MATH 525 - Assignment 2

Emir Sevinc 260682995

February 16, 2019

3.24

We need to minimize $V = \sum_{h=1}^{H} \frac{N_h^2 S_h^2}{nh}(1 - \frac{n_h}{S_h})$ subject to the constraint $C = c_0 + \sum_{h=1}^{H} c_h n_h$
Larangue multiplier says $\nabla V = \lambda \nabla C$
For a general h, the term of the gradient for V will simply be the derivative with respect to n_h that is:
$\frac{\partial V}{\partial n_h} = \frac{\partial}{\partial n_h} \frac{N_h^2 S_h^2}{nh} - \frac{N_h^2 S_h^2}{nh} \frac{n_h}{N_h} = \frac{\partial}{\partial n_h} \frac{N_h^2 S_h^2}{nh} - N_h S_h^2$
$= -\frac{N_h^2 S_h^2}{nh^2}$
We also have $\nabla C = c_h$ for a general h, thus we need $= -\frac{N_h^2 S_h^2}{nh^2} = \lambda c_h \implies n_h^2 = -\frac{N_h^2 S_h^2}{\lambda c_h} \implies n_h = -\frac{N_h S_h}{\sqrt{\lambda c_h}} \implies n_h = \frac{-1}{\sqrt{\lambda}} \frac{N_h S_h}{\sqrt{c_h}}$, so we have that n_h is indeed proportional to $\frac{N_h S_h}{\sqrt{c_h}}$

3.37

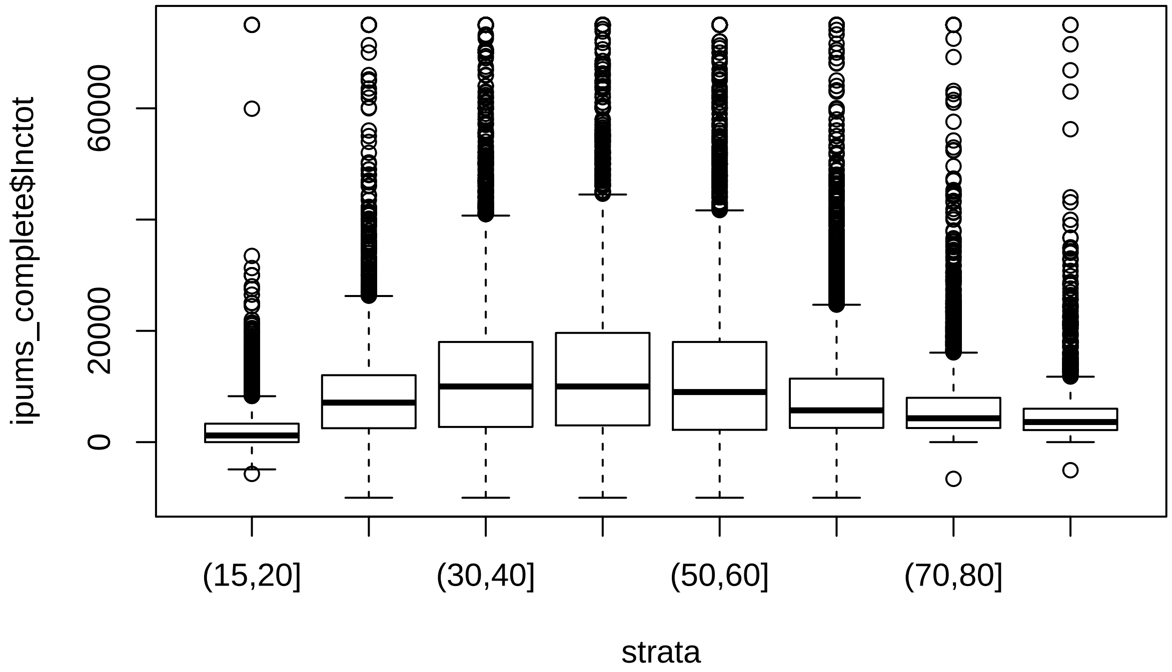
a)

```
set.seed(19950125)
library(tidyverse)
library(survey)
library(srvyr)
library(knitr)
library(kableExtra)

ipums=read_csv("ipums.csv")



ipums_complete = ipums %>% filter(!is.na(Inctot))
strata <- cut(ipums_complete$Age, breaks=c(15,20,30,40,50,60,70,80,90))
set_new <- split(ipums_complete, strata)
plot(ipums_complete$Inctot~strata)
```



b)

Below are the number of elements in each stratum:

```
set.seed(19950125)
samp_n<-1206 #the optimal total sample size we found on assignment 1

dims<-integer(8)

i <- 1
for (i in c(1:8)) {
  dims[i]<-nrow(as.data.frame(set_new[i]))
}
dims
```

```
## [1]  6395 12511  9349  6883  7067  5542  3210  1363
```

