^{-1} * \theta_k^{-1}$) are calibrated to the totals of some auxiliary variables $\boldsymbol{x}$.

We are looking for w_k's that meet the condition
$\sum_{k \in r} w_k \mathbf{x}_k = \sum_{k \in \mathcal{U}} \mathbf{x}_k$.
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 nonresponse bias.

If the used auxiliary variables helps to explain the variable of interest calibrated weights can reduce SE.

Information on the level of the net sample is needed plus aggregated population information.

Weighting Methods - Calibration Weights

Input weights (e.g. π_k^{-1}, or $\pi_k^{-1} * \theta_k^{-1}$) are calibrated to the totals of some auxiliary variables $\boldsymbol{x}$.

We are looking for w_k's that meet the condition
$\sum_{k \in r} w_k \mathbf{x}_k = \sum_{k \in \mathcal{U}} \mathbf{x}_k$.
Sample estimates using the calibrated weights will (exactly) replicated those totals.

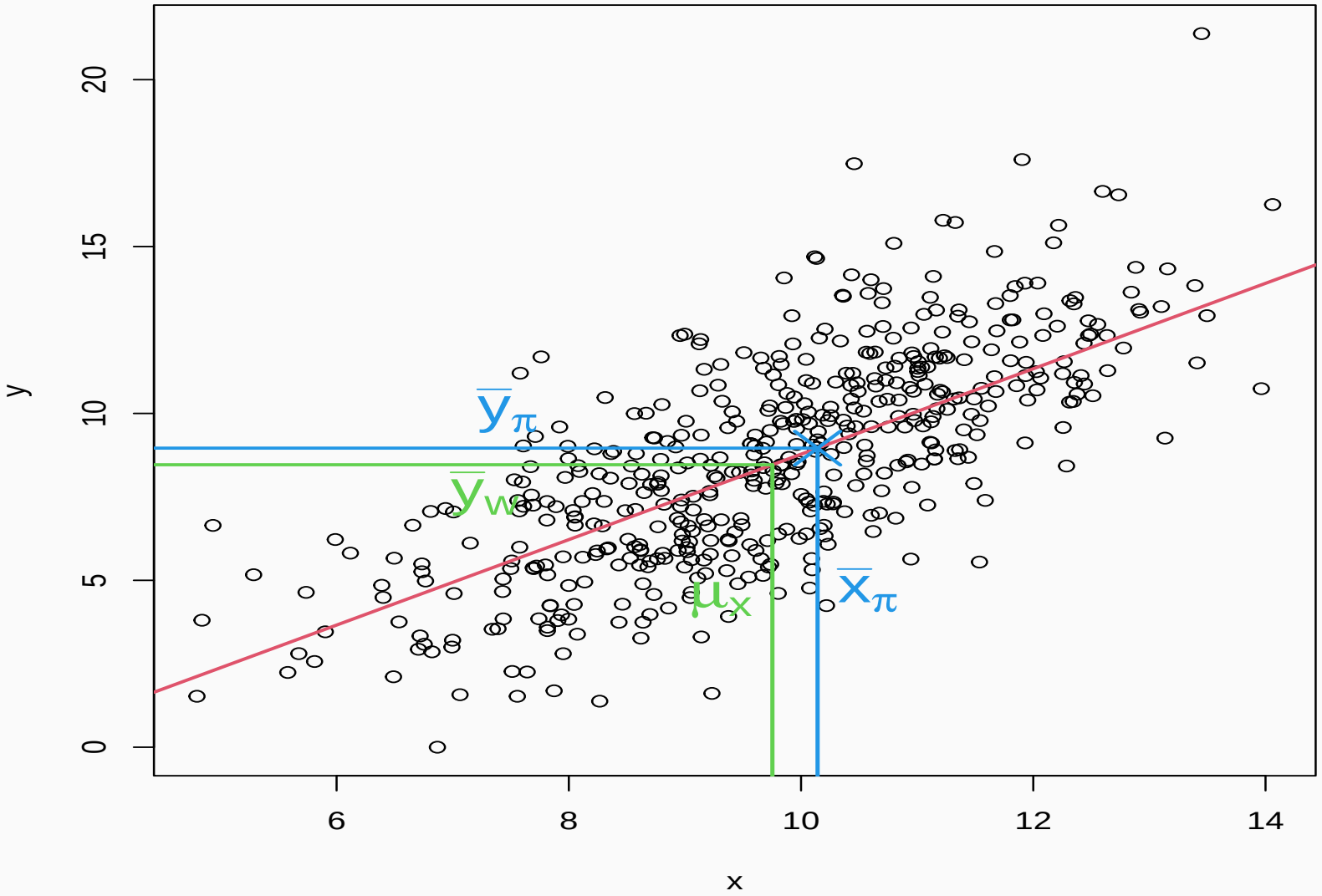
If the used auxiliary variables help to explain the response process the calibrated weight can reduce the nonresponse bias.

