M5

Will Tirone

Q1)

a)

To create the data set, I just copied the data and added columns for the weights and FPC. Since this is a two-stage cluster design, the weights are $\frac{N}{n}\frac{M_i}{m_i}$. Since we're interested in the female smokers, I used the total number of females as M_i for each school and N = 29 for the total number of schools.

Additionally, we interviewed 100 females, so we need a unique ID for each ssu. I added these ID's, and then arbitrarily said the first few number of each ID was a smoker, based on the number of female smokers interviewed. I added a 1 if they were a smoker and 0 if not.

```
school = c(1,2,3,4)
n_student = c(1471, 890, 1021, 1578)
n_female = c(792, 447, 511, 800)
n_female_int = c(25,15,20,40)
n_smoke = c(10, 3, 6, 27)
wt = (29 / 4) * c(792/25, 447/15, 511/20, 800/40)
fpc1 = rep(29, 4)

smoke = data.frame(school, n_student, n_female,
                   n_female_int, n_smoke, wt, fpc1)

smoke = bind_rows(
      smoke |> filter(school == 1) |> bind_cols(1:25),
      smoke |> filter(school == 2) |> bind_cols(26:40),
      smoke |> filter(school == 3) |> bind_cols(41:60),
      smoke |> filter(school == 4) |> bind_cols(61:100)
    ) |>
  rename(femaleID = ...8) |>
```

```
    mutate(smoke_status = 0)
```

```
New names:
New names:
New names:
New names:
* `` -> `...8`
```

```
  # adding column for status
  smoke$smoke_status[1:10] = 1
  smoke$smoke_status[26:28] = 1
  smoke$smoke_status[41:46] = 1
  smoke$smoke_status[61:87] = 1

  smoke_survey = svydesign(id = ~school + femaleID,
                           weights = ~wt,
                           fpc = ~fpc1 + n_female,
                           data=smoke)
```

Now, to estimate the percentage

```
  smean = svymean(~smoke_status, smoke_survey)
  print(smean[1])
```

```
smoke_status
   0.4311765
```

```
  confint(smean)
```

```
                2.5 %    97.5 %
smoke_status 0.23693 0.6254229
```

b)

Estimating the total here:

```
  stotal = svytotal(~smoke_status, smoke_survey)
  print(stotal[1])
```

```
smoke_status
    7971.375
```

```
confint(stotal)
```

```
                2.5 %   97.5 %
smoke_status 2629.159 13313.59
```

Q2)

```
n = 50
t_x = sum(grants$charged)

grants = grants |>
  mutate(e = charged - allowed,
         overcharge = if_else(e ==0, 0, 1),
         wt = t_x / (n * charged))

grant_survey = svydesign(~1, weights = ~wt, data = grants)
```

a)

Here, `overcharge` has a 1 if there is an overcharge and a 0 otherwise. To find the estimated percentage, we just use the svymean. The interval contains 0, so we're not convinced that the true population contains any errors.

```
charge = svymean(~overcharge, grant_survey)
print(charge[1])
```

```
overcharge
0.06519168
```

```
confint(charge)
```

```
                 2.5 %    97.5 %
overcharge -0.01771377 0.1480971
```

b)

Below I've estimated the total error and constructed a confidence interval based on the standard error. Since this interval covers 0, we're not confident that there is a significant amount of error in the population of charges vs. allowed.

```
error = svytotal(~e, grant_survey)
print(error[1])
```

```
       e
4691.408
```

```
print(confint(error))
```

```
      2.5 %   97.5 %
e -468.8526 9851.668
```

c)

Again since our interval covers zero, we're not confident that the population reflects a significant amount of errors.

```
ratio = svyratio(~e, ~charged, grant_survey)
print(ratio[1])
```

```
$ratio
      charged
e 0.009248505
```

```
confint(ratio)
```

```
                  2.5 %     97.5 %
e/charged -0.0009242823 0.01942129
```

Q3)

$$
\begin{aligned}
Bias &= E(\tilde{t_e}) - t_e \\
&= E(N \sum_{i \in S} \frac{e_i}{n}) - t_e \\
&= (N \sum_{i=1}^{N} \frac{e_i}{n} E(I_i)) - t_e \\
&= (N \sum_{i=1}^{N} \frac{e_i}{n} \frac{n x_i}{t_x}) - t_e \\
&= (\frac{1}{t_x} \sum_{i=1}^{N} e_i x_i) - t_e
\end{aligned}
$$

Q4)

a)

```
N = 3
n = 1

t_y = hypo |>
      group_by(cluster) |>
      summarise(t_y = sum(y)) |>
      pull()

T_y_bar = mean(t_y)

# compute values
S_t_2 = (1/(N-1)) * sum((t_y - T_y_bar)^2)
theoretical_var = N^2 * (1 - n/N) * (S_t_2) / n
cat("Theoretical variance for a one cluster sample = ", theoretical_var )
```

```
Theoretical variance for a one cluster sample =  28363254
```

b)

```
M0_Q4 = 30
m_Q4 = 10

# computer values
S_t_2 = var(hypo$y)
theoretical_var = M0_Q4^2 * (1 - m_Q4/M0_Q4) * S_t_2/m_Q4
cat("Theoretical variance for SRS w/ 10 individuals = ", theoretical_var )
```

```
Theoretical variance for SRS w/ 10 individuals =  1956635
```

c)

