Answers exercise non-probability sampling calibration

Peter Lugtig

2023-12-05

The goal of this exercise is to practice (once more) with calibration, raking or imputation. The setting now is a bit different: rather than trying to adjust for unit- or item nonresponse, we are now using these methods to correct for selection bias in a non-probability survey. In principe the methods discussed today can be applied to any data source that one knows is suffering from selection bias. E.g. experiments, social media data, or other sources of Big Data. However you should always be aware that your non-probability dataset should at least exhibit a similar type of variation as you have in the population. For example, it will be impossible to adjust a typical psychological experiment conducted among 18-25 year old students at a Western University, to the general population, because you just have young and highly educated people.

The dataset(s) you will use today are a slightly adapted from data that are publicly available throuh PEW, which is a non-profit survey data collection organisation in the USA. Between June and July 2016, PEW designed a short questionnaire and then asked three organizations that rely on volunteer opt-in survey panels to administer this questionnaire in their panel. This resulted in a dataset of about 30.000 respondents. You are today getting about two/thirds of these respondents to develop an adjustment method. The remaining 10.000 respondents will serve as the hold-out sample against which I will test your adjustment method. I also excluded many variables: your dataset will consist of about 30 variables which you can use for adjustment, and 1 dependent variable called VOTESUM.

The variable VOTESUM asks for the Future voting behavior in the November 2016 Presidential Eelection, with a choice betweeb Clinton, Trump and being undecided. At 1 July, the aggregated polls indicated that Clinton would receive 46% of the vote, and Trump 42%, with the rest being undecided. If I correct for the bias that was present in the polls throughout the 2016 election cycle, my best guess for the true difference between the candidates would have been 45% for Clinton and 43% for Trump.

If you want to know more about the study and data, have a look at: https://www.pewresearch.org/methods/2018/01/26/for-weighting-online-opt-in-samples-what-matters-most/

Lets first load the data.

```
library(foreign)
library(survey)
library(dplyr)
library(plyr)
library(mice)
library(ranger)

nonprob <- readRDS("PEW_nonprob_samples_HOLDOUT.RDS")
# below you see how many cases there are from each vendor
table(nonprob$vendor)
```

Variables

The covariates in these there dataset consist of demographic variables but also a range of other variables for which the people at Pew hoped they would correlate to both selection bias (R) and the dependent variable (Y):

here Vote. They employed a superpopulation approach to assemble additional population level characteristics (more on this later), which were then also included in the survey as questions. Here is the lost of variables:

GENDER # make, female
AGE # age in years EDUCCAT5 # educational level in 5 catagories (lo to hi)
DIVISION # region in 4 categories
MARITAL_ACS # 5 cat marital status
HHSIZECAT #1,2 3+
CHILDRENCAT # 1,2,3_ children at home
CITIZEN_REC # US citizen or not
BORN_ACS # born inside, or outside USA
FAMINC5 #income in5 bracks: <20k, 20-40,40-75,75-150, >150
EMPLOYED # in employment or not
MIL_ACS_REC # never been on active military, has been in military
HOME_ACS_REC # 1. own, 2. rent 3. rent without pay
FDSTMP_CPS # 1.Do you receive foodstamps?
TENURE_ACS # did you live on current address one year ago
PUB_OFF_CPS # have you visited a public official to express your opinion in the past 12 months?
COMGRP_CPS # have you participated in a school, neighboorhood or community associateion in the past year?
TALK_CPS # talk to family 5 cat
TRUST_CPS # do you trust the people in your neighboorhood (5 cat)
TABLET_CPS # do you use a tablet?
TEXTIM_CPS # do yiu every send text messages?
SOCIAL_CPS # active on social media
VOLSUM # volunteering
REGISTERED # registered to vote: yes, no
VOTE14 # voted in 2014 miterm election
PARTYSCALE5 #party attachment 5 categories
RELIGCAT #5 cat religious affiliation: roman catholic, evangelical, main protestant, other, unaffiliated
IDEO3 # liberal, moderate, conservative
OWNGUN_GSS # owns a gun (yes, no)
FOLGOV # do you follow the government

