Stat 138: Introduction to Sampling Designs
Problem Set 6

Anne Christine Amores

June 3, 2025

ASA Certification Survey

The American Statistical Association (ASA) studied whether it should offer a certification designation for its members, so that statisticians meeting the qualifications could be designated as "Certified Statisticians." In 1994, the ASA surveyed its membership about this issue, with data in file certify.dat. The survey was sent to all 18,609 members; 5001 responses were obtained. Results from the survey were reported in the October 1994 issue of *Amstat News*. Assume that in 1994, the ASA membership had the following characteristics: 55% have Ph.D.'s and 38% have Master's degrees; 29% work in industry, 34% work in academia, and 11% work in government. The cross-classification between education and workplace was unavailable.

(a) What are the response rates for the various subclasses of ASA membership? Are the nonrespondents MCAR? Do you think they are MAR?

```
# Loading the necessary packages
library(readxl)
library(dplyr)
library(survey)
library(tidyr)
```

```
# Importing the dataset
cert <- read_excel("certify.xlsx")
head(cert)
```

```
## # A tibble: 6 x 11
##   certify approve speccert wouldyou recert subdisc college employ workenv
##     <dbl>   <dbl>    <dbl>    <dbl>  <dbl> <chr>   <chr>   <chr>  <chr>
## 1       4       2        5        5      3 BI      P       E      G
## 2       2       2        3        2      1 QP      P       E      I
## 3       2       2        4        5      4 BI      P       E      A
## 4       1       1        2        1      5 CE      M       E      A
## 5       1       1        1        1      2 SP      P       E      I
## 6       5       5        5        3      5 BI      P       E      G
## # i 2 more variables: workact <chr>, yearsmem <chr>
```

```
# Cleaning and recoding missing values
cert <- cert %>%
  mutate(
    college = ifelse(is.na(college), "0", college),
    workenv = ifelse(is.na(workenv), "0", workenv)
  )
```

```r
# Collapsing education levels and workplace into groups matching population
   proportions
cert <- cert %>%
  mutate(
    edu_group = case_when(
      college == "P" ~ "PhD",
      college == "M" ~ "Masters",
      TRUE ~ "Other"
    )
  )

# Recoding workenv into full labels
cert <- cert %>%
  mutate(
    workenv_full = case_when(
      workenv == "A" ~ "Academia",
      workenv == "G" ~ "Government",
      workenv == "I" ~ "Industry",
      TRUE ~ "Other"
    )
  )

# Defining total population and proportions
total_pop <- 18609

edu_pop_prop <- c(PhD = 0.55, Masters = 0.38,
    Other = 0.07)
workenv_pop_prop <- c(Industry = 0.29, Academia = 0.34, Government = 0.11, Other = 0.26)

# Converting to population counts
edu_pop_count <- round(edu_pop_prop * total_pop)
workenv_pop_count <- round(workenv_pop_prop * total_pop)

# Counting respondents (non-missing 'certify')
edu_respondents <- cert %>%
  filter(!is.na(certify)) %>%
  count(edu_group) %>%
  filter(edu_group %in% names(edu_pop_count))

workenv_respondents <- cert %>%
  filter(!is.na(certify)) %>%
  count(workenv_full) %>%
  filter(workenv_full %in% names(workenv_pop_count))

# Merging with population counts and computing response rates
edu_response <- edu_respondents %>%
  mutate(
    pop = edu_pop_count[edu_group],
    response_rate = n / pop
  )

workenv_response <- workenv_respondents %>%
  mutate(
    pop = workenv_pop_count[workenv_full],
```

```
    response_rate = n / pop
  )
```

```
# Formatting education summary table
edu_summary <- edu_response %>%
  transmute(
    `Education Level` = edu_group,
    `Response Rate` = paste0(round(response_rate * 100), "%")
  )

# Formatting work environment summary table
workenv_summary <- workenv_response %>%
  transmute(
    `Work Environment` = workenv_full,
    `Response Rate` = paste0(round(response_rate * 100), "%")
  )

cat("Summary: Response Rates by Education Level\n")
```

```
## Summary: Response Rates by Education Level
```

```
print(edu_summary)
```

```
## # A tibble: 3 x 2
##   `Education Level` `Response Rate`
##   <chr>             <chr>
## 1 Masters           23%
## 2 Other             25%
## 3 PhD               30%
```

```
cat("\n Summary: Response Rates by Work Environment\n")
```

```
##
##  Summary: Response Rates by Work Environment
```

```
print(workenv_summary)
```

```
## # A tibble: 4 x 2
##   `Work Environment` `Response Rate`
##   <chr>              <chr>
## 1 Academia           35%
## 2 Government         43%
## 3 Industry           34%
## 4 Other              2%
```