If the used auxiliary variables helps to explain the variable of interest calibrated weights can reduce SE.

Information on the level of the net sample is needed plus aggregated population information.

Note: Both methods results in weights that are *random*, i.e. there value depends on the realized net sample!

Graphical presentation Calibration Estimator (GREG)



The R Survey Package

The R Survey Package

- T. Lumley (2024) "survey: analysis of complex survey samples". R package version 4.4.
- T. Lumley (2004) Analysis of complex survey samples. Journal of Statistical Software 9(1): 1-19
- T. Lumley (2010) Complex Surveys: A Guide to Analysis Using R. John Wiley and Sons.
- "Home page" of the Survey package: `https://r-survey.r-forge.r-project.org/survey/`

Defining A Survey Design

Survey designs are specified using the `svydesign` function.

id to specify sampling units

strata to specify strata

weights to specify sampling weights

calibrate.formula model formula specifying how the weights are *already* calibrated

fpc to specify finite population size corrections, e.g. by giving population sizes.

These arguments should be given as formulas, referring to columns in a data frame given as the data argument. Multiple variables on the rhs of the `id`, `strata`, and `fpc` formulas are used to define a multi-stage sampling design.

Using Survey Weights - ESS 8

ESS 8 Data from Austria: Sampling Design

Sampling Frame Address (household) register from the Austrian Postal Service ("data.door").

Sampling Design A two-domain design will be applied. The first domain is formed by Vienna. The 23 districts of Vienna are used as strata. Within these strata 826 households were drawn with equal probabilities. Within each household one target person is selected at random.

All other municipalities form the second domain. The municipalities are stratified according to the 94 NUTS3 regions. A three stage design is applied in this domain. 272 PSUs (Zählsprengel) are selected at the first stage with probability proportional to the number of households on the sampling frame. Within these PSUs 12 households are drawn with equal probabilities (or 8 in some strata). Finally, one target person is selected within each household

ESS8 - 2016 DOCUMENTATION REPORT, THE ESS DATA ARCHIVE, Edition 2.2.

ESS 8 Data from Austria: Weights

The ESS provides two types of survey weights:

1. Design Weights, i.e. $d_k = \pi_k \; \forall \; k \in \mathscr{r}$.
2. So called Post-Stratification Weights, which are a form of Calibration Weights. For Austria in ESS 8 these weights are calibrate to cross- classification of Gender (2), Age (3), Education (3) and NUTS 2 Region (9).

```r
### load packages
library(haven)
library(dplyr)
library(magrittr)
library(tidyr)
library(stringr)
library(survey)

calculate_mode <- function(x) {
  uniqx <- unique(na.omit(x))
  uniqx[which.max(tabulate(match(x, uniqx)))]
}

apistr2st <- readRDS(file = paste0("data/","apistr2st_.rds"))
apistr2st$Nh <- apistr2st$N_h
apistr2st$NI <- apistr2st$N_I

### merge survey data with sampling design data
ess8  <- read_dta("data/ESS8e02_3/ESS8e02_3.dta")
sddf8 <- read_dta("data/ESS8SDDFe01_1/ESS8SDDFe01_1.dta")
ess8  %<>% mutate(cty_idno = paste(cntry,idno,sep="_"))
sddf8 %<>% mutate(cty_idno = paste(cntry,idno,sep="_"))
```

```r
ess8  %<>% left_join(sddf8 %>% select(-c(idno,essround,cntry)), by="cty_idno")

### data editing

ess8 %>%
  filter(cntry == "AT") %>%
  tidyr::replace_na(list(
    edulvlb =  median(.$edulvlb, na.rm = TRUE),
    agea    =  median(.$agea, na.rm = TRUE),
    gndr    =  median(.$gndr, na.rm = TRUE),
    region  =  calculate_mode(.$region)
  )) %>%
  mutate(
    gndr_c  = as.character(gndr),
    age_c   = as.character(cut(
      agea,
      breaks = c(15, 35, 55, Inf),
      include.lowest = TRUE
    )),
    edulvlb_c =
      case_when((edulvlb >= 0   & edulvlb <= 229) | edulvlb > 800  ~ "low",
                (edulvlb >= 311 & edulvlb <= 423) ~ "medium",
```

```
                (edulvlb >= 510 & edulvlb <= 800) ~ "high",
                TRUE ~ as.character(edulvlb) ) ,
  region_c  = as.factor(region),
  gae_c     = as.factor(stringr::str_c(gndr_c, age_c, edulvlb_c)),
  pspwght_p = pspwght * pweight * 10000,
  dweight_p = dweight * pweight * 10000,
  trstplt_c =
    case_when((trstplt >= 0   & trstplt <= 3)  ~ "low",
              (trstplt >= 4   & trstplt <= 7)  ~ "medium",
              (trstplt >= 8   & trstplt <= 10) ~ "high",
              TRUE ~ as.character(trstplt) ),
  stfeco_c =
    case_when((stfeco >= 0   & stfeco <= 3)  ~ "low",
              (stfeco >= 4   & stfeco <= 7)  ~ "medium",
              (stfeco >= 8   & stfeco <= 10) ~ "high",
              TRUE ~ as.character(stfeco) ) ) %>%
  tidyr::replace_na(list(
    trstplt_c =  calculate_mode(.$trstplt_c),
    stfeco_c  =  calculate_mode(.$stfeco_c) ) )  %>%
mutate(psu      = as.factor(psu),
       stratify = as.factor(stringr::str_c(stratum,sep = "_") ))  ->
ess_at
```

ESS 8 Data from Austria: Defining Sampling Design

We only have the indicator for the Primary Sampling Unit (PSU) and no information on the number of PSU within the strata of the first sampling stage (fpc unknown).

We want to use the ESS post-stratification weights (`pspwght`).

For some reason PSU are not nested within strata so we *force* this.

We need to deal with *lonely* PSU within strata for variance estimation.

```
options(survey.lonely.psu = "adjust")

svyd_w_at <-
  svydesign(
    id =  ~ psu + idno,
    strata = ~ stratify,
    weights =  ~ pspwght_p,
    calibrate.formula = ~ gae_c + region_c,
    data = ess_at,
    nest = TRUE
  )
```

ESS 8 Data from Austria: Estimating Proportions

Estimating proportions for `Trust in politicians` and `How satisfied with present state of economy in country`.

```
(A1 <- svymean( ~ trstplt_c,  svyd_w_at))
```

```
##                       mean     SE
## trstplt_chigh    0.072376 0.0075
## trstplt_clow     0.424131 0.0139
## trstplt_cmedium 0.503492 0.0139
```

ESS 8 Data from Austria: Estimating Proportions

Estimating proportions for `Trust in politicians` and `How satisfied with present state of economy in country`.

```
(A2 <- svymean( ~ stfeco_c ,  svyd_w_at))
```

```
##                    mean     SE
## stfeco_chigh   0.17208 0.0111
## stfeco_clow    0.14770 0.0099
## stfeco_cmedium 0.68022 0.0136
```

