ch08output

2022-04-26

```
# Code from Chapter 8 of R Companion for Sampling: Design and Analysis by Yan Lu
# and Sharon L. Lohr
# All code is presented for educational purposes only and without warranty.

##### Install the R packages needed for the chapter

library(survey)
```

```
## Loading required package: grid
```

```
## Loading required package: Matrix
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survey'
```

```
## The following object is masked from 'package:graphics':
##
##     dotchart
```

```
library(sampling)
```

```
##
## Attaching package: 'sampling'
```

```
## The following objects are masked from 'package:survival':
##
##     cluster, strata
```

```
library(SDAResources)

########## How R Functions Treat Missing Data ##########

data(impute)
table0804<-impute
table0804
```

```
##    person age gender education crime violcrime
## 1       1  47      M        16     0         0
## 2       2  45      F        NA     1         1
## 3       3  19      M        11     0         0
## 4       4  21      F        NA     1         1
## 5       5  24      M        12     1         1
## 6       6  41      F        NA     0         0
## 7       7  36      M        20     1        NA
## 8       8  50      M        12     0         0
## 9       9  53      F        13     0        NA
## 10     10  17      M        10    NA        NA
```

```
## 11     11  53      F        12     0         0
## 12     12  21      F        12     0         0
## 13     13  18      F        11     1        NA
## 14     14  34      M        16     1         0
## 15     15  44      M        14     0         0
## 16     16  45      M        11     0         0
## 17     17  54      F        14     0         0
## 18     18  55      F        10     0         0
## 19     19  29      F        12    NA         0
## 20     20  32      F        10     0         0
```

```
table0804$crime
```

```
##  [1]  0  1  0  1  1  0  1  0  0 NA  0  0  1  1  0  0  0  0 NA  0
```

```
is.na(table0804$crime) # vector with TRUE for missing values
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
```

```
# identify the rows with no missing values in columns 5-6
table0804$cc<-complete.cases(table0804[,5:6])
table0804$cc
```

```
##  [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE
## [13] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
```

```
mean(table0804$crime)  # returns NA
```

```
## [1] NA
```

```
mean(table0804$crime,na.rm=TRUE) # calculates mean of non-missing values
```

```
## [1] 0.3333333
```

```
# excludes values missing in either variable
table(table0804$crime,table0804$violcrime)
```

```
##
##      0  1
##   0 11  0
##   1  1  3
```

```
# counts NAs as category in table
table(table0804$crime,table0804$violcrime,useNA="ifany")
```

```
##
##         0  1 <NA>
##   0    11  0    1
##   1     1  3    2
##   <NA>  1  0    1
```

```
# input design information, use relative weights of 1 for comparison with above
dimpute <- svydesign(id = ~1, weights = rep(1,20), data = table0804)
dimpute
```

```
## Independent Sampling design (with replacement)
## svydesign(id = ~1, weights = rep(1, 20), data = table0804)
```

```
# calculate survey mean and se
svymean(~crime,dimpute) # returns NA
```

```
##       mean SE
## crime   NA NA
```

```
svymean(~crime, dimpute, na.rm=TRUE)
```

```
##          mean    SE
## crime 0.33333 0.114
```

```
svymean(~factor(crime), dimpute, na.rm=TRUE)
```

```
##                    mean    SE
## factor(crime)0 0.66667 0.114
## factor(crime)1 0.33333 0.114
```

```
svytable(~violcrime+crime,dimpute)
```

```
##          crime
## violcrime  0  1
##         0 11  1
##         1  0  3
```