The nonresponse is **not MCAR** because response rates differ by education level and work environment. It is **unlikely to be MAR** because the missingness probably depends on factors beyond the observed variables, such as unmeasured attitudes or opinions.

(b) Use raking to adjust the weights for the six cells defined by education (Ph.D. or non-Ph.D.) and workplace (industry, academia, or other). Start with an initial weight of 18,609/5001 for each respondent. What assumptions must you make to use raking?

```
# Recoding both education and work environment to match population margins
cert <- cert %>%
  mutate(
     # Collapsing education into "PhD" and "non-PhD"
    edu_group_rake = case_when(
      edu_group == "PhD" ~ "PhD",
      TRUE ~ "non-PhD"
    ),
    # Collapsing workenv: Government becomes Other
    workenv_rake = case_when(
      workenv_full == "Government" ~ "Other",
      TRUE ~ workenv_full
    ),
    # Making sure factors match margin levels
    edu_group_rake = factor(edu_group_rake, levels = c("PhD", "non-PhD")),
    workenv_rake = factor(workenv_rake, levels = c("Industry", "Academia",
                                                   "Other"))
  )

# Creating a cross-tabulation of sample counts
initial_counts <- cert %>%
  count(edu_group_rake, workenv_rake) %>%
  tidyr::pivot_wider(names_from = workenv_rake, values_from = n, values_fill = 0)

# Adding row totals
initial_counts <- initial_counts %>%
  mutate(Total = Industry + Academia + Other)

# Adding column totals
col_totals <- initial_counts %>%
  summarise(across(Industry:Total, sum)) %>%
  mutate(edu_group_rake = "Total")

# Combining table with totals
initial_counts_full <- bind_rows(initial_counts, col_totals)

cat("Sample Counts per Cell (PhD vs. non-PhD × Work Environment):\n")
```

```
## Sample Counts per Cell (PhD vs. non-PhD × Work Environment):
```

```
print(initial_counts_full)
```

```
## # A tibble: 3 x 5
##   edu_group_rake Industry Academia Other Total
##   <chr>             <int>    <int> <int> <int>
## 1 PhD                 798     1787   451  3036
## 2 non-PhD            1011      434   520  1965
## 3 Total              1809     2221   971  5001
```

```r
# Defining initial weight as a standalone variable
init_weight <- 18609 / 5001

# Defining population margins
edu_dist <- data.frame(edu_group_rake = c("PhD", "non-PhD"),
                       Freq = c(10235, 8374))

workenv_dist <- data.frame(workenv_rake = c("Industry", "Academia", "Other"),
                           Freq = c(5397, 6327, 6885))

# Count of respondents per cell
initial_counts <- cert %>%
  count(edu_group_rake, workenv_rake)

# Multiplying counts by initial weight to get total weight per cell
initial_weights <- initial_counts %>%
  mutate(weight_total = n * init_weight) %>%
  dplyr::select(-n) %>%
  pivot_wider(names_from = workenv_rake, values_from = weight_total, values_fill = 0)

# Adding row totals
initial_weights <- initial_weights %>%
  mutate(Total = Industry + Academia + Other)

# Adding column totals
col_totals <- initial_weights %>%
  summarise(across(Industry:Total, sum)) %>%
  mutate(edu_group_rake = "Total")

# Final table with row and column totals
initial_weights_table <- bind_rows(initial_weights, col_totals) %>%
  dplyr::select(edu_group_rake, Industry, Academia, Other, Total)

# Rounding all numeric columns to 1 decimal place
initial_weights_table_rounded <- initial_weights_table %>%
  mutate(across(where(is.numeric), ~ round(., 1)))

cat("Initial Sum of Weights per Cell (Before Raking, Rounded to 1 Decimal):\n")
```

```
## Initial Sum of Weights per Cell (Before Raking, Rounded to 1 Decimal):
```

```r
print(initial_weights_table_rounded)
```

```
## # A tibble: 3 x 5
##   edu_group_rake Industry Academia Other  Total
##   <chr>             <dbl>    <dbl> <dbl>  <dbl>
## 1 PhD               2969.    6650. 1678. 11297.
## 2 non-PhD           3762     1615. 1935.  7312.
## 3 Total             6731.    8264. 3613. 18609
```

```r
# Creating initial weights
cert$init_weight <- 18609 / 5001

# Setting up initial survey design
design_init <- svydesign(ids = ~1, data = cert, weights = ~init_weight)
```

```r
# Raking
design_raked <- rake(design_init,
                     sample.margins = list(~edu_group_rake, ~workenv_rake),
                     population.margins = list(edu_dist, workenv_dist))

# Getting weighted counts table
tab <- svytable(~edu_group_rake + workenv_rake, design_raked)

# Adding margins (row and column totals)
tab_with_margins <- addmargins(tab)

# Rounding the table values to 1 decimal place
tab_rounded <- round(tab_with_margins, 1)

# Print the rounded table
print(tab_rounded)
```

```
##                workenv_rake
## edu_group_rake Industry Academia   Other     Sum
##         PhD       2239.2   4980.5  3015.2 10234.9
##         non-PhD   3157.8   1346.5  3869.8  8374.1
##         Sum       5397.0   6327.0  6885.0 18609.0
```