Population data

Pew collected data from the Census bureau, as well as some other Gold-standard administrative sources, and surveys for the eligible voting USA population. They also used some of the surveys to compute (aggregated) correlations between these variables, and used this to create a SuperPopulation dataset of the USA. The dataset inlcudes only 20.000 people, but it is supposed to be representative for the USA on the variables includes in this dataset. The variable names, labels, and codings in this dataset are the exact same as in the non-probability based survey dataset. This is neat, no recoding! Load the data bewlo

```
popdata <- readRDS("PEW_population_data.RDS")
```

What to do?

You have 30 variables to now build a model from. Please note the following:
- It is nice if the covariates predict Y (voting behavior). There is no need to read the literature on voting behavior, but do think about this when you select your adjustment variables.
- If you want to include 30 variables in one go into your model, perhaps with some interactions, you will encounter some issues. First of all, R may become quite slow! Second, there may be estimation issues, because these are perhaps just many variables. Maybe it is worthwhile to first try a model with just a few variables, like in the example below:

How did raking and calibration work again?

```
# poststratification example
# first, specificy the unweighted svydesign object
svy.unweighted <-svydesign(ids=~rid, data=nonprob)
```

```
## Warning in svydesign.default(ids = ~rid, data = nonprob): No weights or
## probabilities supplied, assuming equal probability
```

```
# below, I use gender and whether someone voted in the 2014 midterm election
#gendervote<- as.data.frame(table(nonprob2$GENDER,nonprob2$VOTE14))
gendervote.dist <-xtabs(~GENDER+VOTE14, data=popdata) # the 2x2 table for the population
poststratdesign <- postStratify(design= svy.unweighted, strata =~GENDER+VOTE14,
                                population=gendervote.dist)

# for Raking (easier when you have many variables in your model, but be careful with continuous vars)
gender.dist <- as.data.frame(table(popdata$GENDER))
colnames(gender.dist) <- c("GENDER", "Freq")
vote.dist <- as.data.frame(table(popdata$VOTE14))
colnames(vote.dist) <- c("VOTE14", "Freq")
rakeddesign  <- rake(design=svy.unweighted,sample.margins=list(~GENDER,~VOTE14),
                     population.margins=list(gender.dist,vote.dist))

# add the weights if you want to do diagnostics
nonprob$PSweights <- poststratdesign$prob
nonprob$rakeweights <- rakeddesign$prob

####### and then estimate the outcome
svyby(~VOTESUM,~vendor, design=svy.unweighted,svymean)[2]# p(Trump) across vendors.
```

```
##          VOTESUMTrump
## Vendor 1    0.4430688
## Vendor 2    0.4290268
## Vendor 3    0.4212791
```

```
svyby(~VOTESUM,~vendor, design=poststratdesign,svymean)[2]
```

```
##          VOTESUMTrump
## Vendor 1    0.4394219
## Vendor 2    0.4337107
## Vendor 3    0.4299010
```

```
svyby(~VOTESUM,~vendor, design=rakeddesign,svymean)[2]
```

```
##          VOTESUMTrump
## Vendor 1    0.4383391
## Vendor 2    0.4334508
## Vendor 3    0.4278866
```

Imputation models

Are you curious to try Mass Imputation? Great! I made your life a bit easier by creating an artificial population of the USA, and then match the survey data to this population dataset. So in the dataset below Please note that the population dataset has about 1 million cases, which is only 1/200 of the actual population size. R gets slow however, so just imagine you have the actual population

#{r import population data} #wholepop <- readRDS("complete USA population data with

```
matched survey results.RDS") #
```

Want to see how you design an imputation model? Have a look at the weeks where we talked about imputation.

Ok now what?