The variance for the cluster sample is much larger because we're only getting information about the survey variable from one of three clusters. Importantly, the cluster have significantly different values. Of course, we happen to know the full population which wouldn't normally be the case, but here we know the totals for each cluster are $(t_{y_1}, t_{y_2}, y_{y3}) = (29, 648, 4066)$. Doing a SRS gives us a better chance of sampling data points with different values since we're doing the SRS regardless of cluster.

d)

Note that since n_h and N_h are the same across the strata, we don't have to multiply them individually so I pulled them out of the sum.

```
n_h = 3
N_h = 10

Y_bar_h = hypo |>
      group_by(cluster) |>
      summarise(Y_bar = mean(y))

combined = left_join(hypo, Y_bar_h, by='cluster') |>
  mutate(e = (y-Y_bar)^2)

theoretical_var = (N_h^2 * (1-n_h/N_h))/(n_h * (N_h - 1)) * sum(combined$e)
cat("Theoretical variance for stratified sample w/ n_h = 3 :",
    theoretical_var )
```

```
Theoretical variance for stratified sample w/ n_h = 3 : 686.7778
```

e)

The real problem here is that we have 3 extremely different groups in the 3 clusters. If we cluster sample and select one cluster, we get a bad estimate because we ignore the other two. If we simple random sample, that's a little better. However, we could still get very unlucky and with 10 samples draw all of them from the same cluster and end up with the same issue. However, if we stratify and select 3 individuals from each strata, we guarantee a better estimate because we don't risk drawing all of our samples from the same group.

Q5)

a)

```r
# using code from Sakai here

N = 100

Mis = c(rep(100, 40), rep(300, 40), rep(500, 20))
M0 = sum(Mis)

y = c()
clusterindex = c()
clusterMis = c()

clustermean = c(rep(40, 40), rep(10, 40), rep(100, 20))
clustersd = c(rep(5, 40), rep(2, 40), rep(20,20))

for(i in 1:N) {
  temp = rep(i, Mis[i])
  clusterindex = c(clusterindex, temp)
  temp = rep(Mis[i], Mis[i])
  clusterMis = c(clusterMis, temp)
  temp = rnorm(Mis[i], clustermean[i], clustersd[i])
  y = c(y, temp)
}

# create universe
```

```r
U = data.frame(cbind(cluster = clusterindex, Mis = clusterMis, y = y))
```

b)

Code for the simulation is below, and the reported values are in the chunk after this one:

```r
set.seed(1234)
# save values here
data = matrix(nrow = 1000, ncol = 4)
colnames(data) = c("y_bar", "var_y_bar", "y_bar_HT", "var_Ybar_HT")

for (i in 1:1000) {

  n = 10

  # from 100 clusters draw 10
  sample_indexes = sample(100, n, replace = FALSE)

  # filter by our samples
  # and create grouped values
  cluster_sample = U |> filter(cluster %in% sample_indexes)
  t_yi = cluster_sample |>
    group_by(cluster) |>
    summarise(y_i = sum(y)) |>
    pull(y_i)

  # compute incorrect values
  m = dim(cluster_sample)[1] # number sampled cases
  y_bar = mean(cluster_sample$y)

  var_y_bar = (1 - m/M0) * (var(cluster_sample$y) / m)

  # correct HT values
  t_HT = (N/n) * sum(t_yi)
  y_bar_HT = (1/M0) * t_HT
  s_2_t = (1/(n-1)) * sum((t_yi - (t_HT/N))^2)
  var_Ybar_HT = (N^2 * (1-n/N) * (s_2_t/n)) / M0^2

  data[i,] = c(y_bar, var_y_bar, y_bar_HT, var_Ybar_HT)
```

```
}
```

Reporting values:

```
cat("average of y tilde", mean(data[,'y_bar']), "\n")
```

```
average of y tilde 46.95533
```

```
cat("variance of y tilde", var(data[,'y_bar']), "\n")
```

```
variance of y tilde 259.0359
```

```
cat("average of variance of y tilde", mean(data[, 'var_y_bar']), "\n")
```

```
average of variance of y tilde 0.5333681
```

```
cat("average of HT estimator", mean(data[, 'y_bar_HT']), "\n")
```

```
average of HT estimator 48.09679
```

```
cat("variance of HT estimator", var(data[, 'y_bar_HT']), "\n")
```

```
variance of HT estimator 463.9072
```

```
cat("average of variance of HT estimator", mean(data[, 'var_Ybar_HT']))
```

```
average of variance of HT estimator 454.8781
```

c)

$\tilde{y}$ is biased, though actually not that severely. With the random seed I have, $\tilde{y} = 46.8$ while the HT estimator, which we know is always unbiased, is 48.

d)

$\hat{var}(\tilde{y}) = 0.53$ is extremely biased, considering our unbiased HT estimator of the variance is 451.7. This estimate, ignoring the differences across clusters, severely affects our estimate.

e)

The HT estimator is very close to the theoretical mean, which is a good sign since we know from the theory it should be unbiased.

```
mean(U$y)
```

```
[1] 49.30135
```

Last, we just compare the variance of the mean and the mean of the variance, and since we see they are very close, we are reassured that the theoretical value for the variance and HT estimator for the variance are unbiased.

```
cat("variance of HT estimator", var(data[, 'y_bar_HT']), "\n")
```

```
variance of HT estimator 463.9072
```

```
cat("average of variance of HT estimator", mean(data[, 'var_Ybar_HT']))
```

```
average of variance of HT estimator 454.8781
```