Raking relies on the following assummptions:

- The marginal distributions usedd reflect the true population.
- Response probabilities depend only on the variables used in raking (e.g., row and column margins), not on specific interactions within cells.
- The missing data mechanism is ideally MAR, although this may not always hold in practice.
- Extreme weights are avoided to reduce undue influence and variance inflation.

Observing the tables above, we see that the raking process caused the weights for respondents in the "Other" employment category to become much larger.

(c) Can you conclude from this survey that a majority of the ASA membership opposed certification in 1994? Why, or why not?

```r
# Unweighted proportions including 0
unweighted_props <- cert %>%
  filter(certify %in% 0:5) %>%  # include no response (0)
  group_by(certify) %>%
  summarise(unweighted_count = n()) %>%
  mutate(unweighted_prop = unweighted_count / sum(unweighted_count) * 100)

# Weighted proportions (raked) including 0
weighted_props <- svytable(~certify, design_raked)
weighted_props_df <- as.data.frame(weighted_props) %>%
  rename(weighted_count = Freq) %>%
  filter(certify %in% 0:5)

# Fix factor vs numeric issue
weighted_props_df$certify <- as.numeric(as.character(weighted_props_df$certify))

weighted_props_df <- weighted_props_df %>%
```

```r
  mutate(weighted_prop = weighted_count / sum(weighted_count) * 100)

# Join unweighted and weighted
props_combined <- unweighted_props %>%
  left_join(weighted_props_df, by = "certify") %>%
  dplyr::select(certify, unweighted_prop, weighted_prop) %>%
  mutate(across(where(is.numeric), ~round(., 1)))

print(props_combined)
```

```
## # A tibble: 6 x 3
##   certify unweighted_prop weighted_prop
##     <dbl>           <dbl>         <dbl>
## 1       0             0.2           0.3
## 2       1            26.4          25.8
## 3       2            22.3          22.3
## 4       3             5.4           5.4
## 5       4             6.7           6.9
## 6       5            39            39.3
```

Based on the results, about 39% of respondents chose code 5, which represents oppposition to certification. Since this is less than half of the total responses, the data does not support the conclusion that a majority of ASA members opposed certification in 1994.