ESS 8 Data from Austria: Estimating Proportions

Estimating proportions for `Trust in politicians` and `How satisfied with present state of economy in country`.

```
svyttest(I(trstplt_c == "low") ~ I(trstplt_c == "high"), svyd_w_at)
```

```
##
## ^^IDesign-based t-test
##
## data:  I(trstplt_c == "low") ~ I(trstplt_c == "high")
## t = -31.723, df = 572, p-value < 2.2e-16
## alternative hypothesis: true difference in mean is not equal to 0
## 95 percent confidence interval:
##  -0.4855326 -0.4289146
## sample estimates:
## difference in mean
##         -0.4572236
```

ESS 8 Data from Austria: Naive Estimation

We are using the same weights, but we don't specify the sampling design and the calibration weights.

```
svyd_n_at <-
  svydesign(
    id =  ~ 1,
    weights =  ~ pspwght_p,
    data = ess_at
  )
```

ESS 8 Data from Austria: Naive Estimation

	trs_w	stf_w	trs_n	stf_n
chigh	0.07238	0.17208	0.07238	0.17208
clow	0.42413	0.14770	0.42413	0.14770
cmedium	0.50349	0.68022	0.50349	0.68022

Table 1: Point Estmates

	trs_w	stf_w	trs_n	stf_n
chigh	0.00745	0.01105	0.00654	0.01023
clow	0.01388	0.00990	0.01346	0.00969
cmedium	0.01387	0.01358	0.01363	0.01270

Table 2: SE Estimates

ESS 8 Data from Austria: Estimates for Calibration Variabels

```
 svymean(~ region_c,  svyd_n_at)
```

```
##                    mean     SE
## region_cAT11 0.033886 0.0053
## region_cAT12 0.189865 0.0106
## region_cAT13 0.211426 0.0114
## region_cAT21 0.065085 0.0066
## region_cAT22 0.143351 0.0097
## region_cAT31 0.165827 0.0099
## region_cAT32 0.062514 0.0063
## region_cAT33 0.084796 0.0077
## region_cAT34 0.043250 0.0054
```

ESS 8 Data from Austria: Estimates for Calibration Variabels

```
svymean(~ region_c,  svyd_w_at)

##                   mean SE
## region_cAT11 0.033886  0
## region_cAT12 0.189865  0
## region_cAT13 0.211426  0
## region_cAT21 0.065085  0
## region_cAT22 0.143351  0
## region_cAT31 0.165827  0
## region_cAT32 0.062514  0
## region_cAT33 0.084796  0
## region_cAT34 0.043250  0
```

ESS 8 Data from Austria: Measures of Association

```
 svytable(~trstplt_c + stfeco_c, svyd_w_at)

##           stfeco_c
## trstplt_c       high        low     medium
##    high    292379.39   28884.35  217570.14
##    low     316457.14  828543.70 2012614.12
##    medium  672291.66  242161.48 2833995.12

 svychisq(~trstplt_c + stfeco_c, svyd_w_at, statistic="F")

##
## ^^IPearson's X^2: Rao & Scott adjustment
##
## data:  svychisq(~trstplt_c + stfeco_c, svyd_w_at, statistic = "F")
## F = 57.205, ndf = 3.9109, ddf = 2240.9294, p-value < 2.2e-16
```

ESS 8 Data from Austria: A Generelized Linear Regression I

```
summary(svyglm(I(wrkprty==1) ~ wrclmch + factor(vote), design=svyd_w_at,
               family=quasibinomial()))

##
## Call:
## svyglm(formula = I(wrkprty == 1) ~ wrclmch + factor(vote), design = svyd_w_a
##     family = quasibinomial())
##
## Survey design:
## svydesign(id = ~psu + idno, strata = ~stratify, weights = ~pspwght_p,
##     calibrate.formula = ~gae_c + region_c, data = ess_at, nest = TRUE)
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -4.0807     0.5358  -7.616 1.15e-13 ***
## wrclmch         0.5087     0.1542   3.299  0.00103 **
## factor(vote)2  -1.4427     0.6214  -2.322  0.02062 *
## factor(vote)3  -0.3641     0.4612  -0.789  0.43025
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 1.002496)
```