```
########## Poststratification and Raking ##########

rakewtsum <- data.frame(gender=rep(c("F","M"),each=5),
                race=rep(c("Black","White","Asian","NatAm","Other"),times=2),
                wtsum=c(300,1200,60,30,30,150,1080,90,30,30))
rakewtsum # check data entry
```

```
##    gender  race wtsum
## 1       F Black   300
## 2       F White  1200
## 3       F Asian    60
## 4       F NatAm    30
## 5       F Other    30
## 6       M Black   150
## 7       M White  1080
## 8       M Asian    90
## 9       M NatAm    30
## 10      M Other    30
```

```
# Need data frame with individual records to use rake function
rakedf <- rakewtsum[rep(row.names(rakewtsum), rakewtsum[,3]/6), 1:2]
dim(rakedf)
```

```
## [1] 500   2
```

```
rakedf$wt <- rep(6,nrow(rakedf))
head(rakedf)
```

```
##     gender  race wt
## 1        F Black  6
## 1.1      F Black  6
## 1.2      F Black  6
## 1.3      F Black  6
## 1.4      F Black  6
## 1.5      F Black  6
```

```
# Create the survey design object
drake <- svydesign(id=~1, weights=~wt, data=rakedf)
```

```r
# Create data frames containing the marginal counts
pop.gender <- data.frame(gender=c("F","M"), Freq=c(1510,1490))
pop.race <- data.frame(race=c("Black","White","Asian","NatAm","Other"),
                       Freq=c(600,2120,150,100,30))
# Now create survey design object with raked weights
drake2 <- rake(drake, list(~gender,~race), list(pop.gender, pop.race))
drake2 # describes SRS with replacement
```

```
## Independent Sampling design (with replacement)
## rake(drake, list(~gender, ~race), list(pop.gender, pop.race))
```

```r
# Look at first 10 entries in vector of raked weights
weights(drake2)[1:10]
```

```
##        1      1.1      1.2      1.3      1.4      1.5      1.6      1.7
## 7.511886 7.511886 7.511886 7.511886 7.511886 7.511886 7.511886 7.511886
##      1.8      1.9
## 7.511886 7.511886
```

```r
# Look at sum of raked weights for raking cells
svytable(~gender+race, drake2)
```

```
##       race
## gender      Asian      Black      NatAm      Other      White
##      F   53.71714  375.59431   45.55940   13.66782 1021.46870
##      M   96.28286  224.40569   54.44060   16.33218 1098.53130
```

```r
# Look at sum of raked weights for margins
svytotal(~factor(gender),drake2)
```

```
##                 total     SE
## factor(gender)F  1510 0.0028
## factor(gender)M  1490 0.0028
```

```r
svytotal(~factor(race),drake2)
```

```
##                   total SE
## factor(race)Asian   150  0
## factor(race)Black   600  0
## factor(race)NatAm   100  0
## factor(race)Other    30  0
## factor(race)White  2120  0
```

```r
########## Imputation ##########

##### Example 8.9

##### cell mean imputation #####
table0804$education
```

```
##  [1] 16 NA 11 NA 12 NA 20 12 13 10 12 12 11 16 14 11 14 10 12 10
```

```r
impute.cm<-table0804
# define matrix giving imputation flags, TRUE for each missing value
impute.flag<-is.na(table0804)
impute.flag
```

```
##       person   age gender education crime violcrime    cc
##  [1,]  FALSE FALSE  FALSE     FALSE FALSE     FALSE FALSE
```

```
##  [2,]  FALSE FALSE  FALSE      TRUE FALSE     FALSE FALSE
##  [3,]  FALSE FALSE  FALSE     FALSE FALSE     FALSE FALSE
##  [4,]  FALSE FALSE  FALSE      TRUE FALSE     FALSE FALSE
##  [5,]  FALSE FALSE  FALSE     FALSE FALSE     FALSE FALSE
##  [6,]  FALSE FALSE  FALSE      TRUE FALSE     FALSE FALSE
##  [7,]  FALSE FALSE  FALSE     FALSE FALSE      TRUE FALSE
##  [8,]  FALSE FALSE  FALSE     FALSE FALSE     FALSE FALSE
##  [9,]  FALSE FALSE  FALSE     FALSE FALSE      TRUE FALSE
## [10,]  FALSE FALSE  FALSE     FALSE  TRUE      TRUE FALSE
## [11,]  