gh1

Min Kim

2025-02-03

```
library(here)
```

```
## here() starts at /Users/kim/University of Michigan Dropbox/Min Kim/2025 Winter Term/STATS 480
```

```
library(data.table)
library(readr)
```

4.1

```
classsur <- read.table(here("Graded Homework", "classsur.txt"), header = TRUE, sep = "", fill = TRUE, s
head(classsur)
```

```
##   Gender Age  GPA Class    Ht  Wt StudyHrs SleepHrs Job TextPay Reside
## 1      2  19 2.50     2  70.5 147     12.0        7   2     200       2
## 2      2  20 2.30     3  71.0 158     11.8        7   2     170       1
## 3      2  17 1.00    65 140.0   6      6.4        1 200       2      NA
## 4      1  23 2.80     4  72.0 160      9.0        7   2     111       2
## 5      2  33 3.45     4  65.0 155     10.0        7   1     150       2
## 6      2  20 2.50     3  60.0 138     10.0        8   1     250       1
```

```
summary(classsur)
```

```
##      Gender           Age             GPA            Class
##  Min.   :1.000   Min.   :17.00   Min.   :1.000   Min.   :  1.00
##  1st Qu.:1.000   1st Qu.:19.00   1st Qu.:2.600   1st Qu.:  3.00
##  Median :2.000   Median :20.00   Median :2.930   Median :  3.00
##  Mean   :1.596   Mean   :21.16   Mean   :2.777   Mean   : 13.84
##  3rd Qu.:2.000   3rd Qu.:21.00   3rd Qu.:3.200   3rd Qu.:  4.00
##  Max.   :2.000   Max.   :42.00   Max.   :3.910   Max.   :108.00
##
##        Ht              Wt          StudyHrs        SleepHrs
##  Min.   :  9.00   Min.   :  4.0   Min.   : 2.00   Min.   :  1.0
##  1st Qu.: 64.00   1st Qu.:110.0   1st Qu.: 6.80   1st Qu.:  6.0
##  Median : 67.00   Median :135.0   Median :10.00   Median :  7.0
##  Mean   : 70.72   Mean   :121.2   Mean   :11.62   Mean   : 23.9
##  3rd Qu.: 71.00   3rd Qu.:160.0   3rd Qu.:14.00   3rd Qu.:  8.0
##  Max.   :175.00   Max.   :240.0   Max.   :40.00   Max.   :260.0
##
##       Job            TextPay          Reside
##  Min.   :  1.00   Min.   :  1.0   Min.   :1.000
##  1st Qu.:  1.00   1st Qu.:117.8   1st Qu.:1.000
##  Median :  2.00   Median :200.0   Median :2.000
##  Mean   : 16.92   Mean   :168.1   Mean   :1.733
##  3rd Qu.:  2.00   3rd Qu.:221.2   3rd Qu.:2.000
```

```
##  Max.   :280.00   Max.   :400.0   Max.   :3.000
##                   NA's   :5       NA's   :12
```

a.

Estimate the population mean of one of the measurement variables, such as age, grade point average (GPA), or study hours.

To estimate the population mean for a simple random sample of n accounts, we employ the sample average

$$\bar{y}$$

.

$$\bar{y} = \frac{\Sigma_{i=1}^{n} y_i}{n}$$

```r
sum_age <- sum(classsur$Age)
n <- nrow(classsur)
y_bar <- sum_age / n
print(y_bar)
```

```
## [1] 21.15789
```

b.

Estimate a population proportion for one of the categorical variables, such as gender, class, or job status.

To estimate the population proportion, we employ

$$\hat{p}$$

.

```r
# Estimation population proportion of male samples / total number of samples
male <- sum(classsur$Gender == 1)
pop_prop_male <- male / n
print(pop_prop_male)
```

```
## [1] 0.4035088
```

c.

Compare means on one variable for at least two different groups, such as men and women.

We use the same variable, Gender, as part b) for part c). For a random sample from a population with independent random sample from a population with means

$$\mu_y$$

and

$$\mu_x$$

, an unbiased estimate of

$$mu_y - mu_x$$

is

$$\bar{y} - \bar{x}$$

.

```r
mean_male <- mean(classsur$Gender == 1)
mean_female <- mean(classsur$Gender == 2)
diff_means <- mean_male - mean_female
print(diff_means)
```

```
## [1] -0.1929825
```

d.