ESS 8 Data from Austria: A Generelized Linear Regression II

```
##
## Call:
## svyglm(formula = I(wrkprty == 1) ~ wrclmch + factor(vote), design = svyd_n_a
##     family = quasibinomial())
##
## Survey design:
## svydesign(id = ~1, weights = ~pspwght_p, data = ess_at)
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -4.0807     0.5635  -7.241 6.41e-13 ***
## wrclmch         0.5087     0.1608   3.163  0.00158 **
## factor(vote)2  -1.4427     0.6334  -2.278  0.02286 *
## factor(vote)3  -0.3641     0.4899  -0.743  0.45752
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 1.052228)
##
## Number of Fisher Scoring iterations: 6
```

Variance Estimation under Nonresponse

Resampling and Calibration

Variance estimation can also be conducted via resampling methods.

1. Select a resample of the *gross* sample.
2. Estimate the response propensities for the resample.
3. Calibrated the product of the design and response propensity weights of the resample and compute your estimator.
4. Repeat 1. - 3. and compute the variance of the replications of your estimates.

Note: Resampling just the *net* sample and recomputing the calibration weights might underestimate the variance.

Example Non-Response Weighing and Variance Estimation 1

The sample was selected by a two-stage sampling design with stratification at the first sampling stage. The stratum sizes, are given by variable `N_h` and PSU sizes are given by `N_I` variable. Variable `R` indicates which school responded (`R == 1`) and which did not (`R == 0`). For the non-responding schools only information the sampling design variables (`dnum`, `snum`, `strata`, `N_h`, `N_I`) and variables `stype` (school type) and `api99` (Academic Performance Index in 1999) is available.

Example Non-Response Weighing and Variance Estimation 2

```
svydGross <- svydesign(id      = ~ dnum + snum,
                       strata = ~ strata,
                       fpc     = ~ N_h + N_I,
                       data    = apistr2st)

theta.hat <-
  predict(svyglm(R ~ api99 + stype, design = svydGross, family = "quasibinomial
          type = "response", newdata = svydGross$variables)

svydGross <-
  update(svydGross,
         dweight =  weights(svydGross),
         pw = 1 / ( svydGross$prob * coef(theta.hat) ) )

svydGross$variables %>% filter( R == 1 ) -> apistr2stnet

svydNet <- svydesign(id      = ~ dnum + snum,
                     strata  = ~ strata,
                     fpc     = ~ N_h + N_I,
                     weights = ~ pw,
                     data    = apistr2stnet)
```

Example Non-Response Weighing and Variance Estimation 3

```
data(api)

cal.fm <-  ~ api99+stype-1
(pop.mar <- colSums(model.matrix( cal.fm, apipop)))

##   api99  stypeE  stypeH  stypeM
## 3914069    4421     755    1018

svydNetCal <-
  survey::calibrate(design = svydNet,
                    formula = cal.fm,
                    population = pop.mar,
                    calfun = "raking")
```