Survey of Youth in Custody

Weights are used in the Survey of Youth in Custody to adjust for unit nonresponse. Use a regression pocedure to impute values for the variable measuring with the number of times the youth was arrested. What variables will you use as the predictors?

Step 1: Choosing predictors

Good candidate predictors that are likely correlated with *numarr* are:

- age: age of resident
- race: categorical race variable
- ethnicity: hispanic or not
- sex: male or female
- educ: highest educational attainment
- livewith: who they lived with growing up (family structure)
- famtime: family incarceration history
- crimtype: type of current offense
- everviol: history of violent offense probation
- probtn: number of times on probation
- corrinst: prior corrections
- evertime: prior correctional instution stay
- prviol, prvprop, prdrug, prpub, prjuv: prior arrests for various offenses
- agefirst: age first arrested

- usewepn, alcuse, everdrug: other relevant behavioral variables

Step 2: Regression Imputation in R

```
syc <- read_excel("syc.xlsx")
head(syc)
```

```
## # A tibble: 6 x 28
##   stratum facility psusize initwt finalwt randgrp   age  race
##     <dbl>    <dbl>   <dbl>  <dbl>   <dbl>   <dbl> <dbl> <dbl>
## 1       7       20      53      8      10       5    14     2
## 2       7       20      53      8      10       6    13     2
## 3       7       20      53      8      10       7    12     2
## 4       7       20      53      8      10       1    15     2
## 5       7       20      53      8      10       2    14     2
## 6       7       20      53      8      14       3    14     1
## # i 20 more variables: ethnicity <dbl>, educ <dbl>, sex <dbl>,
## #   livewith <dbl>, famtime <dbl>, crimtype <dbl>, everviol <dbl>,
## #   numarr <dbl>, probtn <dbl>, corrinst <dbl>, evertime <dbl>,
## #   prviol <dbl>, prprop <dbl>, prdrug <dbl>, prpub <dbl>,
## #   prjuv <dbl>, agefirst <dbl>, usewepn <dbl>, alcuse <dbl>,
## #   everdrug <dbl>
```

```
# Recoding missing values for numarr and other numeric variables
  as NA
syc <- syc %>%
  mutate(
    numarr = ifelse(numarr == 99, NA, numarr),
    age = ifelse(age == 99, NA, age),
    probtn = ifelse(probtn == 99, NA, probtn),
    corrinst = ifelse(corrinst == 99, NA, corrinst),
    agefirst = ifelse(agefirst == 99, NA, agefirst),
    educ = ifelse(educ == 99, NA, educ),
  )

# Fitting linear regression model on non-missing numarr
fit <- lm(numarr ~ age + factor(race) + ethnicity + factor(sex) +
       factor(educ) + factor(livewith) + factor(famtime) +
       factor(crimtype) + everviol + probtn + corrinst +
       factor(evertime) + prviol + prprop + prdrug + prpub +
       prjuv + agefirst + factor(usewepn) + factor(alcuse) +
       everdrug, data = syc, na.action = na.exclude)

# Predicting missing numarr values
missing_idx <- which(is.na(syc$numarr))
predicted_values <- predict(fit, newdata = syc[missing_idx, ])

# Imputing predicted values back into data
```

```r
syc$numarr[missing_idx] <- pmax(0, round(predicted_values))

head(syc$numarr)
```

```
## [1] 24  3  4  6  4 20
```

```r
summary(fit)
```

```
##
## Call:
## lm(formula = numarr ~ age + factor(race) + ethnicity + factor(sex) +
##     factor(educ) + factor(livewith) + factor(famtime) + factor(crimtype) +
##     everviol + probtn + corrinst + factor(evertime) + prviol +
##     prprop + prdrug + prpub + prjuv + agefirst + factor(usewepn) +
##     factor(alcuse) + everdrug, data = syc, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.000  -4.689  -1.633   1.509  91.896
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         20.96160    8.50022   2.466  0.01373 *