Proportion of our sample (1206/pop size)

```
set.seed(19950125)
sampling_proprotion<-(1206/nrow(ipums_complete))
sampling_proprotion
```

```
## [1] 0.0225585
```

So we're sampling about 2.25% of the population with the sample size we found on the previous assignment, so if we are to use proportional allocation we would sample 2.25% of each stratum as well. Below we find the correct number for each stratum:

```r
set.seed(19950125)

sample_sizes<-integer(8)

i <- 1
for (i in c(1:8)) {
  sample_sizes[i]<-ceiling((sampling_proprotion*dims[i]))
}
sample_sizes
```

```
## [1] 145 283 211 156 160 126  73  31
```

So we need those sample sizes from each strata and perform SRS.

Below we sample the appropriate amounts (as listed in the vector above) from each individual stratum, and perform simple random samples:

```r
set.seed(19950125)

ipums_145= as.data.frame(set_new[1]) %>% slice(sample(1:nrow(as.data.frame(set_new[1])),
                                                size=145, replace=F))

ipums_283= as.data.frame(set_new[2]) %>% slice(sample(1:nrow(as.data.frame(set_new[2])),
                                                size= 283, replace=F))

ipums_211= as.data.frame(set_new[3]) %>% slice(sample(1:nrow(as.data.frame(set_new[3])),
                                                size=211, replace=F))

ipums_156= as.data.frame(set_new[4]) %>% slice(sample(1:nrow(as.data.frame(set_new[4])),
                                                size=156, replace=F))

ipums_160= as.data.frame(set_new[5]) %>% slice(sample(1:nrow(as.data.frame(set_new[5])),
                                                size=160, replace=F))

ipums_126= as.data.frame(set_new[6]) %>% slice(sample(1:nrow(as.data.frame(set_new[6])),
                                                size=126, replace=F))

ipums_73= as.data.frame(set_new[7]) %>% slice(sample(1:nrow(as.data.frame(set_new[7])),
                                                size=73, replace=F))

ipums_31= as.data.frame(set_new[8]) %>% slice(sample(1:nrow(as.data.frame(set_new[8])),
                                                size=31, replace=F))
```

c)

```r
set.seed(19950125)

ipums_design_1 = survey::svydesign(id=~1,data=ipums_145, fpc=rep(dims[1],sample_sizes[1]))
ipums_design_2 = survey::svydesign(id=~1,data=ipums_283, fpc=rep(dims[2],sample_sizes[2]))
ipums_design_3 = survey::svydesign(id=~1,data=ipums_211, fpc=rep(dims[3],sample_sizes[3]))
ipums_design_4 = survey::svydesign(id=~1,data=ipums_156, fpc=rep(dims[4],sample_sizes[4]))
ipums_design_5 = survey::svydesign(id=~1,data=ipums_160, fpc=rep(dims[5],sample_sizes[5]))
ipums_design_6 = survey::svydesign(id=~1,data=ipums_126, fpc=rep(dims[6],sample_sizes[6]))
ipums_design_7 = survey::svydesign(id=~1,data=ipums_73, fpc=rep(dims[7],sample_sizes[7]))
```

```
ipums_design_8 = survey::svydesign(id=~1,data=ipums_31, fpc=rep(dims[8],sample_sizes[8]))

st1<-svytotal(~X.15.20..Inctot,ipums_design_1) #Survey Totals for Each
st2<-svytotal(~X.20.30..Inctot,ipums_design_2)
st3<-svytotal(~X.30.40..Inctot,ipums_design_3)
st4<-svytotal(~X.40.50..Inctot,ipums_design_4)
st5<-svytotal(~X.50.60..Inctot,ipums_design_5)
st6<-svytotal(~X.60.70..Inctot,ipums_design_6)
st7<-svytotal(~X.70.80..Inctot,ipums_design_7)
st8<-svytotal(~X.80.90..Inctot,ipums_design_8)


FinalTotal<-st1+st2+st3+st4+st5+st6+st7+st8 #Pooling the Results
FinalTotal
```

```
##                       total      SE
## X.15.20..Inctot 508312996 2029180
```

As we can see, the total is estimated as 508312996, and on the previous assignment I had estimated it as 482376254. Note that the standard error here is 2029180, significantly lower than what I had for the previous assignment (16418766).