Example Non-Response Weighing and Variance Estimation 4

```
summary( svyglm(api00 ~ mobility + avg.ed + full + sch.wide, svydNetCal ) )

##
## Call:
## svyglm(formula = api00 ~ mobility + avg.ed + full + sch.wide,
##     design = svydNetCal)
##
## Survey design:
## survey::calibrate(design = svydNet, formula = cal.fm, population = pop.mar,
##     calfun = "raking")
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 126.4091    78.6305   1.608  0.11130
## mobility     -0.4555     0.4895  -0.931  0.35449
## avg.ed       98.9580    13.2196   7.486 3.96e-11 ***
## full          2.7859     1.0402   2.678  0.00875 **
## sch.wideYes  49.8237    17.7703   2.804  0.00615 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 6125.053)
```

```r
cal_est   <- function(repw, grossdat, poptotals){

  grossdat$pw    <- repw
  svyd_gross_rep <- svydesign(id        = ~ 1,
                              weights   = ~ pw,
                              data      = grossdat)

  resp_model <-
    svyglm(R ~ api99 + stype, family = "quasibinomial",
           design = svyd_gross_rep)

  svyd_gross_rep$variables %>% filter(R==1) %>%
    mutate( theta.hat = coef(predict(resp_model, newdata = .,
                                     type="response")),
            pw  = 1/theta.hat * pw
    ) ->  netdata

  svyd_net_rep <- svydesign(id  = ~ dnum + snum,
                            weights = ~ pw,
                            data    = netdata)
  svyd_net_cal_ <-
    calibrate(design  = svyd_net_rep,
```

```
                formula = ~ api99+stype,
                population = poptotals,
                calfun = "raking"
                )

  svyglm(api00 ~ mobility + mobility + avg.ed + full + sch.wide,
         svyd_net_cal_)

}

set.seed(20250314)

svydGrossRep <-
  as.svrepdesign(svydGross,
                 type = "mrbbootstrap",
                 replicates = 100)

svyBootNon <- svrepdesign(
  data    = svydGross$variables,
  type    = "bootstrap",
  repweights = svydGrossRep$repweights,
  combined.weights = FALSE,
```

```
  weights     = weights(svydGross)
)

withReplicates(svyBootNon,
               quote(as.vector(coef(
                 cal_est(
                 repw       = .weights,
                 grossdat   = svydGross$variables,
                 poptotals =  colSums(model.matrix(~  api99+stype, apipop)) )
                 ) ) )
               ,  return.replicates = TRUE )

##           theta      SE
## [1,] 126.40912 85.0311
## [2,]  -0.45552  0.7297
## [3,]  98.95804 11.6020
## [4,]   2.78586  1.0452
## [5,]  49.82370 18.4760
```

Summary & Remarks

Summary & Remarks I

- If you analyse the net sample without considering the sampling design and non-response you risk biased results and wrong test decisions. This also applied to measures of association and statistical models, not only so-called descriptive statistics.
- Using survey weights won't *take care* of the sampling design and non-response process, you still need to use the appropriate SE estimators.
- Weights do not necessarily increase the SE, but appropriate methods for variance estimation have to be used.

Summary & Remarks II

- You should worry about SE estimation as much as you do worry about your substantive parameters estimates.
- Get familiar with how your preferred software handles complex sampling designs and survey weights (check out the `R suvery` or `svycal` and `svyset` in `Stata`).
- Learn how the survey weights of the data you analyse have been computed. In particular, if calibration weights have been constructed, (raking, post-stratification) what calibration variables have been used (important for SE estimation!)

Summary & Remarks III

- Using survey weights for non-probability is similar to weighting under non-response, but the assumption made for inference might be harder to justify.
- There are limitation to using survey weights (design-based inference). In general model-based methods (that rely on distribution assumptions) are not made with survey weights in mind (E.g. Maximum Likelihood, Bayesian Methods).

Litreture i


Y. Chen and P. Li and C. Wu.
Doubly Robust Inference With Nonprobability Survey Samples
Journal of the American Statistical Association, 2019


C.E. Särndal and B. Swensson and J. Wertman
Model Assisted Survey Sampling
Springer, 1992


C.E. Särndal and S. Lundström
Estimation in surveys with nonresponse
John Wiley & Sons, 2005


C. Skinner and J. Wakefield
Introduction to the Design and Analysis of Complex Survey Data
Statistical Science, 2017

Litreture ii

D. Pfeffermann

Modelling of complex survey data: Why model? Why is it a problem? How can we approach it?

Survey Methodology, 2011

S. Rubin-Bleuer and I.S. Kratina

On the two-phase framework for joint model and design-based inference

The Annals of Statistics, 2005