FALSE FALSE  FALSE     FALSE FALSE     FALSE FALSE
## [12,]  FALSE FALSE  FALSE     FALSE FALSE     FALSE FALSE
## [13,]  FALSE FALSE  FALSE     FALSE FALSE      TRUE FALSE
## [14,]  FALSE FALSE  FALSE     FALSE FALSE     FALSE FALSE
## [15,]  FALSE FALSE  FALSE     FALSE FALSE     FALSE FALSE
## [16,]  FALSE FALSE  FALSE     FALSE FALSE     FALSE FALSE
## [17,]  FALSE FALSE  FALSE     FALSE FALSE     FALSE FALSE
## [18,]  FALSE FALSE  FALSE     FALSE FALSE     FALSE FALSE
## [19,]  FALSE FALSE  FALSE     FALSE  TRUE     FALSE FALSE
## [20,]  FALSE FALSE  FALSE     FALSE FALSE     FALSE FALSE
```

```
# fit two-way model with interaction, omit NAs from model-fitting
edmodel<-lm(education~factor(gender)*factor(age>=35),
            data=impute.cm,na.action=na.omit)
# replace missing values with imputations from model
newdata<- table0804[is.na(table0804$education),]
impute.cm$education[is.na(table0804$education)] <- predict(edmodel,newdata)
impute.cm$education
```

```
##  [1] 16.00 12.25 11.00 11.25 12.00 12.25 20.00 12.00 13.00 10.00 12.00 12.00
## [13] 11.00 16.00 14.00 11.00 14.00 10.00 12.00 10.00
```

```
#####use Hmisc
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:SDAResources':
##
##     seals

##
## Attaching package: 'Hmisc'

## The following object is masked from 'package:survey':
##
##     deff

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# impute() function imputes missing value using user defined statistical
# method (mean, max, mean). It's default is median
# impute NA of education with mean value 12.70588
mean(table0804$education,na.rm=TRUE)
```

```
## [1] 12.70588
```

```
table0804$edu2<-with(table0804,impute(education, mean))
table0804
```

```
## # A tibble: 20 x 8
##    person   age gender education crime violcrime cc    edu2
##     <dbl> <dbl> <chr>      <dbl> <dbl>     <dbl> <lgl> <impute>
##  1      1    47 M             16     0         0 TRUE  16.00000
##  2      2    45 F             NA     1         1 TRUE  12.70588
##  3      3    19 M             11     0         0 TRUE  11.00000
##  4      4    21 F             NA     1         1 TRUE  12.70588
##  5      5    24 M             12     1         1 TRUE  12.00000
##  6      6    41 F             NA     0         0 TRUE  12.70588
##  7      7    36 M             20     1        NA FALSE 20.00000
##  8      8    50 M             12     0         0 TRUE  12.00000
##  9      9    53 F             13     0        NA FALSE 13.00000
## 10     10    17 M             10    NA        NA FALSE 10.00000
## 11     11    53 F             12     0         0 TRUE  12.00000
## 12     12    21 F             12     0         0 TRUE  12.00000
## 13     13    18 F             11     1        NA FALSE 11.00000
## 14     14    34 M             16     1         0 TRUE  16.00000
## 15     15    44 M             14     0         0 TRUE  14.00000
## 16     16    45 M             11     0         0 TRUE  11.00000
## 17     17    54 F             14     0         0 TRUE  14.00000
## 18     18    55 F             10     0         0 TRUE  10.00000
## 19     19    29 F             12    NA         0 FALSE 12.00000
## 20     20    32 F             10     0         0 TRUE  10.00000
```

```
# impute NA of crime with random value
table0804$crime2 <- with(table0804, impute(crime, 'random'))
table0804[1:10,]
```

```