Confidence Interval:

```
confint(FinalTotal, level=0.95)
```

```
##                       2.5 %    97.5 %
## X.15.20..Inctot 504335877 512290115
```

d)

Normally optimal allocation means sample sizes proportional to $\frac{N_h S_h}{\sqrt{c_h}}$ and the optimal n_h as we have seen in class ought to be $\frac{N_h S_h/\sqrt{c_h}}{\sum_{t=1}^{H} N_t S_t/\sqrt{c_t}} * n$: but we dont have a cost function c_h so the optimanl sample size n_h for each stratum will be $\frac{N_h S_h}{\sum_{t=1}^{H} N_t S_t} * n$
To get this we will need to estimate the variances from each stratum. Below we draw samples of 200 from each stratum and estimate the variances:

```
set.seed(19950125)

pilot_sample_1= as.data.frame(set_new[1]) %>% slice(sample(1:nrow(as.data.frame(set_new[1])),
                                               size=200, replace=F))

svar1<-var(pilot_sample_1$X.15.20..Inctot)


pilot_sample_2= as.data.frame(set_new[2]) %>% slice(sample(1:nrow(as.data.frame(set_new[2])),
                                               size=200, replace=F))

svar2<-var(pilot_sample_2$X.20.30..Inctot)


pilot_sample_3= as.data.frame(set_new[3]) %>% slice(sample(1:nrow(as.data.frame(set_new[3])),
                                               size=200, replace=F))

svar3<-var(pilot_sample_3$X.30.40..Inctot)
```

```
pilot_sample_4= as.data.frame(set_new[4]) %>% slice(sample(1:nrow(as.data.frame(set_new[4])),
                                               size=200, replace=F))

svar4<-var(pilot_sample_4$X.40.50..Inctot)


pilot_sample_5= as.data.frame(set_new[5]) %>% slice(sample(1:nrow(as.data.frame(set_new[5])),
                                               size=200, replace=F))

svar5<-var(pilot_sample_5$X.50.60..Inctot)


pilot_sample_6= as.data.frame(set_new[6]) %>% slice(sample(1:nrow(as.data.frame(set_new[6])),
                                               size=200, replace=F))

svar6<-var(pilot_sample_6$X.60.70..Inctot)


pilot_sample_7= as.data.frame(set_new[7]) %>% slice(sample(1:nrow(as.data.frame(set_new[7])),
                                               size=200, replace=F))

svar7<-var(pilot_sample_7$X.70.80..Inctot)


pilot_sample_8= as.data.frame(set_new[8]) %>% slice(sample(1:nrow(as.data.frame(set_new[8])),
                                               size=200, replace=F))

svar8<-var(pilot_sample_8$X.80.90..Inctot)

VarTotal<-svar1+svar2+svar3+svar4+svar5+svar6+svar7+svar8

Stratum_variances<-c(svar1,svar2,svar3,svar4,svar5,svar6,svar7,svar8)
Stratum_variances
```

```
## [1]  15913790  74460993 101747457 177207220 159904259  82575540  60083680
## [8]  55891194
```

So we found the variances for each stratum.

Before proceeding we need $\sum_{t=1}^{H} N_t S_t$, the denominator of our optimal sample size term:

```
set.seed(19950125)

vec_denom<-integer(8)

i <- 1
for (i in c(1:8)) {
  vec_denom[i]<-(dims[i]*Stratum_variances[i]) #t'th element of the vector is Nt*St
}
denom<-sum(vec_denom) #adding them all to find the sum
denom
```

```
## [1] 5.06103e+12
```

So we found $\sum_{t=1}^{H} N_t S_t$ as 5.06103e+12. So now we proceed to find the optimal sample sizes for each stratum using our optimal sample size formula:

```
set.seed(19950125)

optimal_sample_sizes<-integer(8)

i <- 1
for (i in c(1:8)) {
  optimal_sample_sizes[i]<-ceiling((((dims[i]*Stratum_variances[i])/denom)*1206))
}
optimal_sample_sizes
```

```
## [1]  25 222 227 291 270 110  46  19
```

