MATH525,Assignment1

Emir Sevinc 260682995

January 26, 2019

2.24

We have $L(n)+C(n) = k(1-\frac{n}{N})*\frac{S^2}{n}+c_0+c_1n$, we need to find n that minimizes this. So we can differentiate with respect to n, set it equal to 0. First we rewrite: $L(n)+C(n)$
$= k(1-\frac{n}{N})*\frac{S^2}{n}+c_0+c_1n$
$= (k-\frac{nk}{N})*\frac{S^2}{n}+c_0+c_1n$
$= \frac{kS^2}{n}-\frac{nkS^2}{nN}+c_0+c_1n$
$= \frac{kS^2}{n}-\frac{kS^2}{N}+c_0+c_1n$. Differentiating with respect to n gives:
$\frac{-kS^2}{n^2}+c_1$. Now we set to 0 and solve;
$\frac{-kS^2}{n^2}+c_1 = 0 \implies c_1 = \frac{kS^2}{n^2} \implies \frac{n^2}{kS^2} = \frac{1}{c_1} \implies n^2 = \frac{kS^2}{c_1} \implies n = \pm\sqrt{\frac{kS^2}{c_1}}$. Since a sample can't be negative we take the positive value and so n has to be $\sqrt{\frac{kS^2}{c_1}}$

To verify that it is indeed the minimum, we take the second derivative and ensure that it is positive. Differentiating once again gives $\frac{2kS^2}{n^3}$. sˆ2 and n are definitely positive, since sˆ2 is variance and n is a set size, so we only need k to be positive. But one of the original cost functions was $L(n)+ = k(1-\frac{n}{N})*\frac{S^2}{n}$, which must be positive since it represents cost. We know that $n \leq N$, thus $\frac{n}{N} \leq 1 \implies 1-\frac{n}{N} \geq 0$, and since we already established that sˆ2 and n are positive, than we must have $k \geq 0$ also, and our n indeed corresponds to a minima.

2.26

Note that the probability of a unit being included in the sample is equivalent to the probability choosing any integer between 1 to k, $\frac{1}{k} = \frac{n}{N}$

However, suppose for a systematic sample N=10,n=2 so k = 5, then all of the subsets we can choose of size n = 2 are: (1,6), (2,7), (3,8), (4,9), (5,10), 5 in total, and the probability of choosing any one of them depends entirely on what number we picked between 1 and k, so it's 1/5, but we have that $1/\binom{N}{n} = 1/\binom{10}{2} = 45 \neq \frac{1}{5}$. We can generalise this: Given N,n, and k, the following is a list of all the possible samples: (1,1+k,1+2k,...),(2,2+k,2+2k,...),...,(k,2k,3k,...). So there are k of them and the probability of choosing any is $\frac{1}{k}$

2.28

a)

The multinomial distribution corresponds to n indpendent, identical trials, where the outcome of each trial falls into one of k "classes". The probability that the outcome of a single trial falls into class i is p_i, and remains the same for each trial. The radom variables of interest are $Y_1, Y_2, ..., Y_k$ where Y_i is the number of trials for which the outcome falls into class i.

Here, our categories are the units themselves, that is there are N classes and the i'th class is "the picked unit is unit i", and the random variables are $Q_1, Q_2, ...$ where Q_i is the number of times unit i appeared in the sample. Clearly since each sample taken is independend, the probabilty of our unit i being picked is $\frac{1}{N}$, and since the sample size is n this corresponds to n trials. Due to replacement, wether or not unit i is the one we picked does not depend on our previous pick. So we have fulfilled the requirements of a multinomial distribution, and the joint distribution of $Q_1, Q_2, ...$ is multinomial with n trials.

b)

We have that $t = \sum_{i=1}^{N} y_i$, so $E[\hat{t}] = E[\frac{N}{n}\sum_{i=1}^{N} Q_i y_i] = \frac{N}{n}\sum_{i=1}^{N} E(Q_i y_i) = \frac{N}{n}\sum_{i=1}^{N} y_i E(Q_i)$. By part a),$E[Q_i] = np_i = \frac{n}{N} \implies$
$E[\hat{t}] = \frac{N}{n}\sum_{i=1}^{N} \frac{n}{N} y_i = \sum_{i=1}^{N} y_i = t$

c)