## # A tibble: 10 x 9
##    person   age gender education crime violcrime cc    edu2     crime2
##     <dbl> <dbl> <chr>      <dbl> <dbl>     <dbl> <lgl> <impute> <impute>
##  1      1    47 M             16     0         0 TRUE  16.00000 0
##  2      2    45 F             NA     1         1 TRUE  12.70588 1
##  3      3    19 M             11     0         0 TRUE  11.00000 0
##  4      4    21 F             NA     1         1 TRUE  12.70588 1
##  5      5    24 M             12     1         1 TRUE  12.00000 1
##  6      6    41 F             NA     0         0 TRUE  12.70588 0
##  7      7    36 M             20     1        NA FALSE 20.00000 1
##  8      8    50 M             12     0         0 TRUE  12.00000 0
##  9      9    53 F             13     0        NA FALSE 13.00000 0
## 10     10    17 M             10    NA        NA FALSE 10.00000 0
```

```
#####regression imputation#####

# regression fit
# input design information dimpute
```

```r
myfit<-svyglm(education~age, design=dimpute)
summary(myfit)
```

```
##
## Call:
## svyglm(formula = education ~ age, design = dimpute)
##
## Survey design:
## svydesign(id = ~1, weights = rep(1, 20), data = table0804)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.20977    1.17615   9.531 9.37e-08 ***
## age          0.04031    0.02976   1.355    0.196
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 6.654205)
##
## Number of Fisher Scoring iterations: 2
```

```r
missing <-which(is.na(table0804$education))
missing
```

```
## [1] 2 4 6
```

```r
length(missing)
```

```
## [1] 3
```

```r
newdata <- data.frame(age=table0804$age[missing])
newdata
```

```
##   age
## 1  45
## 2  21
## 3  41
```

```r
aa<-predict(myfit, newdata) #3 predicted
aa
```

```
##     link     SE
## 1 13.024 0.7003
## 2 12.056 0.7214
## 3 12.862 0.6516
```

```r
# #use b0+ b1*age to do prediction
#check predicted/imputed values one by one
myfit$coefficients[1]+myfit$coefficients[2]*45  #No.2 missing
```

```
## (Intercept)
##     13.0236
```

```r
myfit$coefficients[1]+myfit$coefficients[2]*21  #No.4 missing
```

```
## (Intercept)
##    12.05622
```

```
myfit$coefficients[1]+myfit$coefficients[2]*41  #No.6 missing
```

```
## (Intercept)
##    12.86237
```

```
# #complete data
impute.cm2<-table0804
impute.cm2$education[missing]<-aa
impute.cm2
```

```
## # A tibble: 20 x 9
##    person   age gender education crime violcrime cc    edu2     crime2
##     <dbl> <dbl> <chr>      <dbl> <dbl>     <dbl> <lgl> <impute> <impute>
##  1      1    47 M           16       0         0 TRUE  16.00000 0
##  2      2    45 F           13.0     1         1 TRUE  12.70588 1
##  3      3    19 M           11       0         0 TRUE  11.00000 0
##  4      4    21 F           12.1     1         1 TRUE  12.70588 1
##  5      5    24 M           12       1         1 TRUE  12.00000 1
##  6      6    41 F           12.9     0         0 TRUE  12.70588 0
##  7      7    36 M           20       1        NA FALSE 20.00000 1
##  8      8    50 M           12       0         0 TRUE  12.00000 0
##  9      9    53 F           13       0        NA FALSE 13.00000 0
## 10     10    17 M           10      NA        NA FALSE 10.00000 0
## 11     11    53 F           12       0         0 TRUE  12.00000 0
## 12     12    21 F           12       0         0 TRUE  12.00000 0
## 13     13    18 F           11       1        NA FALSE 11.00000 1
## 14     14    34 M           16       1         0 TRUE  16.00000 1
## 15     15    44 M           14       0         0 TRUE  14.00000 0
## 16     16    45 M           11       0         0 TRUE  11.00000 0
## 17     17    54 F           14       0         0 TRUE  14.00000 0
## 18     18    55 F           10       0         0 TRUE  10.00000 0
## 19     19    29 F           12      NA         0 FALSE 12.00000 0
## 20     20    32 F           10       0         0 TRUE  10.00000 0
```

```
##### For most imputation applications, we recommend using one of the R packages
#described in the text
```