So we have computed the optimal sample sizes for each straum respectively. Now we need to repeat the previous part with these sample sizes instead. We implement it below:

```
set.seed(19950125)

ipums_25= as.data.frame(set_new[1]) %>% slice(sample(1:nrow(as.data.frame(set_new[1])),
                                               size=25, replace=F))

ipums_222= as.data.frame(set_new[2]) %>% slice(sample(1:nrow(as.data.frame(set_new[2])),
                                               size= 222, replace=F))

ipums_227= as.data.frame(set_new[3]) %>% slice(sample(1:nrow(as.data.frame(set_new[3])),
                                               size=227, replace=F))

ipums_291= as.data.frame(set_new[4]) %>% slice(sample(1:nrow(as.data.frame(set_new[4])),
                                               size=291, replace=F))

ipums_270= as.data.frame(set_new[5]) %>% slice(sample(1:nrow(as.data.frame(set_new[5])),
                                               size=270, replace=F))

ipums_110= as.data.frame(set_new[6]) %>% slice(sample(1:nrow(as.data.frame(set_new[6])),
                                               size=110, replace=F))

ipums_46= as.data.frame(set_new[7]) %>% slice(sample(1:nrow(as.data.frame(set_new[7])),
                                               size=46, replace=F))

ipums_19= as.data.frame(set_new[8]) %>% slice(sample(1:nrow(as.data.frame(set_new[8])),
                                               size=19, replace=F))
```

```
set.seed(19950125)

optimal_design_1 = survey::svydesign(id=~1,data=ipums_25, fpc=rep(dims[1],optimal_sample_sizes[1]))
optimal_design_2 = survey::svydesign(id=~1,data=ipums_222, fpc=rep(dims[2],optimal_sample_sizes[2]))
optimal_design_3 = survey::svydesign(id=~1,data=ipums_227, fpc=rep(dims[3],optimal_sample_sizes[3]))
optimal_design_4 = survey::svydesign(id=~1,data=ipums_291, fpc=rep(dims[4],optimal_sample_sizes[4]))
optimal_design_5 = survey::svydesign(id=~1,data=ipums_270, fpc=rep(dims[5],optimal_sample_sizes[5]))
optimal_design_6 = survey::svydesign(id=~1,data=ipums_110, fpc=rep(dims[6],optimal_sample_sizes[6]))
optimal_design_7 = survey::svydesign(id=~1,data=ipums_46, fpc=rep(dims[7],optimal_sample_sizes[7]))
optimal_design_8 = survey::svydesign(id=~1,data=ipums_19, fpc=rep(dims[8],optimal_sample_sizes[8]))

optimal_total_1<-svytotal(~X.15.20..Inctot,optimal_design_1) #Survey Totals for Each
optimal_total_2<-svytotal(~X.20.30..Inctot,optimal_design_2)
```

```
optimal_total_3<-svytotal(~X.30.40..Inctot,optimal_design_3)
optimal_total_4<-svytotal(~X.40.50..Inctot,optimal_design_4)
optimal_total_5<-svytotal(~X.50.60..Inctot,optimal_design_5)
optimal_total_6<-svytotal(~X.60.70..Inctot,optimal_design_6)
optimal_total_7<-svytotal(~X.70.80..Inctot,optimal_design_7)
optimal_total_8<-svytotal(~X.80.90..Inctot,optimal_design_8)


OptimalTotal<-optimal_total_1+optimal_total_2+optimal_total_3+optimal_total_4+optimal_total_5+optimal_to
OptimalTotal
```

```
##                     total      SE
## X.15.20..Inctot 515525529 5571746
```

Thus we have found our total estimate to be 515525529 and the standard error is found as 5571746

```
confint(OptimalTotal, level=0.95)
```

```
##                     2.5 %    97.5 %
## X.15.20..Inctot 504605108 526445951
```

The confidence interval is provided above.

e)

The standard error we found through optimal allocation, 5571746, is higher than the one we had for the regular stratified sampling, which was 2029180, but STILL lower than what we had for assignment 1 (16418766). For optimal sampling to perform much better, we need a very significant differences in variances between the strata, however this is not the case for this data set, as we found on part d), and we can look at the coefficient of variation to confirm:

```
SD<-sd(Stratum_variances)
MEAN<-mean(Stratum_variances)
CV<-(SD/MEAN)
CV
```

```
## [1] 0.5943655
```

A 60% difference on average is “relatively high” but evidentally not high enough to justify the usage of optimal allocaiton instead of proportional.

f)

Summarising the Standard Errors:

Simple Random Sample (Assignment 1):16418766
Stratified Sampling with Proportional Allecation:2029180
Stratified Sampling with Optimal Allecation:5571746

As we can see the lowest we got by far was using PA, but what we had for OA was still lower than the SRS so I would say that the stratifying was worth while, but Optimal Allication was not (relative to PA, it is still worth stratifying with OA if our other option is SRS). Comparing estimate totals:

Simple Random Sample (Assignment 1):482376254
Stratified Sampling with Proportional Allecation:504335877
Stratified Sampling with Optimal Allecation:504605108

PA and OA gave extremely similar estimates for the total. This further contributes to the conclusion that stratifying was worth while, as the lower SE cause their estimates to be more trustworthy.

If I were to start over, what I would do differently would depend on what I'm trying to accomplish. Stratifying by age differences gave a "good" overall result, with lower SE than a SRS, but I may want to consider observing race, maritial status, sex or others to see if the stratatification causes extreme differences in variance between the strata. If so, then optimal allocation may end up giving a more precise result.