$Var[\hat{t}] = Var(\frac{N}{n}\sum_{i=1}^{N} Q_i y_i) = (\frac{N}{n})^2 * Var[\sum_{i=1}^{N} Q_i y_i]$
Using the variance of a linear function of random variables, we have:
$Var[\hat{t}] = (\frac{N}{n})^2(\sum_{i=1}^{N} y_i^2 Var(Q_i) + 2\sum_{i=1}^{N}\sum_{j\neq i=1}^{N} y_i y_j Cov(Q_i, Q_j))$
. From part a), we know that $Var[Q_i] = np_i q_i = n(\frac{1}{N})(\frac{N-1}{N}) = \frac{n(N-1)}{N^2}$, and $Cov(Q_i, Q_j) = -np_i p_j = -n(\frac{1}{N})\frac{1}{N} = \frac{-n}{N^2} \implies$
$Var[\hat{t}] = (\frac{N}{n})^2 * [\frac{n(N-1)}{N^2}\sum_{i=1}^{N} y_i^2 + \sum_{i=1}^{N}\sum_{j\neq i}^{N} y_i y_j \frac{-n}{N^2}]$
$= (\frac{N}{n})^2 \frac{n}{N}\sum_{i=1}^{N} y_i^2 - \frac{n}{N^2}\sum_{i=1}^{N}\sum_{i=1}^{N} y_i y_j = \frac{N}{n}\sum_{i=1}^{N} y_i^2 - \frac{1}{n}\sum_{i=1}^{N}\sum_{i=1}^{N} y_i y_j = \frac{N}{n}N(\sum_{i=1}^{N} y_i^2 - \frac{1}{N}(\sum_{i=1}^{N} y_i)^2)$.
Since $S^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2$ we get: $= (\frac{N}{n}) * (N-1) * S^2$ for the variance.

2.29

Induction on N. If N = 1, there is only one possible sample such that that is n = 1, and the probability of it being choosen is $\frac{1}{\binom{1}{1}} = 1$ so the “0th” sample is trivially SRS.

Assume S_{N-n} is an SRS. If ES_{N-n} corresponds to the event of it being selected, we have $P(ES_{N-n}) = \frac{1}{\binom{N}{n}}$
Case 1: $u_{N+1-n} > \frac{n}{N+1}$. Since u_{N+1-n} is uniformly distributed, and ES_{N-n} is assumed to be SRS, and further assuming that u_{N+1-n} and ES_{N-n} are independent events, the event we want is " ES_{N-n} AND" $u_{N+1-n} > \frac{n}{N+1}$. Due to the independence assumed, this gives:

$= \frac{1}{\binom{N}{n}}[1 - P(u_{N+1-n} < \frac{n}{N+1})]$
$= \frac{(N-n)!n!}{(N)!}[1 - \int_0^{\frac{n}{N+1}} 1 * d_{u_{N+1-n}}]$
$= \frac{(N-n)!n!}{(N)!}[1 - [u_{N+1-n}]_0^{\frac{n}{N+1}}]$
$= \frac{(N-n)!n!}{(N)!}[1 - \frac{n}{N+1}]$
$= \frac{(N-n)!n!}{(N)!}[\frac{N+1-n}{N+1}]$
. Since $k! = k(k-1)!$ and $(n+k)! = (n+k)(n+k-1)!$; this gives $\frac{n!(N+1-n)!}{(N+1)!}$. Note that this is equal to $\frac{1}{\binom{N+1}{n}}$, so it's a simple random sample of size n from a list of N+1 elements.

Case 2: $u_{N+1-n} < \frac{n}{N+1}$. Let “Rep” be the event of selecting a unit being selected from S_{N-n} and replaced with another, and “Old”" be the event that the set every element that isn't being replaced belongs to S_{N-n} (that is the event of having selected the members of the previous sample that aren't being replaced). We have that $P(Old) = (N+1-n)\frac{1}{\binom{N}{n}}$, and the event we want now is “Rep AND Old AND $u_{N+1-n} < \frac{n}{N+1}$
” $= P(Rep/u_{N+1-n} < \frac{n}{N+1}, Old) * P(u_{N+1-n} < \frac{n}{N+1}, Old)$
$= P(Rep/u_{N+1-n} < \frac{n}{N+1}, Old) * P(u_{N+1-n} < \frac{n}{N+1}) * P(Old)$
by independence, and: $= \frac{1}{n} * \frac{n}{N+1} * (N+1-n)\frac{1}{\binom{N}{n}}$ since “Rep” can be assumed to be discrete uniform. This gives
$= \frac{1}{n} * \frac{n}{N+1} * (N+1-n)\frac{1}{\binom{N}{n}}$

$= \frac{1}{N+1} * (N+1-n) * \frac{(N-n)!n!}{N!}$
$= \frac{n!(N+1-n)!}{(N+1)!}$. Note that this is equal to $\frac{1}{\binom{N+1}{n}}$, so it's an SRS. We're done.