## age                  0.46973    0.15249   3.080  0.00209 **
## factor(race)2        0.01500    0.51889   0.029  0.97695
## factor(race)3       -0.79777    2.40629  -0.332  0.74027
## factor(race)4       -0.75087    1.40787  -0.533  0.59385
## factor(race)5        1.61620    1.55824   1.037  0.29975
## factor(race)9        2.99448    6.41792   0.467  0.64084
## ethnicity            0.58018    0.44360   1.308  0.19104
## factor(sex)2        -0.56523    0.99631  -0.567  0.57055
## factor(educ)1        2.42372   13.48528   0.180  0.85738
## factor(educ)3       -7.47850   11.00700  -0.679  0.49693
## factor(educ)4       24.70576   10.12339   2.440  0.01474 *
## factor(educ)5       -0.28373    9.26851  -0.031  0.97558
## factor(educ)6        2.08183    7.94082   0.262  0.79321
## factor(educ)7       -0.40215    7.87267  -0.051  0.95926
## factor(educ)8       -0.48360    7.85736  -0.062  0.95093
## factor(educ)9        0.87470    7.84654   0.111  0.91125
## factor(educ)10       0.95535    7.84342   0.122  0.90307
## factor(educ)11       0.99891    7.85584   0.127  0.89883
## factor(educ)12       1.88414    7.88191   0.239  0.81109
## factor(educ)13       0.44944    8.35995   0.054  0.95713
## factor(educ)14       1.84924    8.22641   0.225  0.82216
## factor(livewith)2   -1.62536    1.01242  -1.605  0.10853
## factor(livewith)3   -0.44205    0.52030  -0.850  0.39563
## factor(livewith)4    0.24873    0.80279   0.310  0.75671
```

```
## factor(livewith)5  -0.61891    1.47295  -0.420  0.67439
## factor(livewith)6  -2.96220    4.92948  -0.601  0.54795
## factor(livewith)7  -3.51603    1.79199  -1.962  0.04987 *
## factor(livewith)8  -2.10651    5.59175  -0.377  0.70642
## factor(livewith)9   0.61977    1.82610   0.339  0.73434
## factor(livewith)99 -9.75451    6.38570  -1.528  0.12675
## factor(famtime)2   -0.04848    0.45959  -0.105  0.91600
## factor(famtime)9    4.58860    2.11024   2.174  0.02977 *
## factor(crimtype)2   0.72233    0.78911   0.915  0.36009
## factor(crimtype)3   0.63943    1.14435   0.559  0.57637
## factor(crimtype)4  -0.99153    1.23069  -0.806  0.42051
## factor(crimtype)5   2.68480    1.94282   1.382  0.16713
## factor(crimtype)9  -3.32516    3.93609  -0.845  0.39831
## everviol           -0.30774    0.93006  -0.331  0.74076
## probtn              0.27174    0.11855   2.292  0.02197 *
## corrinst            0.49710    0.06221   7.991 2.04e-15 ***
## factor(evertime)2  -0.01101    0.53658  -0.021  0.98363
## factor(evertime)9  -4.55009    4.65180  -0.978  0.32810
## prviol              0.29278    0.71014   0.412  0.68017
## prprop              1.38072    0.55898   2.470  0.01358 *
## prdrug              1.60183    0.57883   2.767  0.00569 **
## prpub               1.38945    0.62088   2.238  0.02532 *
## prjuv               0.95042    0.51677   1.839  0.06602 .
## agefirst           -1.80405    0.11386 -15.844  < 2e-16 ***
## factor(usewepn)2   -1.24912    0.54206  -2.304  0.02128 *
## factor(usewepn)9   -2.90752    3.51098  -0.828  0.40768
## factor(alcuse)2    -2.01536    0.91559  -2.201  0.02782 *
## factor(alcuse)3    -0.76607    0.64230  -1.193  0.23310
## factor(alcuse)9    14.92024    6.57158   2.270  0.02327 *
## everdrug            0.01915    0.44596   0.043  0.96575
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.94 on 2436 degrees of freedom
##   (130 observations deleted due to missingness)
## Multiple R-squared:  0.2586, Adjusted R-squared:  0.2422
## F-statistic: 15.74 on 54 and 2436 DF,  p-value: < 2.2e-16
```