The idea is that you now develop your own adjustment model. You may try out different models (raking, poststratification, mass imputation), and also try out a model with different covariates. Please send your R-code of your final model (just 1!) in a plan-text e-mail to Peter before the next class. Please do not change the names of the variables so your code will actually run. he person who has designed the best adjustment model (closest to the poll aggregate on July 1st), which is 45% Clinton/ 43% Trump.

Good luck, and have fun!

Answers 2023:

1. Florian

```r
# create variable to differentiate between popdata and nonprobdata
nonprob$synt <- 1
popdata$synt <- 0

# omit NA because Refused does not exist in popdata
nonprob[nonprob == "Refused"] <- NA
nonprob <- na.omit(nonprob)

#recode  AGE: 18-29, 30-39, 40-49, 50-59, 60-69, 70+
nonprob$AGE[nonprob$AGE >= 60] <- "60+"
nonprob$AGE[nonprob$AGE %in% 18:29] <- "18-29"
nonprob$AGE[nonprob$AGE %in% 30:39] <- "30-39"
nonprob$AGE[nonprob$AGE %in% 40:49] <- "40-49"
nonprob$AGE[nonprob$AGE %in% 50:59] <- "50-59"

nonprob$AGE <- factor(nonprob$AGE, levels = c("18-29", "30-39", "40-49", "50-59", "60+"), labels = c("18

popdata$AGE[popdata$AGE >= 60] <- "60+"
popdata$AGE[popdata$AGE %in% 18:29] <- "18-29"
popdata$AGE[popdata$AGE %in% 30:39] <- "30-39"
popdata$AGE[popdata$AGE %in% 40:49] <- "40-49"
popdata$AGE[popdata$AGE %in% 50:59] <- "50-59"

popdata$AGE <- factor(popdata$AGE, levels = c("18-29", "30-39", "40-49", "50-59", "60+"), labels = c("18

# create standartised inverse propensity score weights
alldata <- nonprob %>%
  select(!c(rid, vendor, VOTESUM, sample, PSweights,rakeweights)) %>%
  rbind(popdata)

set.seed(1337)
forest <- ranger(synt ~ AGE+GENDER+EDUCCAT5+RELIGCAT+PARTYSCALE5,
                 data = alldata,
                 probability = TRUE)
```

```
predictions <- predict(forest, data = alldata)$predictions

rm(forest) # rm model bc it slows R down heavily

alldata$forest1 <- predictions[,1]
alldata$forest0 <- predictions[,2]

alldata$propensityscore <- alldata$forest1/alldata$forest0

alldata$inversepropensityscore <- 1/alldata$propensityscore

survey_weights <- alldata$inversepropensityscore[1:14651]

nonprob$sips <- survey_weights/sum(survey_weights)

ipsdesign <- svydesign(ids=~rid, weights = ~sips, data=nonprob)

# raking

AGE.dist <- as.data.frame(table(popdata$AGE))
colnames(AGE.dist) <- c("AGE", "Freq")

GENDER.dist <- as.data.frame(table(popdata$GENDER))
colnames(GENDER.dist) <- c("GENDER", "Freq")

EDUCCAT5.dist <- as.data.frame(table(popdata$EDUCCAT5))
colnames(EDUCCAT5.dist) <- c("EDUCCAT5", "Freq")

FAMINC5.dist <- as.data.frame(table(popdata$FAMINC5))
colnames(FAMINC5.dist) <- c("FAMINC5", "Freq")

RELIGCAT.dist <- as.data.frame(table(popdata$RELIGCAT))
colnames(RELIGCAT.dist) <- c("RELIGCAT", "Freq")

PARTYSCALE5.dist <- as.data.frame(table(popdata$PARTYSCALE5))
colnames(PARTYSCALE5.dist) <- c("PARTYSCALE5", "Freq")

IDEO3.dist <- as.data.frame(table(popdata$IDEO3))
colnames(IDEO3.dist) <- c("IDEO3", "Freq")

FDSTMP_CPS.dist <- as.data.frame(table(popdata$FDSTMP_CPS))
colnames(FDSTMP_CPS.dist) <- c("FDSTMP_CPS", "Freq")

combdesign  <- rake(design = ipsdesign, sample.margins = list(~AGE, ~GENDER, ~EDUCCAT5, ~FAMINC5,
                                                              ~RELIGCAT, ~PARTYSCALE5, ~IDEO3, ~
                    population.margins = list(AGE.dist, GENDER.dist, EDUCCAT5.dist, FAMINC5.dist,
                                              RELIGCAT.dist, PARTYSCALE5.dist, IDEO3.dist, FDSTMP_CPS
# get the projected voting distribution
Florian <- svyby(~VOTESUM,~vendor, design=combdesign,svymean)
Florian$name <- "Florian"
```