2.37)

a)

We're working with a type of censored data; and since income above 75000 is topcoded our estimates will likely underestimate the true income by quite a bit.

b)

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

ipums=read_csv("ipums.csv")
```

```
head(ipums)
```

```
## # A tibble: 6 x 16
##   Stratum   Psu Inctot   Age   Sex  Race Hispanic Marstat Ownershg Yrsusa
##     <int> <int>  <int> <int> <int> <int>    <int>   <int>    <int>  <int>
## 1       1     1   4105    18     1     2        0       5        0      0
## 2       1     1   7795    20     1     1        0       5        2      0
## 3       1     1  16985    24     1     1        0       1        1      0
## 4       1     1   7045    21     1     1        0       1        2      0
## 5       1     1   2955    23     1     1        0       5        2      0
## 6       1     1      0    17     1     1        0       5        1      0
## # ... with 6 more variables: School <int>, Educrec <int>, Labforce <int>,
## #   Occ <int>, Classwk <int>, VetStat <int>
```

```
ipums_complete = ipums %>% filter(!is.na(Inctot)) #cleaning any potential NA values just in case

ipums_complete %>% summarise(TotalInctot = sum(Inctot)) #actual total income, for later verification
```

```
## # A tibble: 1 x 1
##   TotalInctot
##         <int>
## 1  491533095
```

```
set.seed(19950125)
ipums_50 = ipums_complete %>% slice(sample(1:nrow(ipums_complete),
                                           size=50, replace=F)) #grabbing a sample of 50
dim(ipums_50) #verifying that the size is correct
```

```
## [1] 50 16
```

```
head(ipums_50)
```

```
## # A tibble: 6 x 16
##   Stratum   Psu Inctot   Age   Sex  Race Hispanic Marstat Ownershg Yrsusa
##     <int> <int>  <int> <int> <int> <int>    <int>   <int>    <int>  <int>
```

```
## 1        9     84  15005     38      1      1         0        1          1        0
## 2        8     78  13415     54      1      1         0        1          1        0
## 3        3     25    765     22      2      1         0        5          1        0
## 4        2     19      0     42      2      1         0        1          1        0
## 5        7     62   4125     17      1      1         0        5          0        0
## 6        1     10  52010     59      2      1         0        4          1        0
## # ... with 6 more variables: School <int>, Educrec <int>, Labforce <int>,
## #   Occ <int>, Classwk <int>, VetStat <int>
```

```
N <- 53461 #Population Size
ipums_50  %>%
  summarise(SampleMean=mean(Inctot),
                        SampleVar = var(Inctot))  %>% #Vital Informaiton
  gather(statistic,value) %>%
  kable(.,format="latex",digits=0) %>%
  kable_styling(.)
```

statistic	value
SampleMean	10884
SampleVar	157283328

So the sample variance is found to be 157283328. We have shown in class that for a desired absolute error e, the sample size n needs to be $\frac{s^2 * z^2_{\alpha/2}}{e^2 + \frac{s^2 * z^2_{\alpha/2}}{N}}$. If we are to assume normality, we will have that $z_{\alpha/2} = 1.96$. The rest were found to be $s^2 = 157283328$, $N = 53461$ and e is supposed to be 700. Plugging all those in gives $\frac{(157283328*1.96^2)}{(700^2 + \frac{(157283328*1.96^2)}{53461})} = 1205.3$, rolled up to 1206.

c)

```
set.seed(19950125)
n<-1206
ipums_1206= ipums_complete %>% slice(sample(1:nrow(ipums_complete),
                                           size=1206, replace=F)) #grabbing a sample of 407
dim(ipums_1206)
```

```
## [1] 1206   16
```

```
head(ipums_1206)
```

```
## # A tibble: 6 x 16
##   Stratum   Psu Inctot   Age   Sex  Race Hispanic Marstat Ownershg Yrsusa
##     <int> <int>  <int> <int> <int> <int>    <int>   <int>    <int>  <int>
## 1        9     84  15005     38      1      1         0        1          1        0
## 2        8     78  13415     54      1      1         0        1          1        0
## 3        3     25    765     22      2      1         0        5          1        0
## 4        2     19      0     42      2      1         0        1          1        0
## 5        7     62   4125     17      1      1         0        5          0        0
## 6        1     10  52010     59      2      1         0        4          1        0
## # ... with 6 more variables: School <int>, Educrec <int>, Labforce <int>,
## #   Occ <int>, Classwk <int>, VetStat <int>
```

```
ipums_design = survey::svydesign(id=~1,data=ipums_1206, fpc=rep(53461,1206)) #estimating total income

svytotal(~Inctot,ipums_design)
```

```
##             total       SE
## Inctot 482376254 16418766
```

So we can see the estimated total,

Confidence Interval:

```r
confint(svytotal(~Inctot,ipums_design),level=0.95)
```

```
##             2.5 %    97.5 %
## Inctot 450196064 514556444
```

and the CI's are sen above. It captures theactual population total of 491533095 we found earlier.