Compare proportions on one categorical variable for at least two different groups (i.e., class standing or location of permanent residence).

We compare the proportions of class standing. Class standing, provided in the Classsur dataset, is a categorical variable. Therefore, we treat classes 2.0, 3.0, 65.0, 4.0, 68.6, 1.0, 68.0, 61.0, 64.0, 67.0, 66.5, 108.0, and 5.0 as separate groups.

```r
# identifying the unique groups in the Class standing variable
# unique(classsur$Class)

classes_to_compare <- c(2.0, 4.0, 68.0)
class_prop <- prop.table(table(classsur$Class))
filtered_prop <- class_prop[names(class_prop)
                            %in% classes_to_compare]
cat("proportions of classes 2.0, 4.0, and 68.0 each: ", filtered_prop)
```

```
## proportions of classes 2.0, 4.0, and 68.0 each:  0.1052632 0.2631579 0.03508772
```

SRS 3

```r
# read the Turkey2011fv.csv dataset, then check the structure of the csv file and a brief numerical sum
turkey2011 <- read.csv(here("Datasets", "Turkey2011fv.csv"))
str(turkey2011)
```

```
## 'data.frame':    199555 obs. of  14 variables:
##  $ X           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ region      : chr  "Adana" "Adana" "Adana" "Adana" ...
##  $ town        : chr  "Seyhan" "Seyhan" "Seyhan" "Seyhan" ...
##  $ neighborhood: chr  "İl/İlçe merkezi" "İl/İlçe merkezi" "İl/İlçe merkezi" "İl/İlçe merkezi" ...
##  $ electdist   : chr  "Ahmet Remzi Yüreğir" "Ahmet Remzi Yüreğir" "Ahmet Remzi Yüreğir" "Ahmet Remzi
##  $ SB          : chr  "Ahmet Remzi Yüreğir" "Ahmet Remzi Yüreğir" "Ahmet Remzi Yüreğir" "Ahmet Remzi
##  $ SB.Tip      : chr  "Mah." "Mah." "Mah." "Mah." ...
##  $ precinct    : int  3001 3002 3003 3004 2001 2002 2003 2004 2005 2006 ...
##  $ eftype      : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ NVoters     : int  277 278 277 278 287 287 289 287 289 288 ...
##  $ NValid      : int  224 228 214 240 235 228 245 238 244 246 ...
##  $ Votes       : int  88 95 86 96 5 8 23 1 6 16 ...
##  $ Ntfraudmean : num  0 8.34 0 8.38 0 ...
##  $ Nfraudmean  : num  0 42.7 0 42.7 0 ...
```

```r
summary(turkey2011)
```

```
##        X             region              town           neighborhood
##  Min.   :     1   Length:199555      Length:199555      Length:199555
##  1st Qu.: 49924   Class :character   Class :character   Class :character
##  Median : 99833   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 99831
```

```
##  3rd Qu.:149736
##  Max.   :199657
##   electdist              SB               SB.Tip             precinct
##  Length:199555      Length:199555      Length:199555      Min.   :1001
##  Class :character   Class :character   Class :character   1st Qu.:1059
##  Mode  :character   Mode  :character   Mode  :character   Median :1171
##                                                           Mean   :1488
##                                                           3rd Qu.:1494
##                                                           Max.   :9999
##      eftype         NVoters          NValid          Votes
##  Min.   :1.000   Min.   :  3.0   Min.   :  1.0   Min.   :  0.0
##  1st Qu.:1.000   1st Qu.:236.0   1st Qu.:197.0   1st Qu.: 75.0
##  Median :2.000   Median :284.0   Median :237.0   Median :113.0
##  Mean   :1.678   Mean   :251.8   Mean   :214.5   Mean   :111.7
##  3rd Qu.:2.000   3rd Qu.:294.0   3rd Qu.:253.0   3rd Qu.:149.0
##  Max.   :3.000   Max.   :406.0   Max.   :406.0   Max.   :297.0
##   Ntfraudmean       Nfraudmean
##  Min.   : 0.000   Min.   :  0.00
##  1st Qu.: 0.000   1st Qu.:  0.00
##  Median : 6.285   Median : 35.44
##  Mean   : 5.275   Mean   : 27.69
##  3rd Qu.: 8.581   3rd Qu.: 47.02
##  Max.   :55.443   Max.   :164.98
```

```r
# Population size N
N <- dim(turkey2011)[1]
cat("population size N: ", N)
```

```
## population size N:  199555
```

```r
# assign sample size n = 1500 as provided in problem instructions
n <- 1500

# filter data based on eftypes (1, 2, 3)
pop_1 <- subset(turkey2011, eftype == 1)$Nfraudmean
pop_2 <- subset(turkey2011, eftype == 2)$Nfraudmean
pop_3 <- subset(turkey2011, eftype == 3)$Nfraudmean
```