#3. My best solution

See also the Pew report First, do diagnostics, and find variables that predict both Y and R

```r
# for Raking (easier when you have many variables in your model, but be careful with continuous vars)
# what predics vote (Y)
nonprob2 <- nonprob[nonprob$VOTESUM!="Undecided",]
regression <- glm(VOTESUM~GENDER+AGE+EDUCCAT5+DIVISION+MARITAL_ACS+HHSIZECAT+
                  TENURE_ACS+CHILDRENCAT+CITIZEN_REC+
     BORN_ACS+FAMINC5+EMPLOYED+MIL_ACS_REC+HOME_ACS_REC+FDSTMP_CPS+PUB_OFF_CPS+COMGRP_CPS+
     TALK_CPS+TRUST_CPS+TABLET_CPS+TEXTIM_CPS+SOCIAL_CPS+VOLSUM+REGISTERED+VOTE14+RELIGCAT+
     IDEO3+OWNGUN_GSS+FOLGOV
  ,data=nonprob2, family="binomial")
summary(regression)
# strong predictors for voting are are

#  IDEO3
#  VOTE14
#  OWNGUN
#  FAMINC5
#  HOME_ACS_REC
#  SOCIAL_CPS
#  BORN_ACS
#  EDUCCAT5
#  GENDER
#  FOLGOV
#  RELIGCAT
# COMGRP

# now what predicts R

complete <-readRDS("complete_USA_population_data_with_matched_survey_results.RDS")
prop.table(table(complete$IDEO3))
prop.table(table(nonprob$IDEO3))
prop.table(table(nonprob$PARTYSCALE5,nonprob$VOTESUM))
complete$missing <- 0
complete$missing[!is.na(complete$VOTESUM)] <- 1
table(complete$missing)
# predict missingness ith just the good Y predictors.
# (I select on Y predictors, because relation between X -> Y more important for corrections
## see slides on week nonresponse)
Rregression <- glm(missing~IDEO3+VOTE14+OWNGUN_GSS+FAMINC5+HOME_ACS_REC+
                     SOCIAL_CPS+BORN_ACS+EDUCCAT5+GENDER+RELIGCAT+FOLGOV+COMGRP_CPS,data=complete,family=
#table(complete$missing)
summary(Rregression)
# all are good predictors!
```

```r
partyscale5.dist <- as.data.frame(table(popdata$PARTYSCALE5))
colnames(partyscale5.dist) <- c("PARTYSCALE5", "Freq")
ideo3.dist <- as.data.frame(table(popdata$IDEO3))
colnames(ideo3.dist) <- c("IDEO3", "Freq")
owngun.dist <- as.data.frame(table(popdata$OWNGUN_GSS))
colnames(owngun.dist) <- c("OWNGUN_GSS", "Freq")
faminc5.dist <- as.data.frame(table(popdata$FAMINC5))
colnames(faminc5.dist) <- c("FAMINC5", "Freq")
home_acs_rec.dist <- as.data.frame(table(popdata$HOME_ACS_REC))
colnames(home_acs_rec.dist) <- c("HOME_ACS_REC", "Freq")
social_cps.dist <- as.data.frame(table(popdata$SOCIAL_CPS))
```

```r
colnames(social_cps.dist) <- c("SOCIAL_CPS", "Freq")
born_acs.dist <- as.data.frame(table(popdata$BORN_ACS))
colnames(born_acs.dist) <- c("BORN_ACS", "Freq")
educcat5.dist <- as.data.frame(table(popdata$EDUCCAT5))
colnames(educcat5.dist) <- c("EDUCCAT5", "Freq")
gender.dist <- as.data.frame(table(popdata$GENDER))
colnames(gender.dist) <- c("GENDER", "Freq")
folgov.dist <- as.data.frame(table(popdata$FOLGOV))
colnames(folgov.dist) <- c("FOLGOV", "Freq")
religcat.dist <- as.data.frame(table(popdata$RELIGCAT))
colnames(religcat.dist) <- c("RELIGCAT", "Freq")
comgrp_cps.dist <- as.data.frame(table(popdata$COMGRP_CPS))
colnames(comgrp_cps.dist) <- c("COMGRP_CPS", "Freq")

svy.unweighted <-svydesign(ids=~rid, data=nonprob)
```

```
## Warning in svydesign.default(ids = ~rid, data = nonprob): No weights or
## probabilities supplied, assuming equal probability
```

```r
#summary(complete)
#table(nonprob2$GENDER)
#table(complete$GENDER)

rakeddesign  <- rake(design=svy.unweighted,
                     sample.margins=list(~IDEO3,~OWNGUN_GSS,~FAMINC5,~SOCIAL_CPS,
                                         ~BORN_ACS,~HOME_ACS_REC,~EDUCCAT5,~GENDER,~VOTE14,~FOLGOV,
                                         ~RELIGCAT,~COMGRP_CPS,~PARTYSCALE5),
                     population.margins=list(ideo3.dist,owngun.dist,faminc5.dist,social_cps.dist,
                                  born_acs.dist,home_acs_rec.dist,educcat5.dist,gender.dist,vote.dist,
                                  folgov.dist,religcat.dist,comgrp_cps.dist,partyscale5.dist))
Peter <- svyby(~VOTESUM,~vendor, design=rakeddesign,svymean)
Peter$name <- "Peter"
```