We estimate the difference of means of the number of eforensics-fraudulent votes (variable Nfraudmean) of two populations (eftype = 1, 2) based on srswor in the Turkey2011fv.csv file. From our computational calculation, we obtain the results as the following:

```r
# calculate means and standard errors
mean_1 <- mean(pop_1)
mean_2 <- mean(pop_2)
std_1 <- sd(pop_1)
std_2 <- sd(pop_2)

# compare the difference of means
diff_means <- mean_2 - mean_1

# standard error of difference
se_diff <- sqrt((std_1^2 / length(pop_1)) + (std_2^2 / length(pop_2)))

# construct a 95% CI
ci_lower <- diff_means - (1.96 * se_diff)
```

```r
ci_upper <- diff_means + (1.96 * se_diff)

cat("Population 1 (eftype 1) Mean number of eforensics-fraudulent votes:", mean_1, "\n")
```

```
## Population 1 (eftype 1) Mean number of eforensics-fraudulent votes: 0
```

```r
cat("Population 2 (eftype 2) Mean eforensics-fraudulent votes:", mean_2, "\n")
```

```
## Population 2 (eftype 2) Mean eforensics-fraudulent votes: 41.11932
```

```r
cat("Difference of Means:", diff_means, "\n")
```

```
## Difference of Means: 41.11932
```

```r
cat("Standard Error of Differences of Means", se_diff, "\n")
```

```
## Standard Error of Differences of Means 0.03424931
```

```r
cat("95% CI: [", ci_lower, ci_upper, "]", "\n")
```

```
## 95% CI: [ 41.05219 41.18645 ]
```

Now, let us estimate the mean number of eforensics-fraudulent votes by type (eftype = 2, 3) using sample size n = 1500 and estimate the difference of means between the two types, evaluating the bound on the error of estimation as the following:

```r
# create subset populations for eftypes 2 and 3
pop_2 <- turkey2011$Nfraudmean[turkey2011$eftype == 2]
pop_3 <- turkey2011$Nfraudmean[turkey2011$eftype == 3]

# create random samples; get sample mean and stdev
set.seed(123)
sample_2 <- sample(pop_2, n, replace = FALSE)
sample_3 <- sample(pop_3, n, replace = FALSE)

mean_sample_2 <- mean(sample_2)
mean_sample_3 <- mean(sample_3)
sd2 <- sd(sample_2)
sd3 <- sd(sample_3)

cat("Mean for eftype == 2:", mean_sample_2, "\n")
```

```
## Mean for eftype == 2: 41.60681
```

```r
cat("Mean for eftype == 3:", mean_sample_3, "\n")
```

```
## Mean for eftype == 3: 76.60147
```

```r
cat("Standard Deviation for eftype = 2:", sd2, "\n")
```

```
## Standard Deviation for eftype = 2: 11.93312
```

```r
cat("Standard Deviation for eftype = 3:", sd3, "\n")
```

```
## Standard Deviation for eftype = 3: 33.10292
```

```r
# diff in sample means
diff_means <- mean_sample_3 - mean_sample_2

n2 <- length(sample_2[!is.na(sample_2)])
n3 <- length(sample_3[!is.na(sample_3)])
```

```r
# standard error of difference
se_diff <- sqrt((sd2^2 / n2) + (sd3^2 / n3))

cat("Difference of Means:", diff_means, "\n")
```

```
## Difference of Means: 34.99466
```

```r
cat("Standard Error of the Difference:", se_diff, "\n")
```

```
## Standard Error of the Difference: 0.908553
```

```r
ci_lower <- diff_means - (1.96 * se_diff)
ci_upper <- diff_means + (1.96 * se_diff)

cat("Margin of Error:", 1.96 * se_diff)
```

```
## Margin of Error: 1.780764
```

```r
cat("95% CI: [", ci_lower, ci_upper, "]", "\n")
```

```
## 95% CI: [ 33.21389 36.77542 ]
```