now, combine results

```r
# add the raw percentages
raw <- svyby(~VOTESUM,~vendor, design=svy.unweighted,svymean)
raw$name <- "raw"

total <- rbind(Florian[,c(2,3,8)],Peter[,c(2,3,8)],raw[,c(2,3,8)])
# now subtract the actual numbers (our best estimate): 45% for Clinton and 43%
total$VOTESUMTrump <- as.numeric(total$VOTESUMTrump)
total$VOTESUMClinton <- as.numeric(total$VOTESUMClinton)
total$diffClinton <- abs(total$VOTESUMClinton - 0.45)
total$diffTrump <- abs(total$VOTESUMTrump - 0.43)
total$diff <- total$diffClinton+total$diffTrump

result <- total %>%
  group_by(name) %>%
  dplyr::summarize(Mean = mean(diff, na.rm=TRUE))

print(result)
```

```
## # A tibble: 3 x 2
##   name      Mean
##   <chr>    <dbl>
## 1 Florian 0.0547
## 2 Peter   0.0714
## 3 raw     0.121
```

```
#Florian 'wins' by using a double robust estimator. My solution gets a larger  MSE by weighting on a le
# An issue is that the % for both candidates is usually too high, because 3rd party voters are not meas
```

A final note: In the exercise people typically to weight the entire dataset. However, weighting by vendor (these are 3 separate datasets)is perhaps a better idea! Below is how this works using Peters weighting variables.

```
vendor1 <- subset(nonprob,vendor=="Vendor 1")
vendor2 <- subset(nonprob,vendor=="Vendor 2")
vendor3 <- subset(nonprob,vendor=="Vendor 3")


svy.vendor1 <-svydesign(ids=~rid, data=vendor1)
```

```
## Warning in svydesign.default(ids = ~rid, data = vendor1): No weights or
## probabilities supplied, assuming equal probability
```

```
svy.vendor2 <-svydesign(ids=~rid, data=vendor2)
```

```
## Warning in svydesign.default(ids = ~rid, data = vendor2): No weights or
## probabilities supplied, assuming equal probability
```

```
svy.vendor3 <-svydesign(ids=~rid, data=vendor3)
```

```
## Warning in svydesign.default(ids = ~rid, data = vendor3): No weights or
## probabilities supplied, assuming equal probability
```

```
# and now rake three times
rakedvendor1  <- rake(design=svy.vendor1,
                     sample.margins=list(~IDEO3,~OWNGUN_GSS,~FAMINC5,~SOCIAL_CPS,
                                         ~BORN_ACS,~HOME_ACS_REC,~EDUCCAT5,~GENDER,~VOTE14,~FOLGOV,
                                         ~RELIGCAT,~COMGRP_CPS,~PARTYSCALE5),
                     population.margins=list(ideo3.dist,owngun.dist,faminc5.dist,social_cps.dist,
                                  born_acs.dist,home_acs_rec.dist,educcat5.dist,gender.dist,vote.dist,
                                  folgov.dist,religcat.dist,comgrp_cps.dist,partyscale5.dist))
rakedvendor2  <- rake(design=svy.vendor2,
                     sample.margins=list(~IDEO3,~OWNGUN_GSS,~FAMINC5,~SOCIAL_CPS,
                                         ~BORN_ACS,~HOME_ACS_REC,~EDUCCAT5,~GENDER,~VOTE14,~FOLGOV,
                                         ~RELIGCAT,~COMGRP_CPS,~PARTYSCALE5),
                     population.margins=list(ideo3.dist,owngun.dist,faminc5.dist,social_cps.dist,
                                  born_acs.dist,home_acs_rec.dist,educcat5.dist,gender.dist,vote.dist,
                                  folgov.dist,religcat.dist,comgrp_cps.dist,partyscale5.dist))
rakedvendor3  <- rake(design=svy.vendor3,
                    sample.margins=list(~IDEO3,~OWNGUN_GSS,~FAMINC5,~SOCIAL_CPS,
                                         ~BORN_ACS,~HOME_ACS_REC,~EDUCCAT5,~GENDER,~VOTE14,~FOLGOV,
                                         ~RELIGCAT,~COMGRP_CPS,~PARTYSCALE5),
                     population.margins=list(ideo3.dist,owngun.dist,faminc5.dist,social_cps.dist,
                                  born_acs.dist,home_acs_rec.dist,educcat5.dist,gender.dist,vote.dist,
                                  folgov.dist,religcat.dist,comgrp_cps.dist,partyscale5.dist))
```

```r
Peter1 <- svyby(~VOTESUM,~vendor, design=rakedvendor1,svymean)
Peter2<- svyby(~VOTESUM,~vendor, design=rakedvendor2,svymean)
Peter3 <- svyby(~VOTESUM,~vendor, design=rakedvendor3,svymean)
Petervendor <- rbind(Peter1,Peter2,Peter3)
Petervendor$name <- "Peter"
print(Petervendor)
```

```
##            vendor VOTESUMTrump VOTESUMClinton VOTESUMUndecided se.VOTESUMTrump
## Vendor 1 Vendor 1    0.4712469      0.4939263       0.03482677      0.01856283
## Vendor 2 Vendor 2    0.4344760      0.4919075       0.07361651      0.02084060
## Vendor 3 Vendor 3    0.4592945      0.4948289       0.04587651      0.02226401
##          se.VOTESUMClinton se.VOTESUMUndecided  name
## Vendor 1        0.01860303          0.01358059 Peter
## Vendor 2        0.01706514          0.01835079 Peter
## Vendor 3        0.02010909          0.01653771 Peter
```